

Matrix inversion algorithm: applications in high speed MIMO LTE receiver

Md. Abdul Latif Sarker^{1,2,*}, Moon Ho Lee^{1,2}

¹Dept. Of Electronic Engineer, Jeonju City, Republic of Korea

²Chonbuk National University, Jeonju City, Republic of Korea

Email address:

latifsarker@jbnu.ac.kr (Md. Abdul Latif Sarker), moonho@jbnu.ac.kr (M. Ho Lee)

To cite this article:

Md. Abdul Latif Sarker, Moon Ho Lee. Matrix Inversion Algorithm: Applications in High Speed MIMO LTE Receiver. *American Journal of Networks and Communications*. Vol. 1, No. 1, 2012, pp. 1-6. doi: 10.11648/j.ajnc.20120101.11

Abstract: Matrix inversion algorithm acting an important role in MIMO wireless communication. In this paper, we present a matrix inversion algorithm and its applications in the high speed MIMO LTE receiver which is based on floating point DSP. Matrix operations are the most costly computational module within MIMO receivers but a matrix inversion algorithm is very easy to compute and significantly reduce the computational module cost. We will demonstrate the MIMO LTE applications for reducing the module cost by square matrix inversion algorithms.

Keywords: System Model; MIMO-LTE, Matrix Inversion Algorithm; Polar Decomposition, Matrix Inversion; Floating Point DSP, Implementation

1. Introduction

Multi Input Multi Output (MIMO) -Long Term Evolution (LTE) is the one of the newest technologies in wireless communications to improve bandwidth utilization efficiency. The access mode of multi-user MIMO LTE using a popular digital scheme Orthogonal Frequency Division Multiple Access (OFDMA) for downlink and Sub-Carrier Frequency Division Multiple Access (SC-FDMA) for uplink which provides high data rate in wireless environments. Multiple access channels are achieved in OFDMA by assigning narrow sub-bands, each narrow sub-band has a flat frequency response and frequency selective channel is converted into a lot of flat-fading sub-channels. This can achieve a higher MIMO spectral efficiency averaging interferences from neighbouring cells and less affected to various kinds of impulse noise. The floating point DSP's have better precision, higher dynamic range, and a smaller development cycle. These features result in reduced receiver complexity. This paper is organized as follows: In section 2, 3 and 4, we have described system model, matrix inversion algorithms and polar decomposition with matrix inversion respectively. The implementation and simulation results are illustrated in Section 5. Finally, conclusions are presented in Section 6.

2. System Model

Most of the detection or the equalization process needs to invert a matrix which is either the channel state information (W) or a nonlinear function of it (function(W)). If increasing the number of transmitter and receiver antennas is provide a higher data rate and the dimension of matrix function $f(w)$ increases, for more computations to invert the matrix in fewer times. In computing, floating point system represents real numbers which support a wide range of values. The numbers are in general represented to a fixed point or fixed numbers are of significant digits and scaled using an exponent. The base for the scaling is normally 2, 10, and 16. Symbolic form $s*b^e$, where s is the value of the significant, b is the base, e is the exponent. The advantage of floating point over fixed point representation is that it can support a much wider range of values. The floating point format needs slightly more storage, if stored in the same space; floating point numbers achieve their greater range at the expense of precision. The speed of floating point operations is an important measure of performance for computation. For LTE application we have used MMSE equalization or detection algorithm. In Figure 1 shows the MMSE detection block diagram.

Let A is the 4×4 matrix for 4-stream MIMO-LTE and \hat{F}_{MMSE} , the MMSE output equalizer, the solution can be written as below [1]:

$$A = \left(W^H R_{nn}^{-1} W + \frac{I_{rx}}{\sigma_x} \right) \quad (1)$$

and

$$\hat{F}_{MMSE} = A^{-1} W^H R_{nn}^{-1} y \quad (2)$$

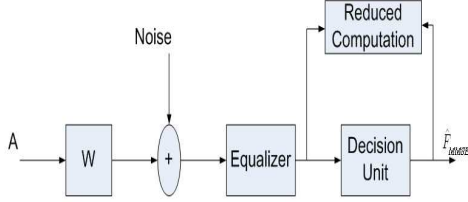


Figure 1. Block diagram of MMSE detector.

3. Matrix Inversion Algorithms

For matrix inversion, the size of the matrix is larger than 2×2 in fixed-point implementation. The fixed-point algorithm has poor stability but floating-point work well and computation time requires fewer cycles like as cofactor, block-wise and systolic array method for matrix inversion operations as bellows:

3.1. Matrix Inversion Using Cofactors Algorithm

Let A be the $n \times n$ matrix, the M_{ij} is the determinant of the sub-matrices of A and C_{ij} is the cofactor of a_{ij} , the

matrix of cofactor formed $A = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$ and

the cofactor $C_{ij} = (-1)^{i+j} M_{ij}$, where i^{th} is the row, j^{th} the column of the matrix A . The adjoint of A is the transpose of the matrix of cofactors and it is denoted by $adj(A)$. If

A is invertible matrix then $A^{-1} = \frac{1}{\det(A)} adj(A)$, for a

MIMO LTE application we have used 4×4 matrix as an example, the inversion of a matrix A can be written as [10], [11]:

$$A^{-1} = \frac{1}{\det(A)} adj(A) = \frac{1}{\det(A)} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{bmatrix} \quad (3)$$

To compute sub-matrices M_{12} where $i=1$ and $j=2$ which will be needed for C_{12} . To compute the determinant of the matrix we get by removing the first row and second column. Here is that work

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \Rightarrow M_{12} \Rightarrow \begin{bmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{bmatrix}$$

Now we can get the cofactor of the matrix A ,

$$C_{12} = (-1)^3 M_{12} = (-1)^3 \begin{bmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{bmatrix}. \text{ The determinant}$$

of sub-matrices can be calculated from its diagonal values. For floating-point implementation, one division is used for computation of $1/\det(A)$.

3.2. Block-wise Matrix Inversion Algorithm

Matrices can be inverted block-wise by using Schur complement. For matrix inversion divides a square matrix into equal small matrices. Let 4×4 matrixes A , it is divided into four 2×2 sub-matrices and inverses by using the following formula [8]:

$$A = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \quad (4)$$

where $P = m \times m$, $Q = m \times n$, $R = n \times m$, $S = n \times n$ matrix and S is the invertible. The inversion each sub-matrices of matrix A as shown in Figure 5 and given by,

$$A^{-1} = \begin{bmatrix} X & -P^{-1}QY^{-1} \\ -Y^{-1}RP^{-1} & Y^{-1} \end{bmatrix} \quad (5)$$

where $X = P^{-1} + P^{-1}QY^{-1}RP^{-1}$ and $Y = S - RP^{-1}Q$.

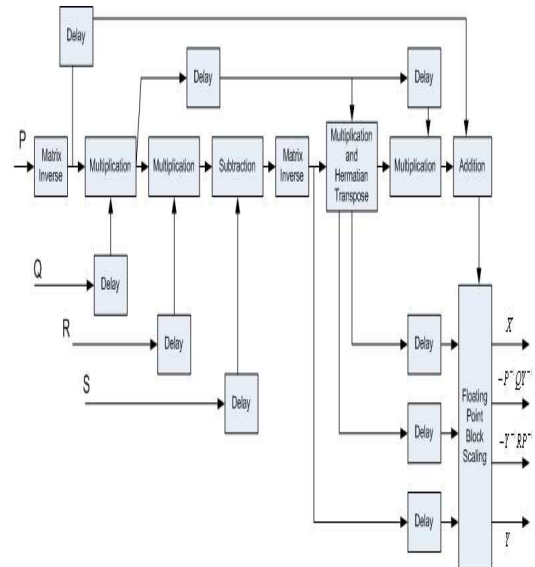


Figure 5. Block-wise matrix inversion algorithm.

The common parts of P^{-1} , RP^{-1} and Y^{-1} contributes to complexity reduction. If P , Q , R and S are all scalars, the inverse of a 2×2 matrix,

$$A^{-1} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}^{-1} = \frac{1}{ps - qr} \begin{pmatrix} s & -q \\ -r & p \end{pmatrix} \quad (6)$$

The matrix A has complex conjugate symmetric values of non-diagonal components for all targets sub-matrices in the MMSE detection and $(ps - qr)$ becomes a real value in this case.

3.3. Systolic Array Process

Systolic array is a computing communication processing which has some advantage like as synchrony, modularity and regularity, spatial and temporal locality, pipelining etc. For computation, it is a simple and regular design, network and balancing computation with input and output. The block diagram of the matrix inversion process and the complete multiplication of sub-matrices using a systolic array process in Figure 2 [2], [3], [6], Figure 3 and Figure 4 [6] the element-wise inversion algorithm for matrix triangulization and it involves local communication between nodes which is suitable for hardware implementation. The triangle has two different blocks. One is real values block and another is imaginary values block. These two parts perform rotations on each element in input matrix as shown in Figure 2. Vector operation on imaginary inputs valued which means it nullifies the imaginary parts of a complex number and outputs the rotation angle to the real parts. Let A is the 4×4 input matrix and I is the identity matrix or the controller of the system as shown in Figure 2. Initial state the operation of input matrix inversion is stored in storage block1. After receiving the first value, imaginary block start processing it. Imaginary block does not have to wait for the first step to finish and these two states can be pipelined. The imaginary block performs vector zing on the values and sends the rotation angle to the real block. The output of the imaginary blocks enters the real blocks for more processing. After processing, all values are stored in storage block 2. The final output inverse of a matrix A is stored in storage block 2. The identity matrix I controls the flow of inputs and outputs to each imaginary & real block and storage blocks of the system. In Figure 4, the tri-array provides upward and rightward communication channels, as opposed to the downward and rightward ones provided in the first phase processing. In Figure 4, if $\theta = 45^\circ$ and

Let,

$$U = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix},$$

We can express by [11]:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} \begin{bmatrix} e^{i\Delta\theta} & 0 \\ 0 & e^{-i\Delta\theta} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} = U \Lambda U^{-1} \quad (7)$$

where Λ is diagonal and phase shift matrices.

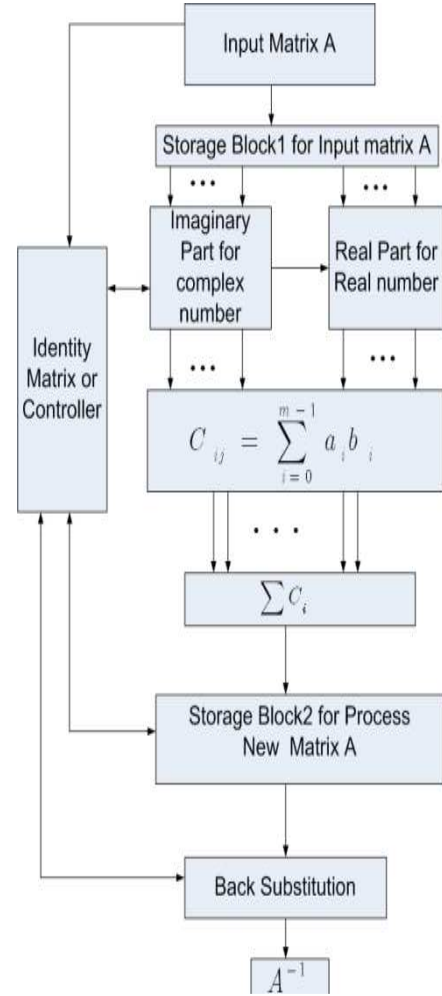


Figure 2. Element-wise matrix inversion.

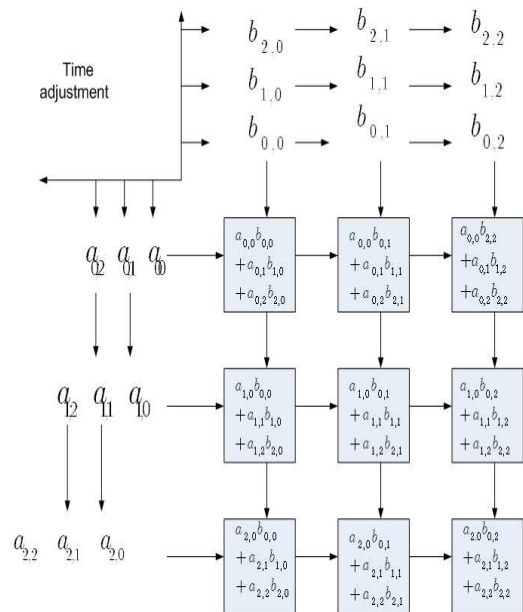


Figure 3. Sub-matrices (3*3) multiplication using a systolic array algorithm.

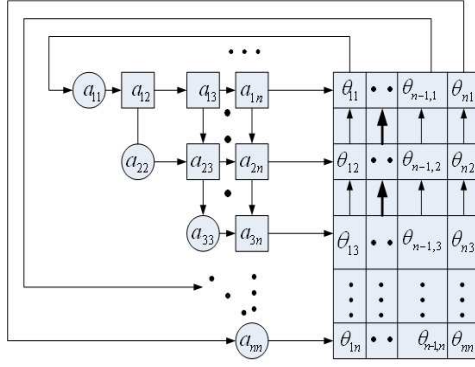


Figure 4. The Parameters of the rotation angles $\{\theta_{ij}\}$ emerge from the right side and data buffer to store these parameters with the top -row connected to the diagonal processing elements of the tri-array.

4. Polar Decomposition with Matrix Inversion

If A is non-singular square $n \times n$ matrices, $A \in \mathbb{C}^{n \times n}$ this algorithm computes the polar decomposition,

$$A = QU = VQ \quad (8)$$

where Q is the orthogonal matrix, U is the symmetric positive definite matrix and right stretches. The forward (upper) operation of the polar decomposition in A ,

$$U^2 = A^* A \quad (9)$$

where $(\cdot)^*$ denotes transpose conjugate. Similarly the reverse (lower) case,

$$A = VQ \quad (10)$$

where $V^2 = AA^*$ where V is symmetric and left stretches. The Principal rotation matrix as given by

$$Q = U^{-1}A = AV^{-1} \quad (11)$$

where $U = Q^*A = Q^*VQ$ and $V = AQ^* = QUQ^*$. If A is nonsingular non-square ($m \times n$) matrices and $A \in \mathbb{C}^{m \times n}$ this can first calculate QR decomposition [7],

$$A = QR \quad (12)$$

where $Q \in \mathbb{C}^{m \times n}$ has orthogonal column, R is upper triangular and nonsingular. The polar decomposition of A is given in term of that of R by

$$A = QR = Q(U_R V_R) = (QU_R) V_R = UV \quad (13)$$

Let Sub-matrices $M_{ij} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ of matrix A , then a factor that makes unit columns vectors

$$Q = M_{ij} + \text{sign det}(M_{ij}) \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}. \text{The Polar decomposition}$$

factor Q is the closest possible to matrix M_{ij} , with norm

measured using Frobeniusnorm $\|Q - M_{ij}\|_F^2$. We can express $\|M_{ij}\|_F^2$ as the diagonal sum and trace of M^*M using a symmetric Lagrange multiplier matrix r .

$$\text{trace}[(Q - M_{ij})^T (Q - M_{ij}) + (Q^T Q - I)r] \quad (14)$$

We can differentiate with respect to Q and express by

$$M_{ij} = Q(I + r) = QU \quad (15)$$

where $(I + r)$ is the forward polar symmetric positive upper triangular and right stretches matrix. Similarly we can write,

$$M_{ij} = (I - r)Q = VQ \quad (16)$$

where $(I - r)$ is the reverse polar symmetric lower triangular and left stretches matrix.

This factorization is the polar decomposition of M_{ij} . We can write

$$(MU^{-1})^* (MU^{-1}) = I \quad (17)$$

where $Q^T Q = I$ and symmetric inverse is given by

$$U^{-1} M^* M U^{-1} = I \quad (18)$$

$$\therefore U^2 = M^* M$$

Similarly we can write by equation (17)

$$V^2 = MM^* \quad (19)$$

From equation (2) matrix inversion of A to get an MMSE solution using lower and upper triangular, forward polar and reverse polar operations as below:

$$\hat{F}_{MMSE} = A^{-1} \hat{y} \quad (20)$$

where $\hat{y} = W^H R_m^{-1} y$.

For lower triangular case:

$$\hat{y} = V^2 \hat{F}_{MMSE} = MM^* \hat{F}_{MMSE} \quad (21)$$

and for upper triangular case:

$$\hat{y} = \hat{F}_{MMSE} U^2 = \hat{F}_{MMSE} M^* M \quad (22)$$

5. Implementation and Simulation Results

Matrix A is a 4×4 matrix of complex, floating point values and output is the inverse matrix. In Figure 5, the sub-matrix $-P^{-1}QY^{-1}$ can be computed from the hermitian transpose of $-Y^{-1}RP^{-1}$. Matrix A has four 2×2 sub-matrix arithmetic units. Matrix A has four 2×2 sub-matrix

arithmetic units. These units are to be implemented using pipeline circuit in Figure 5 and Figure 6. The number of computation depends on the number antennas L and the cost of MMSE equalizer for 4×4 matrix inversion as follows [4]:

$$f_{flops} = 10L^3 + 5L^2 + L \quad (23)$$

The noise covariance matrix R_{nn}^{-1} in equation (1) is a diagonal matrix and all diagonal elements are equal. We assume equal noise variance and additive white noise for all receive antennas. In Figure 6, the inverse channel matrix:

$$W_{sc_k}^{-1} = (W_{sc_k}^H W_{sc_k} + \sigma^2 I)^{-1} W_{sc_k}^H \quad (24)$$

The target matrix $T_{sc} = W_{sc_k}^H W_{sc_k} + \sigma^2 I$ and the output of the equalizer:

$$\hat{F}_{MMSE}(t) = W_{sc_k}^{-1} y_{sc_k}(t) = T_{sc_k}^{-1} W_{sc_k}^H y_{sc_k}(t) = A^{-1} W_{sc_k}^H y_{sc_k}(t) \quad (25)$$

where $T_{sc_k}^{-1} = A^{-1}$ and sc_k is the number of sub-carrier and t the number of step for MIMO receiver. The receiver complexity $O\{(E)^L\}$ where L is the number of receiver antennas and the processing elements E . In Figure 6, the block floating scaling factors are settled to decrease the finite length errors in the 2×2 sub-matrix multiplication and inverse units. Complex floating point operations (flops) scaling for each scalar complex addition count one flop and each complex multiplication count three flops. For square matrices $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times n}$, $C \in \mathbb{C}^{n \times n}$ and complex vector $\hat{d} \in \mathbb{C}^{n \times 1}$, the computational cost of the matrix operation, number of flops and useful vector counts between QR and polar decompositions are as shown in Table 1. Simulation results from the matrix inversion algorithms using polar decomposition, floating point operation (25) we tested as shown in Figure 7 and Figure 8. The BER performance on without channel matrix inversion is compared to that of with channel matrix inversion algorithm for 4×4 case in which four receivers with the highest norm values are selected out of number of users has been shown in Figure 9. It can be mitigated noise enhancement very fast.

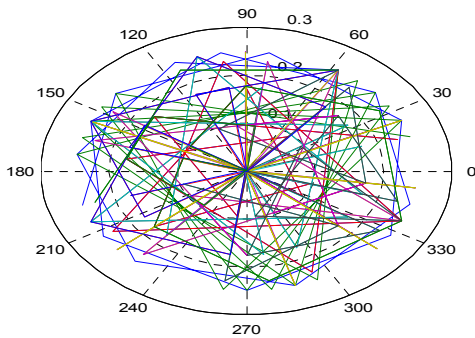


Figure 7. Matrix inversion using polar decomposition (matrix size 16x16).

Table 1. Compare to QR and Polar Decomposition.

Matrix Operation	Number of Flops (f_{flops})	
	QR [4], [1] Decomposition	Polar Decomposition
A+B	n^2	n^2
A*B	$2n^3/3$	$n^3/3$
A*A ^H	$(4n^3 + n^2 - n)/2$	$(N - (1/6))n^3$ where $N = \text{no. of iterations}$ $A = MM^*$; $M^* = OA^{\frac{1}{2}}$
C ⁻¹	$4n^3 + n^2 + n$	$2n^3 + n^2 - n$
A*a	$4n^2 - n$	$2n^2 - n$

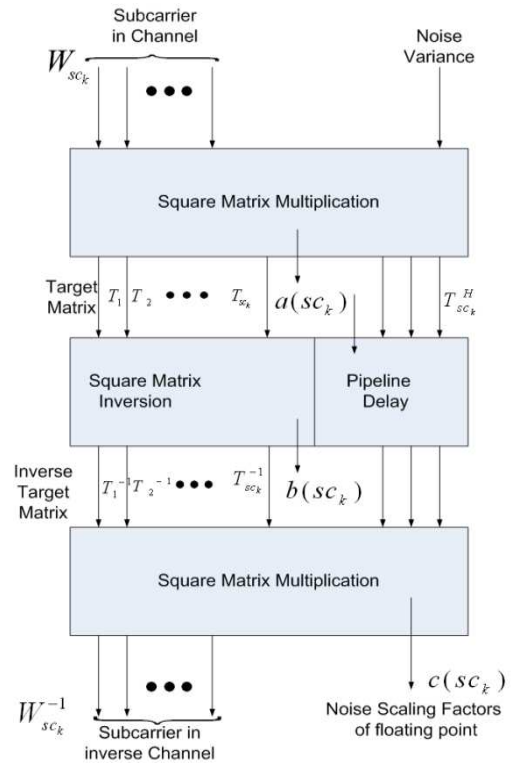


Figure 6. Block diagram of transmission architecture.

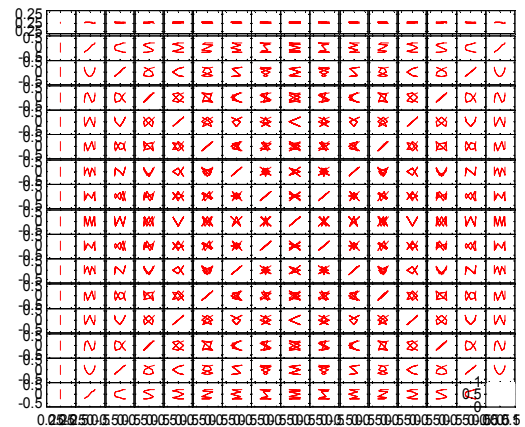


Figure 8. 16x16 channel matrix inversion.

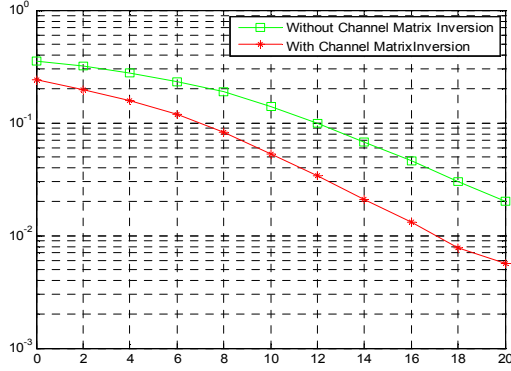


Figure 9. BER performance using matrix inversion algorithm.

6. Conclusions

Matrix inversion is designed and implemented on various types of DSP's like as Xilinx vertex 5 using cofactors, block-wise, systolic array and polar decomposition methods in floating point. All 32 bits are real and imaginary precision. One sign bit and 8 exponents bit. Word length (l) of 4×4 the matrix has a floating-point C block-wise cycles $l + A^{-1}y_{sc_k}(t)$ and floating point cofactor $l + A^{-1}y_{sc_k}(t)$ and block-wise methods save about 3 to 4 times the fixed-point polar decomposition based inversion in terms of cycle counts. Polar decomposition factor (8) QU which are unique, independent coordinates, both efficient and easy to compute. So our proposed method increasing the matrix inversion processor clock speed and reduces the receiver computational module cost.

Acknowledgements

This work was supported by World Class University-R32-2010-000-20014-0, the NRF, BSRP-2010-0020942, NRF, BK21 and MEST-2012-002521, NRF, Republic of Korea.

References

- [1] M. Myllyla, J. Hintikka, J. Cavallaro, M. Juntti, M. Limingoja, and A. Byman, "Complexity Analysis of MMSE Detector Architecture for MIMO-OFDM System," in Proceedings of the 2005 Asilomar Conference, Pacific Grove, CA, Oct 30-Nov 2 2005.
- [2] "www.Altera.com", "www.Xilinx.com".
- [3] H. K. W. M Gentleman, "Matrix Triangularization by Systolic Arrays", Real-Time Signal Processing, vol.298, pp.19-26, 1981.
- [4] Mohamed Chouayakh, Andress Knopp, and Berthold Lankl, "Low Complexity Two Stage Detection Scheme for MIMO Systems" IEEE Information Theory for Wireless Network, 1-4244-1200-5/07/2007.
- [5] Di Wu, Johan Eilert and Dake Liu, Dandan Wang, Naofal Al-Dhahir and Hlaing Minn, "Fast Complex Valued Matrix Inversion for Multi-User STBC MIMO Decoding" VLSI, 2007, ISVLSI '07, IEEE Computer Society Annual Symposium, pp.325-330, March 2007.
- [6] Marjan Karkooti, Josep R. Cavallaro, Chris Dick, "FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm," 1-4244-0132-1/2005 IEEE.
- [7] Luo Jianwen, Jong Ching Chuen, "Matrix Inversion on Reconfigurable Hardware Using Binary -Coded Z-Path COR-DIC," APCCAS 2006.
- [8] Shodai Yokoyama, Kazuya Matsumoto, and Stanislar G. Sedukhin, "Matrix Inversion on the Cell/B.E. Process," 11th IEEE International Conference on High Performance Computing and Communication, 2009.
- [9] Moon Ho Lee, Sundar Rajan, J.Y. Park, "A Generalized Reverse Jacket Transform," IEEE Transactions on Circuit Systems-II: Analog and Digital Signal Processing, Vol.48 No.7, July 2001.
- [10] Moon Ho Lee, "A New Reverse Jacket Transform and Its Fast Algorithm," IEEE Transactions on Circuit Systems-II: Analog and Digital Signal Processing, Vol.47 No.1, July 2000.