A MATRIX INVERSION HARDWARE ARCHITECTURE BASED ON GAUSS-JORDAN ELIMINATION FOR MIMO APPLICATIONS.

HAMMAM ORABI

A project report submitted in partial fulfilment of the

requirements for the award of the degree of

Master of Engineering (Electrical-Electronics and Telecommunications)

Faculty of Electrical Engineering

Universiti Teknologi Malaysia

JUNE 2014

To Father, Mother, Brothers, Sisters,

Sidra,

Friends,

& to the Falling Leaves of Syria.

ACKNOWLEDGEMENT

I would like to start by expressing my sincere appreciation to my supervisor, Dr. Rabia Bakhteri who did not withhold any effort in supporting me by help, advice, and guidance throughout both of the project's phases; Research Project Proposal and Project Thesis.

I would like to extend my gratitude to UTM's FKE staff and lecturers for providing me with enthusiasm as well as knowledge, of which I needed both to finish this work.

ABSTRACT

Digital Communications require faster and more dedicated electronic platforms to satisfy its real time performance. Highly computational algorithmic kernels can create a bottleneck to the systems where it is implemented on, due to the time it take to provide outputs, which reduces the throughput of the overall communication system. Fully Dedicated Hardware Architectures provide solutions to the speed crisis, since they provide the ability of performing parallel computations. Matrix Inversion is a complex algorithm that is implemented in MIMO systems. Since software solutions are not fast enough to satisfy the speed requirements, hardware solutions stood up to the challenge. In this project, a fully dedicated hardware architecture to find matrix inversion using Gauss-Jordan Elimination algorithm is presented, complex arithmetic operation is performed to match the nature of the elements of estimated propagation matrix in MIMO systems. The design is parameterized and universal for any complex matrix. Developing the design was done using SystemVerilog HDL, while simulation was done through Altera ModelSim. Software kernels were done on MATLAB for comparison purposes. The proposed architecture performance shows its advantage over software based kernels. Further comparisons have been done with previous designs for a variety of matrix sizes.

ABSTRAK

Komunikasi digital memerlukan platform elektronik yang lebih cepat dan lebih khusus untuk memenuhi prestasi masa sebenar. Kernel algoritma sangat pengiraan boleh membuat kesesakan kepada sistem di mana ia dilaksanakan pada , kerana masa yang diambil untuk menyediakan output, yang mengurangkan daya pemprosesan sistem komunikasi keseluruhan. Perkakasan Seni Bina sepenuhnya Dedicated menyediakan penyelesaian kepada krisis kelajuan, kerana mereka memberikan keupayaan melaksanakan pengiraan selari. Matrix penyongsangan adalah algoritma kompleks yang dilaksanakan dalam sistem MIMO. Sejak penyelesaian perisian tidak cukup pantas untuk memenuhi keperluan kelajuan, penyelesaian perkakasan menyahut cabaran itu. Dalam projek ini, seni bina perkakasan berdedikasi sepenuhnya untuk mencari matriks penyongsangan menggunakan algoritma Gauss -Jordan Penghapusan dibentangkan , operasi aritmetik kompleks dijalankan untuk menyesuaikan unsur unsur-unsur dianggarkan matriks perambatan dalam sistem MIMO. Reka bentuk adalah parameterized dan sejagat untuk sebarang matriks kompleks. Membangunkan reka bentuk telah dilakukan dengan menggunakan SystemVerilog HDL. Kernel perisian telah dijalankan ke atas MATLAB untuk tujuan perbandingan. Prestasi seni bina yang dicadangkan menunjukkan kelebihan ke atas kernel perisian berasaskan. Perbandingan selanjutnya telah dilakukan dengan reka bentuk sebelum ini untuk pelbagai saiz matriks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

G-J	Gauss-Jordan
HA	Hardware Architecture
HA.C.A	Hardware Architecture for Complex Numbers optimized for Area
HA.R.A	Hardware Architecture for Real Numbers optimized for Area
HA.R.S	Hardware Architecture for Real Numbers optimized for Speed
HA.R.S	Hardware Architecture for Complex Numbers optimized for Speed
ICI	Inter Carrier Interference
ISI	Inter Symbol Interference
MIMO	Multiple input multiple output
OFDM	Orthogonal Frequency Division Multiplexing
H	Complex Propagation Matrix
Nt	Number of transmitted signals
Nt	Number of received signals
W	Zero mean complex Additive White Gaussian Noise
X	Transmitted symbols
Y	Received vector from all the Nt receiver's antennas

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Orthogonal Frequency Division Multiplexing (OFDM) is becoming a very popular multi-carrier modulation technique for transmission of signals over wireless channels. OFDM divides the high-rate stream into parallel lower rate data and hence prolongs the symbol duration, thus helping to eliminate Inter Symbol Interference (ISI). It also allows the bandwidth of subcarriers to overlap without Inter Carrier Interference (ICI) as long as the modulated carriers are orthogonal. OFDM therefore is considered as an efficient modulation technique for broadband access in a very dispersive environment.

Recently, high data rate and strong reliability in wireless communication systems are becoming the dominant factors for a successful exploitation of commercial networks. MIMO-OFDM (multiple input multiple output orthogonal frequency division multiplexing), a new wireless broadband technology, has gained great popularity for its capability of high rate transmission and its robustness against multi-path fading and other channel impairments. The block diagram of MIMO-OFDM transmitter and receiver is shown in Figure 1.1 and 1.2 respectively.

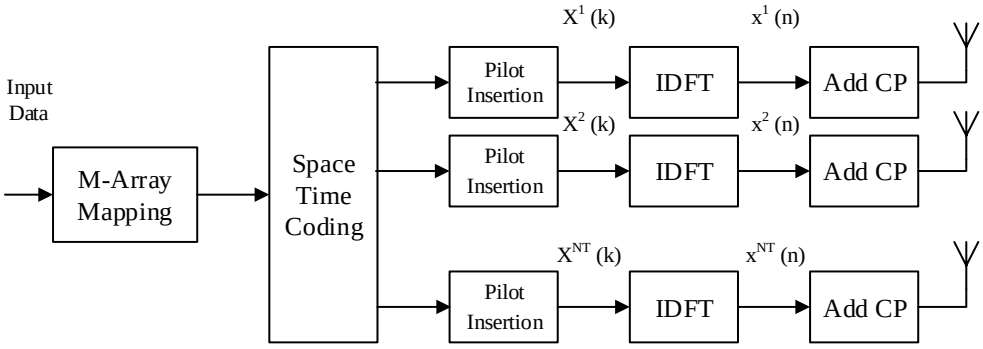

Figure1.1 MIMO-OFDM Transmitter

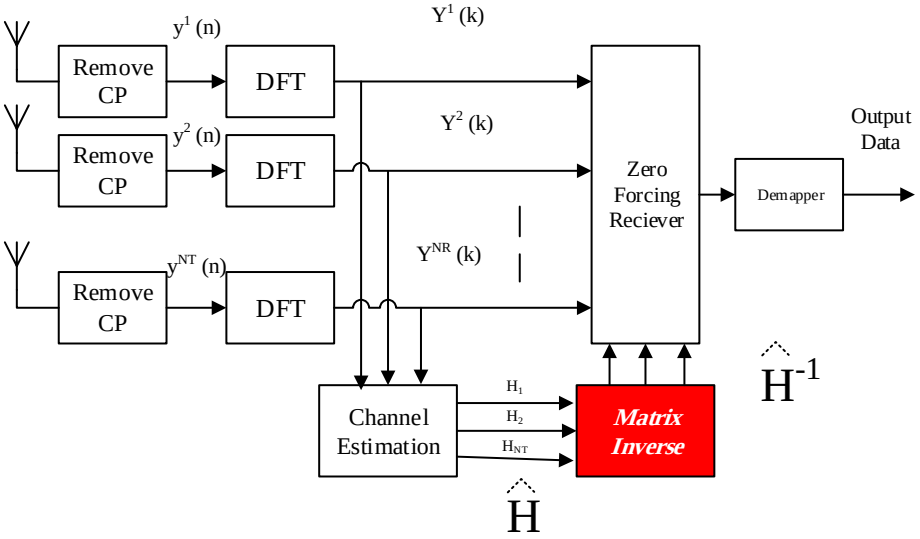

Figure1.2 MIMO-OFDM Reciever

As a MIMO signaling technique, Nt different signals are transmitted simultaneously over $Nt \times Nr$ transmission paths, and each of those Nr received signals is a combination of all the Nt transmitted signals and the distorting noise. The received signal will be the convolution of the channel and the transmitted signal. After removing the cyclic prefix at the receiver's side, the output of FFT module, as the demodulated received signal, can be expressed in the matrix form equation as following (assuming that the channel is static during an OFDM block):

$$Y = HX + W \quad (1)$$

Equation (1) can be further elaborated in matrix form as equation (2).

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{Nr} \end{bmatrix} = \begin{bmatrix} H_{1,1} & H_{1,1} & \cdots & H_{1,Nr} \\ H_{2,1} & H_{1,1} & \cdots & H_{1,Nr} \\ H_{3,1} & H_{1,1} & \cdots & H_{1,Nr} \\ \vdots & \vdots & \ddots & \vdots \\ H_{Nr,1} & H_{Nr,1} & \cdots & H_{1,Nr} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_{Nt} \end{bmatrix} + \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_{Nt} \end{bmatrix} \quad (2)$$

Where Y is the received vector from all the Nt receiver's antennas, H is the channel transform function (Complex Propagation Matrix), X represents the transmitted symbols from all the transmitting antennas for subcarrier k, W represents zero mean complex Additive White Gaussian Noise. The transfer function of the matrix-valued channel impulse response is given by

$$H\left(e^{j2\pi\theta}\right) = \sum_{l=0}^{L-1} H_l \, e^{-j2\pi l\theta}, \qquad 0 \leq \theta < 1 \quad (3)$$

The data symbols $\hat{X}$ are then estimated by linear detection algorithm such as Zero Forcing and is given by:

$$\hat{X} = H^{-1}Y \quad (4)$$

While (4) has to be evaluated at the symbol rate, the channel inverses H^{-1}can be pre-computed and have to be updated only when the channel changes (Borgmann, and Bölcskei, 2004).

From the numerous matrix inversion algorithms that exist, Gauss-Jordan Algorithm (G-J) is characterized as a simple, efficient, direct, parallelizable, and universal algorithm to find the inverse of any kind of square matrices (Duarte *et al.*, 2009). Although its higher complexity ($O(n^3)$) in contrast to other software algorithms (e.g. Strassen ($O(n^{2.807})$) and Coppersmith-Winograd ($O(n^{2.376})$), G-J is very important for developing practical architectural implementations, since the bottleneck of Von-Neumann architecture can be avoided (Jacobi *et al.*, 2011).

G-J Elimination requires simple arithmetic operations only i.e. Addition/Subtraction, Multiplication, and Division. In comparison, other numerical matrix inversion methods require more complicated operations, such as square root operation as in QR decomposition by Gram-Schmidt Orthogonalization, and sine and cosine operations as in QR decomposition by Rotation (Pozrikidis, 2008).

1.2 Problem Statement

Matrix Inversion is one of the most costly computational operations to be performed either in software or hardware (Hanzo *et al*., 2010). It has vast implementations in many of nowadays technologies, especially in MIMO systems such as OFDM MIMO (Moussa *et al.*, 2013) and Long-Term Evolution (LTE) MIMO receivers to remove the effect of the channel on the received signal (Yan *et al.*, 2010). It can be used as a pre-coding stage in OFDM MIMO in order to cancel the interference at the Base station (Cho *et al.*, 2010).

This operation is significant in MIMO systems because of its good performance in high data rates applications, as it is a straight forward concept without iterations (Haustein *et al.* 2002). Software based matrix inversion modules suffer from long latency because of the complicated decoding of the executed instruction, and because of Von-Neumann bottleneck, which is a result of sharing the same memory for data and instructions. Figure 1.3 depicts the time required to perform matrix inversion on matrices of different sizes. The long latency time (0.02 second for a 36x36 matrix) makes the utilization of this technique less desirable in real time application.

In IEEE 802.11a-1999, fifty two OFDM subcarriers are used. Of the fifty two subcarriers, 48 are for data and 4 are for pilot subcarriers. Performing Matrix Inversion

in a system that uses such large number of subcarriers is a critical challenge of optimizing the design for speed and cost (Moussa *et al.*, 2013).

On the other hand, a regular (direct) method of matrix inversion such as Cramer's require evaluation of $(N + 1) \times N \times N$ determents. So for a 10x10 system, 359 million multiplications have to be performed (Cho *et al.*, 2010). For a 20x20 system the number of multiplications is factorial(21). A fast computer will do the task in 1600 years (Turner, 2000).

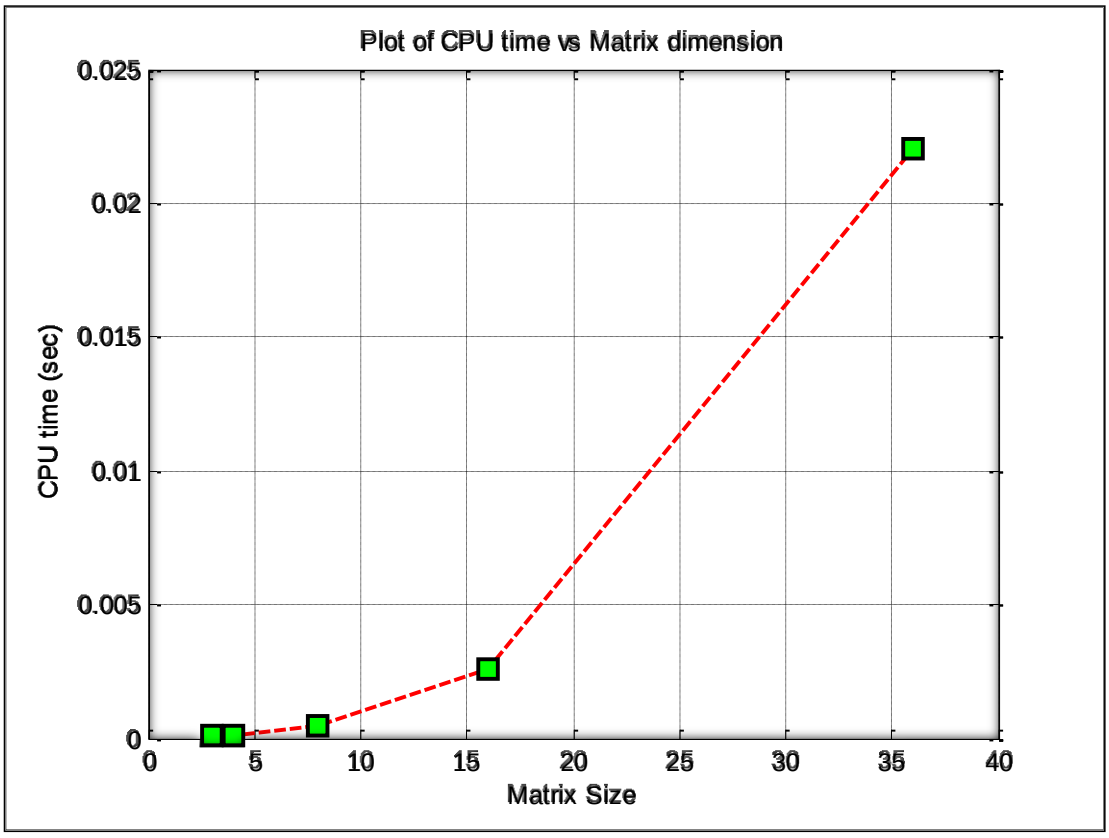

Figure 1.3 CPU time vs matrix's size

1.3 Objective

The objective of this project is to build an accelerated, scalable, and low area cost hardware architecture that performs matrix inversion for complex numbers, using the Gauss–Jordan method. The architecture should accomplish the complex inversion in a better time than software based approaches. It should be able to operate on any matrix regardless of its size. Finally, it should achieve better area consumption in comparision to some other hardware architectures.

1.4 Scope

The proposed system is responsible for finding the inverse of the square complex propagation matrix in MIMO systems. Numerical computation methods are addressed and compared to both numerical and regular methods.

Hardware Description Languages are the medium of implementing these algorithms, such as Verilog and SystemVerilog.

The following software are used in this project:

- MATLAB 2013 is used for benchmarking and for results verification.
- Quartus II is used for modelling the system in Verilog and SystemVerilog.
- ModelSIM Altera is used for simulating the design and for acquiring performance statistics.

The Field Programmable Logic Array (FPGA) platform is Altera DE2-115 Development and Education Board, powered by Altera Cyclone IV E FPGA.

Single precision floating point (IEEE 745 standard) Complex Addition/Subtraction, Multiplication and Division operations are used in the Inversion process; this is necessary since the estimated matrix has complex values.

1.5 Project Overview

The top level functional block diagram is shown in Figure (1.4). The system will receive the complex matrix entries and find the inverse upon setting start signal high. The results will be ready after the done signal is set, while the fail signal indicates that the input matrix is a singular matrix, thus it has no inverse.

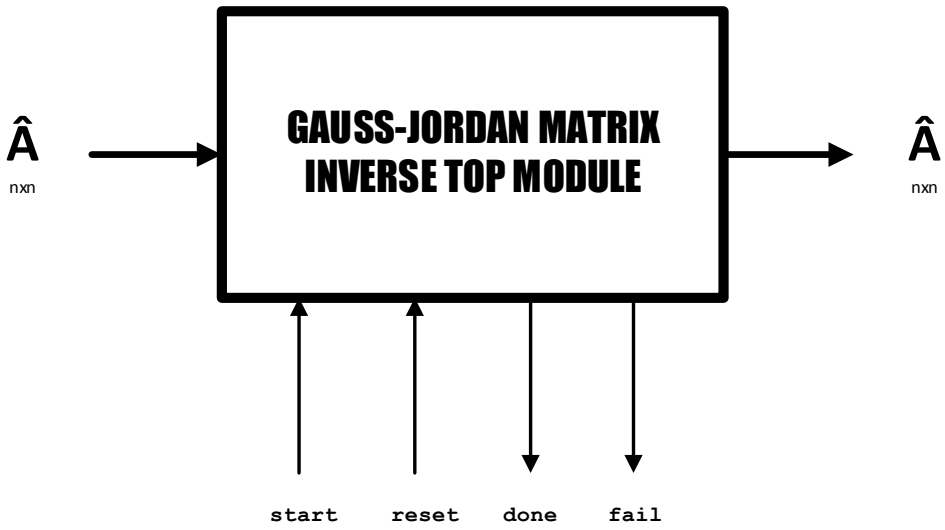

Figure 1.4 Top level module of G-J

The design is made up of two basic modules, the Control Unit (CU) and the Data-path Unit (DU). CU has the main Finite State Machine (FSM) controller, and controls the sequence of the operations and processes of G-J. The DU contains the

basic modules necessary for G-J, named: Pivoting, Normalization, Elimination, Storage, Memory Address Generator (MAG), and several counters designated for generating the indices of the matrix's elements. Figure (1.5) shows the CU and DU organization of G-J top level module.

A tradeoff between the area cost and speed will be done, resulting in 4 different designs, 2 for each real and complex plane, each design is optimized either for speed or area requirements.

The 4 different developed designs performance will be evaluated: Real (optimized for area and speed) and Complex (optimized for area and speed). The aspects of evaluation are the number of consumed clock cycles to finish the operation, latency, Look Up Tables (LUTs) consumption, M9K memory unit utilization, and RAM usage.

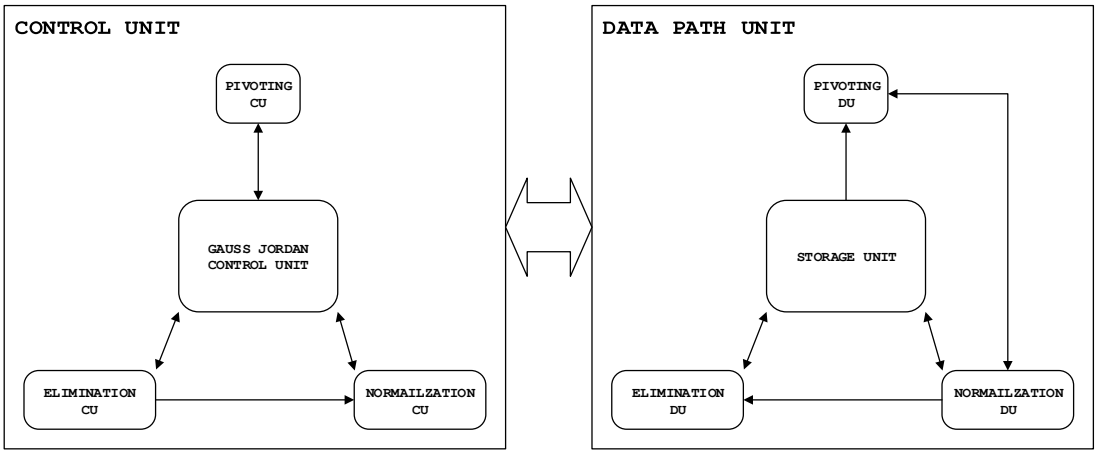

Figure (1.5) CU and DU of G-J

1.6 Thesis Structure

This thesis consists of five chapters. Chapter one has described the background of the project, problem statement, objectives, scope and project overview. Chapter two describes the theory and background of Gauss-Jordan Elimination method as well as previous works done in this field. Chapter three explains the research methodology used in this thesis including the system architecture. Chapter four shows the experimental results followed by analysis and discussion. Chapter five states the conclusion and the possible future improvements. Finally, Appendices A, B, and C contains the MATLAB, C, and SystemVerilog source code of G-J algorithm respectively.

REFERENCES

Bagadi, K. Y., & Das, S. (2010). MIMO-OFDM Channel Estimation using Pilot Carries. *International Journal of Computer Applications (0975 – 8887) Volume 2 – No.3, May 2010*

Borgmann, M. and Bölcskei, H. (2004). Interpolation-Based Efficient Matrix Inversion for MIMO-OFDM Receivers. *Communication Technology Laboratory, Swiss Federal Institute of Technology*

Cho, Y. S., Kim, J., Yang, W. Y. and Kang, C. G. (2010). *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons, Ltd, Chichester, UK. doi: 10.1002/9780470825631.refs

Duarte, R., Neto, H. and Vestias, M. (2009). Double-precision gauss-jordan algorithm with partial pivoting on FPGAs. *12th Euromicro Conference* on, aug. 2009, pp. 273-280.

Gene H. Golub and Charles F. Van Loan (1996). *Matrix computations.* (3rd Edition). Johns Hopkins University Press Baltimore, MD, USA.

Hanzo, L., Akhtman, Y., Wang, L. and Jiang, M. (2010) Bibliography, in MIMO-OFDM for LTE, Wi-Fi and WiMAX, John Wiley & Sons, Ltd, Chichester, UK. doi: 10.1002/9780470711750.biblio

Haustein, T., Helmolt, C., Jorswieck, E., Jungnickel, V. and Pohl, V. (2002). Performance of MIMO systems with channel inversion. in Proc. *IEEE Vehicular Technology Conference*, VTC Spring, Birmingham, AL, May 2002, pp. 35–39.

Irturk, A., Benson, B., Mirzaei, S. and Kastner, R. (2008). An FPGA Design Space Exploration Tool for Matrix Inversion Architectures. *SASP 2008*: 42-47.

Jacobi, R., Arias-Garcia, J., Llanos, C. and Ayala-Rincon, M. (2011). A suitable FPGA implementation of floating-point matrix inversion based on gauss-jordan elimination. *VII Southern Conference* on, April 2011, pp. 263-268.

Jacobi, R., Arias-Garcîa, J., P., Llanos, C. H., & Ayala-Rincön, M. (2011). A Suitable FPGA Implementation Of Floating-Point Matrix Inversion Based On Gauss-Jordan Elimination.

McIlhenny, R.and Ercegovac, M. (2005). On the Design of an On-line Complex Matrix Inversion Unit. *Proc. 39th Asilomar Conference on Signals, Systems and Computers*, 5pps., 2005.

Moussa, S., Abdel Razik, A. M., Dahmane, A. O., & Hamam, H. (2013) FPGA Implementation Of Floating-Point Complex Matrix Inversion Based On Gauss-Jordan Elimination (2013). *26th IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE)*

Pozrikidis, C. (2008). *Numerical computation in science and engineering*. Oxford: Oxford University Press.

Senthilvelan, M. ; Iancu, D. ; Glossner, J. ; Moudgill, M. and Schulte, M. (2007). Software Solutions for Converting a MIMOOFDM Channel into MultipleSISO-OFDM Channels. *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2007. WiMOB 2007.

Turner, Peter R. (2000.). *Guide 2 Scientific Computing*. CRC Press, 2000.

Yan, M., Feng, B. and Song, T. (2010). On Matrix Inversion for LTE MIMO Applications Using Texas Instruments Floating Point DSP. *Signal Processing (ICSP),* 2010 IEEE 10th International Conference on.