



UNIVERSITY OF BATNA

Semiconductor parameter extraction using EBIC and Cathodoluminescence

by

Souhila SOUALMIA

**A dissertation submitted for the degree of
Doctor of Science**

Department of material science, College of Science, University of Batna

July 2012

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieure et de la Recherche Scientifique
Université de Batna

Thèse

Présentée au Département de science de la matière
Faculté de science

Pour l'obtention du diplôme de
Docteur en Science
Option : physique des matériaux

par
Souhila SOUALMIA

Thème
Extraction des Paramètres de Semiconducteur en
Utilisant les Méthodes EBIC et Cathodoluminescence

Soutenue le : Juillet 2012

Devant la commission d'examen composée de:

M.Chahdi	Professeur	Université de Batna	Président
A.Bouldjedri	Professeur	Université de Batna	Rapporteur
A.Benhaya	Professeur	Université de Batna	Rapporteur
A.Bouabellou	Professeur	Université de Constantine	Examineur
K.Bouamama	Professeur	Université de Sétif	Examineur
Z.Ounenoughi	Professeur	Université de Sétif	Examineur
Z.Ahmida	M.C	Université de Skikda	Examineur

Dedication

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ " هَذَا مِنْ فَضْلِ رَبِّي لِيَبْلُوَنِي أَأَشْكُرُ أَمْ أَكْفُرُ وَمَنْ شَكَرَ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ وَمَنْ كَفَرَ فَإِنَّ رَبِّي غَنِيٌّ كَرِيمٌ " (النمل 40)

I dedicate this work to my dearest husband; without his constant help, close interactions and encouragement this work would not have been possible.

To my mother and the pure spirit of my dear father may the rahma of Allah be upon him.

To my dearest lovely daughter honey Amira, to all my family, my husband's family, and all my friends every where in the world.

Acknowledgement

First of all, my great thankfulness and praise to Allah; the Most Beneficent, the Most Merciful for the innumerable bounties and blessings.

I express my gratitude to my advisors Pr Abdelhamid Bouldjedri and Pr Abdelhamid Benhaya, for there support and guidance throughout the different phases of this dissertation.

I would like to greatly thank Pr Mohamed Chahdi President of the dissertation committee, and all the members of the dissertation committee: Pr A. Bouabellou , Pr K.Bouamama, Pr Z.Ouennoughi, and Dr Z.Ahmida ; For the time and the interest they gave to my work.

My warmest thanks are reserved for the joy of my life, my best friend; the one who can always bring back a smile on face, my husband, Dr Abdelouahab Bentrchia. I am highly indebted to him, for the constant help, support and encouragement. I appreciate his great effort in improving my writing skills and in revising my papers. I would like to highly thank him for his helpful guidance in my academic research and also in building my future.

I do not forget to highly thank Prof. Nouar Thebet from the KFUPM University for introducing me to a new field, for all his great guidance and help.

Special thanks are due to my friends: Salima Messadi and Afaf Djaraoui for their constant help in Administrative aspects within the University of Batna.

Most importantly, I would like to express my gratitude to my family especially my parents for there love and continuous prayers. My regards are also due to the rest of my family for their love and encouragement, and my husband's family.

Finally, I do not forget to thank my honey daughter Amira for the happiness and joy she added in my life. Thank you my love!!!

Souhila

Abstract

The performance and functionality of semiconductor devices is directly affected by the transport properties of carriers, as such many techniques for the extraction of semiconductor parameters (i.e. the absorption coefficient α , the diffusion length L , the dead layer thickness Zt , the surface recombination velocity S , and the relative quantum efficiency Q) using the cathodoluminescence/EBIC in the Scanning Electron Microscopy (SEM) were developed. In this work we develop novel and effective approaches for the extraction of these semiconductor parameters directly from any theoretical/experimental steady state cathodoluminescence (CL) signal/ Electron Beam Induced Current (EBIC). Our extraction techniques based on Artificial Neural Networks ANN /genetic algorithms (GA) allow us to obtain simultaneously near – optimum values for the semiconductor parameters. Compared to other techniques in the literature, our approaches are found to be efficient and successful.

La performance et la fonctionnalité des dispositifs semi-conducteurs est directement affecté par les propriétés de transport des minoritaires, de nombreuses techniques pour l'extraction de ces paramètres de semi-conducteurs (tels que: le coefficient d'absorption α , la longueur de diffusion L , l'épaisseur de la couche isolatrice Zt , la vitesse de recombinaison de surface S , et le facteur de rendement quantique relative Q) pour les méthodes cathodoluminescence/ EBIC de Microscopie Electronique à Balayage (SEM) ont été développés. Dans ce travail, nous développons de nouvelles approches d'extraction de ces paramètres directement des signaux cathodoluminescence(CL) / Courant induit de faisceau électronique (EBIC) . Nos techniques d'extraction baser sur les réseaux artificiels de neurones ANN / algorithmes génétiques (GA) permettent l'obtention

simultanée des valeurs proche de l'optimale des paramètres de semi-conducteurs. Comparant avec des techniques trouvées dans la littérature, nos approches sont efficace et très réussites.

ان نوعية اجهزة اشباه النواقل و فاعليتها تعتمد على خصائص النقل لدى حاملات الشحنة ذات الاقلية , طرق ونماذج عديدة وكثيرة تم انجازها لاستخراج قيم معاملات اشباه النواقل (على سبيل المثال: معامل الامتصاص, طول الانتقال الحر, سماكة الطبقة العازلة, سرعة الاندماج السطحية, معامل الارجاع الكمي النسبي) باستخدام الاضاءة الكاثودية المحفزة التيار المحفز بالحرارة الكاثودية لميكروسكوب المسح الالكتروني . قمنا في هذا العمل بانجاز طرق جديدة لاستخراج هذه المعاملات مباشرة من اي اضاءة كاثودية تيار مستحث من الحزمة الالكترونية. طرقنا الجديدة و المطورة التي تعتمد اساسا على :خوارزمات الشبكات العصبية الاصطناعية, الخوارزمات الجينية تقترب من الحصول على القيم المثلى لهذه المعاملات. مقارنة بالطرق الموجودة حاليا اثبتت طرقنا نجاعتها و فاعليتها من خلال النتائج المحصل عليها.

Table of Contents

Dedication	
Acknowledgement	
Abstract	iii
Table of Contents	v
List of Figures	ix
List of Tables	xii
Chapter 1 Introduction	14
1.1 Background	12
1.2 Aim of this Work	14
1.3 Thesis Organization	15
1.4 Thesis Contribution	17
Chapter references	18
Chapter 2 Scanning Electron Microscopy fundamentals	23
2.1 Introduction	23
2.2 Interaction of the incident electrons with the irradiated matter	24
2.2.1 Physical process	24
2.2.2 function of energy dissipation	26
2.2.3 electron range	27
2.2.4 generation of pairs	28
2.2.5 Generation volume	31
2.3 Cathodoluminescence (CL)	34
2.3.1 Physical process	34
2.3.2 Formation of the CL signal	36
2.3.3 Calculation of the CL signal using Hergert et al model	37
2.4 Electron beam induced current (EBIC)	42
2.4.1 Physical phenomena	42
2.4.2 Calculation of the EBIC in a normal collector p-n junction configuration	43
2.4.2.1 The charge collection probability of Donolato	44
2.4.2.2 Calculation of the EBIC	45
Chapter references	49
Chapter 3 Artificial Neural Networks	53
3.1 Introduction	53
3.2 The neuron	54

3.3 Neural networks topology	57
3.4 Feedforward networks	57
3.4.1 Single layer feedforward networks	57
3.4.2 Multilayer feedforward networks	58
3.5 The learning process	59
3.6 The learning algorithm	59
3.6 The backpropagation algorithm	60
3.6 The Levenberg Marquardt algorithm	61
3.7 ANN for function approximation	62
3.7.1 System identification	63
3.7.2 Inverse modeling	64
Chapter references	64
Chapter 4 Genetic algorithms	66
4.1 Introduction	66
4.2 The GA operators	67
4.2.1 Selection	67
4.2.2 Crossover	67
4.2.3 Mutation	67
4.3 The GA parameters	68
4.3.1 Population options	68
4.3.2 Fitness scaling options	68
4.3.3 Selection options	69
4.3.4 Crossover options	69
4.3.5 Mutation options	69
4.3.6 Stopping criteria options	70
4.4 The continuous GA algorithm	71
Chapter references	73
Chapter 5 Semiconductor parameter extraction using artificial neural networks and exhaustive search	74
5.1 Introduction	74
5.2 Parameter extraction based on ANN and exhaustive search	75
5.2.1 Preparation of the training and test data sets	75
5.2.2 Training the ANN algorithm	76
5.2.3 Testing the ANN algorithm	76
5.2.4 Oversampling of the signal using ANN	77
5.2.5 Parameter extraction through exhaustive search	77
5.3 Application to cathodoluminescence	79

5.3.1 Training the ANN	80
5.3.2 Testing the algorithm	85
5.3.3 Oversampling of the CL signal	88
5.3.4 Exhaustive search	90
5.3.5 Effect of measurement noise	92
5.4 Application to EBIC	94
5.4.1 Training the ANN	95
5.4.2 Testing the algorithm	99
5.4.3 Oversampling of EBIC	102
5.4.4 Exhaustive search	105
5.5 Conclusion	106
Chapter references	107
Chapter 6 Semiconductor parameter extraction using artificial neural networks and inverse modeling	109
6.1 Introduction	109
6.2 Parameter extraction based on ANN and inverse modeling	110
6.2.1 Preparation of the training and test data sets	110
6.2.2 Training the ANN algorithm	111
6.2.3 Testing the ANN algorithm	111
6.3 Application to cathodoluminescence	112
6.3.1 Training the algorithm	112
6.3.2 Testing the algorithm	117
6.4 Application to EBIC	120
6.4.1 Training the algorithm	120
6.4.2 Testing the algorithm	124
6.5 Conclusion	126
Chapter references	127
Chapter 7 Semiconductor parameter extraction using genetic algorithms	129
7.1 Introduction	129
7.2 Parameter extraction using genetic algorithms	130
7.2.1 Initialize the parameters	130
7.2.2 Define the objective function	130
7.2.3 Apply the genetic algorithm	130
7.2.4 Extract the solution	131
7.3 Application to cathodoluminescence	133
Effect of the initial population size	138
7.4 Application to EBIC	139

7.5 Conclusion	144
Chapter references	143
Chapter 8 Conclusion and future work	144
8.1 Conclusion	144
8.2 Future work	145
Author publications	147
Appendix A	148

List of Figures

- Figure 2.1: Schematic geometry of the initial steps of electron scatterings . 24
- Figure 2.2: A schematic illustration of elastic and inelastic scattering of an incident electron with a sample's atom 25
- Figure 2.3: Schematic diagram of signals generated due to electron beam interaction with bombarded specimen. 26
- Figure 2.4: Monte Carlo simulation of the energy dissipation in a Silicon sample using the slowing down approximation of Bethe for various values of the incident energy beam. 30
- Figure 2.5: Monte Carlo simulation showing the size of the interaction volume as a function of the energy beam (number of electrons is 250). 32
- Figure 2.6: Monte Carlo simulation showing the size of the interaction volume as a function of the atomic number (number of electrons is 250 with initial energy of 15KeV). 32
- Figure 2.7: Monte Carlo simulation showing the interaction volume as a function of the tilt angle (number of electrons is 250 with initial energy of 15KeV). 33
- Figure 2.8: The three main approximations of the generation volume shape: (a) pear, (b) spherical, and (c) hemispherical. 33
- Figure 2.9: Schematic illustration of intrinsic luminescence in direct (a) and indirect (b) semiconductors. 35
- Figure 2.10: Schematic illustration of extrinsic luminescence: (1), (2) band-donor or acceptor, (3), (4) donor-acceptor. 35
- Figure 2.11: Schematic illustration of the model employed for deriving CL signal. 36
- Figure 2.12: CL signal calculated using the model of Hergert et al with a Wu and Wittry generation rate for a GaAs semiconductor: (a) dependence on diffusion length L , (b) dependence on dead layer thickness Z_t , and (c) dependence on absorption coefficient α . 41
- Figure 2.13: Different configurations used for EBIC measurements, the shaded area represents the built-in potential: (a) and (c) are normal configurations, (b) and (d) are planar configurations. 43
- Figure 2.14: Normal collector configuration of a p-n junction, with x distance between the beam and the junction. 44
- Figure 2.15: EBIC profiles for the region around the edges of a depletion layer located at $0.3\mu\text{m}$: (a) different values of the diffusion length L ($S=0$), (b) different values of the surface recombination velocity ($L=3\mu\text{m}$). 48
- Figure 3.1: Biological neuron (a), and an artificial neuron (b). 55

Figure 3.2: Types of activation functions: (a) Threshold function, (b) Piecewise-linear function, (c) Sigmoid function.	56
Figure 3.3: Feedforward network with a single layer.	58
Figure 3.4: Typical feedforward network with two hidden layers and an output layer.	58
Figure 3.5: ANN system identification.	63
Figure 3.6: ANN inverse modeling.	64
Figure 5.1: Training curve of the ANN (CL).	82
Figure 5.2: Percentage error between theoretical training CL signal and the training CL signal calculated by ANN.	83
Figure 5.3: Probability density function PDF(bars) and cumulative density function CDF (solid line) of the percentage error of the training set (CL).	84
Figure 5.4: Percentage error between the theoretical test CL signal and the test CL signal calculated by ANN.	86
Figure 5.5: PDF and CDF of the percentage error for the test set (CL).	87
Figure 5.6: Percentage error between the theoretical oversampled CL signal and the oversampled CL signal calculated by ANN.	90
Figure 5.7: PDF and CDF of the percentage error for the oversampled set (CL).	90
Figure 5.8: PDF and CDF of percentage error of the parameters α , L, Zt, Q.	91
Figure 5.9: PDF and CDF of percentage error of the parameters α , L, Zt, Q with a SNR of 10 dB.	93
Figure 5.10: PDF and CDF of percentage error of the parameters α , L, Zt, Q with a SNR of 20 dB.	93
Figure 5.11: PDF and CDF of percentage error of the parameters α , L, Zt, Q with a SNR of 30 dB.	94
Figure 5.12: Training curve of the ANN (EBIC).	96
Figure 5.13: Percentage error between theoretical training EBIC and the training EBIC calculated by ANN.	97
Figure 5.14: Probability density function PDF (bars) and cumulative density function CDF (solid line) of the percentage error for the training set (EBIC).	98
Figure 5.15: Percentage error between the theoretical test EBIC and the test EBIC calculated by ANN.	100
Figure 5.16: PDF and CDF of the percentage error for the test set (EBIC).	101
Figure 5.17: Percentage error between the theoretical oversampled EBIC and the oversampled EBIC calculated by ANN	103
Figure 5.18: PDF and CDF of the percentage error for the oversampled set (EBIC).	104
Figure 5.19: PDF and CDF of percentage error of the parameters L, S.	105
Figure 6.1: Training curve of the ANN (CL)	114

Figure 6.2: Percentage error between theoretical parameters and the parameters calculated by ANN for the training set (CL)..	115
Figure 6.3: Probability density function PDF (bars) and cumulative density function CDF (solid line) of the percentage error of L, α , S, Q, and Zt for the training set	116
Figure 6.4: Percentage error between the theoretical parameters and the parameters calculated by ANN for the test set (CL).	118
Figure 6.5: PDF and CDF of the percentage error of L, α , S, Q, and Zt for the test set.	119
Figure 6.6: Training curve of the ANN (EBIC)	122
Figure 6.7: Percentage error between theoretical training parameters and the parameters calculated by ANN (EBIC).	123
Figure 6.8: PDF (bars) and CDF (solid line) of the percentage error for the training set (EBIC).	123
Figure 6.9: Percentage error between the theoretical parameters and the parameters calculated by ANN for the test set (EBIC)	125
Figure 6.10: PDF and CDF of the percentage error of L, S of the test set	126
Figure 7.1: Flowchart of the proposed parameter extraction algorithm based on GA	132
Figure 7.2: Cost surface of the objective function of the two parameters L and α .	134
Figure 7.3: Convergence of the GA: (a) best and mean fitness, (b) average distance between individuals	135
Figure 7.4: PDF (bars) and CDF (solid line) of the percentage error for each of the parameters L, α , S, Q, and Zt..	137
Figure 7.5: PDF and CDF of the percentage error for each of the parameters L, and S.	141

List of Tables

Table 5.1: Summary of different steps of the parameter extraction algorithm based on ANN and exhaustive search	79
Table 5.2: Semiconductor parameters used for training (CL)	81
Table 5.3: ANN parameters used for training (CL)	82
Table 5.4: Error for 95% and 100% of the training samples (CL)	84
Table 5.5: Semiconductor parameters used for testing the ANN (CL)	85
Table 5.6: Error for 95% and 100% of the test samples (CL)	87
Table 5.7: Semiconductor parameters used to obtain the oversampled CL signal	88
Table 5.8: Error for 95% and 100% of the oversampled samples (CL)	90
Table 5.9: Error of the four parameters α , L, Zt, Q for 95% and 100% of the cases	92
Table 5.10: Semiconductor parameters used for training (EBIC)	95
Table 5.11: ANN parameters used for training (EBIC)	96
Table 5.12: Error for 95% and 100% of the training samples (EBIC)	99
Table 5.13: Semiconductor parameters used for testing the ANN (EBIC)	99
Table 5.14: Error for 95% and 100% of the test samples (EBIC)	101
Table 5.15: Semiconductor parameters used to obtain the oversampled EBIC	102
Table 5.16: Error for 95% and 100% of the oversampled samples (EBIC)	104
Table 5.17: Error of the two parameters L, S for 95% and 100% of the cases (EBIC)	106
Table 6.1: Semiconductor parameters used for training (CL)	113
Table 6.2: ANN parameters used for training.(CL)	114
Table 6.3: Percentage error of the five parameters L, α , S, Q, and Zt for 95% and 100% of the training samples (CL)	117
Table 6.4: Semiconductor parameters used for testing the ANN (CL).	117
Table 6.5: Percentage error of the five parameters L, α , S, Q, Zt for 95% and 100% of the test samples .	119
Table 6.6: Semiconductor parameters used for training (EBIC).	121
Table 6.7: ANN parameters used for training (EBIC).	121
Table 6.8: percentage error of the two parameters for 95% and 100% of the samples (EBIC).	124
Table 6.9: Semiconductor parameters used for testing the ANN (EBIC).	124
Table 6.10: Percentage error of the two parameters L and S for 95% and 100% of the samples.	126
Table 7.1: Lower and upper bounds of semiconductor parameters (CL)	133

Table 7.2: GA parameters (CL)	134
Table 7.3: Maximum percentage error of the five parameters L, α , S, Q, and Zt for 95% and 100% of the runs, respectively	137
Table 7.4: Maximum percentage error of the five parameters L, α , S, Q, and Zt for 95% and 100% of the runs, respectively, with different initial population size	139
Table 7.5: Lower and upper bounds of semiconductor parameters (EBIC)	140
Table 7.6: GA parameters (EBIC)	140
Table 7.1: Maximum percentage error of the two parameters L and S for 95% and 100% of the runs, respectively	141

Chapter 1

Introduction

Contents

1.1 Background	
1.2 Aim of this work	
1.3 Thesis organization	
1.4 Thesis contributions	
chapter references	

1.1 Background

The investigation of electronic and optical properties of semiconductors is of fundamental importance as the performance of electronic devices is dictated by the properties of the carriers inside the materials. The scanning electron microscopy has proven to be an effective, useful and nondestructive technique in the characterization of semiconductor materials. In scanning electron microscopy, Electron Beam Induced current (EBIC) and Cathodoluminescence (CL) have been extensively used as an analytical tool in the characterization of semiconductor materials. They have been widely used for the investigation of crystal defects in semiconductors [1-7] and for the extraction of defect free region semiconductor parameters such as: absorption coefficient (α), diffusion length (L), dead layer thickness (Zt), relative quantum efficiency (Q) and the normalized surface recombination velocity (S) [8-14]. The

extraction of these parameters is difficult because of their nonlinear and complex effect on the CL/EBIC signal, and the high interaction between them.

The parameter extraction problem is a multi-minimum optimization problem [15,16] and it can be solved using several optimization techniques, which can be roughly classified into two categories: deterministic optimization algorithms (pseudo-objective function substitution method (POSM)[17], Levenberg-Marquardt [18], the gradient descent methods [19],...) and stochastic optimization algorithms (simulated annealing method (SA) [20], simulated 1-diffusion method (SD)[21], genetic algorithms (GA) [22, 23]...). These methods were applied to parameter extraction of semiconductor devices and integrated circuits [22-32].

Optimization algorithms tend to adjust the inputs such that the cost function, also known as the objective function, is either minimized or maximized (depending whether it is a minimization or a maximization problem). One of the largest problems in optimization is to determine whether the solution found is a global solution (corresponds to finding the global minimum/maximum) or a suboptimum solution (corresponds to finding the local minimum/maximum).

The parameter extraction can be described as an optimization problem as follows: suppose that a process f has an output \mathbf{y} which is a function of a set of parameters \mathbf{x} , that is: $\mathbf{y} = f(\mathbf{x})$. Given a process output measurement \mathbf{y}' , the parameter extraction problem is equivalent to minimizing the error function $E(\mathbf{x})$ with respect to the process parameters \mathbf{x} , subject to a set of constraints \mathbf{C} . The error function $E(\mathbf{x})$ is a general error function between \mathbf{y}' and $f(\mathbf{x})$, that is: $E(\mathbf{x}) = E(\mathbf{y}' - f(\mathbf{x}))$ and it represents the objective function to be minimized. The optimization process determines the set of parameters \mathbf{x}^* that minimizes the objective function $E(\mathbf{x})$ subject to the set of constraints \mathbf{C} (i.e. the upper and lower bounds on parameters), that is:

$$\mathbf{x}^* = \underset{\substack{\mathbf{x} \\ \text{subject to } C}}{\text{arg min}} (E(\mathbf{x})) \quad (1-1)$$

1.2 Aim of this work

The aim of this work is to extract the defect free region semiconductor parameters from the EBIC/CL signals using advanced signal processing techniques. Particularly, we are interested in the simultaneous extraction of related semiconductor parameters (α , L, S, Zt, Q) from any EBIC/CL; our novel approaches are based on powerful modeling and optimization techniques: Artificial Neural Networks and Genetic Algorithms. To the best of our knowledge, no work has been reported in this regard.

Artificial neural networks have many desirable properties such as: their ability to learn by example through training, their ability to generalize and predict, and finally their inherent parallel computation capability. These characteristics suggest that they can be used for modeling of complex systems and processes and consequently motivated us to employ them for the modeling of the EBIC/CL signal generation process.

At the other hand, by being able to formulate the semiconductor parameter extraction problem as an optimization problem, suggests that powerful optimization techniques can be used to find near-optimum solutions with reasonable cost. This motivated us to explore the feasibility of using evolutionary optimization algorithms that exhibit many desirable properties such as, their moderate computational complexity and their excellent performance. Because genetic algorithms are the most popular evolutionary algorithms, we use them here for semiconductor parameter extraction.

1.3 Thesis organization

The dissertation is divided into the following chapters:

- Chapter 1 gives a brief background about the semiconductor parameter extraction problem, describes the aim of the dissertation and its major contributions to the literature.
- Chapter 2 talks about the fundamentals of CL and EBIC in the Scanning Electron Microscopy. The chapter also describes the Monte Carlo simulation of the electron beam interactions with the semiconductor material. Additionally, MATLAB related codes are reported in the Appendix.
- Chapter 3 contains fundamentals of Artificial Neural Networks.
- Chapter 4 contains fundamentals of Genetic Algorithms.
- In chapter 5 a new approach based on feedforward ANN and exhaustive search for the simultaneous extraction of related semiconductor parameters is developed.
- In chapter 6 a new approach for the simultaneous extraction of related semiconductor parameters based on feedforward ANN and inverse modeling is developed and described.
- In chapter 7 a new approach for the simultaneous extraction of related semiconductor parameters based on genetic algorithms is developed.
- Chapter 8 consists of a conclusion that summarizes the results and the contributions of the dissertation, and discusses some recommendations for future work.

1.4 Thesis contributions

The major contribution of this work resides in the introduction of new signal processing techniques to the field of scanning electron microscopy which opens the door widely to the use of these techniques in other related fields. The main contributions of the dissertation are summarized into the following points:

- Development of a new approach that is based on ANN and exhaustive search for the simultaneous extraction of related semiconductor parameters.
- Development of a new approach that is based on ANN and inverse modeling for the simultaneous extraction of related semiconductor parameters.
- Development of a new approach that makes use of Genetic Algorithms for the simultaneous extraction of related semiconductor parameters.

Chapter references

- 1- H, J, Leamy, "Charge collection scanning electron microscopy". J.Appl.Phys, 53(6). pp. R51-R80. 1982
- 2- C, Donolato. "On the theory of SEM charge collection imaging of localized defects in semiconductors". Optik 52No. 1. 19-36. 1978
- 3- A, Jackubowicz. "On the theory of electron beam induced current contrast from pointlike defects in semiconductors". J. Appl.Phys, 57(4), pp. 1194-1199, 15. 1985
- 4- A, Jackubowicz. "Theory of cathodoluminescence contrast from localized defects in semiconductors". J. Appl.Phys. 59(6). pp. 2205-2209. 15. 1986

- 5- A, Jakubowicz. "Theory of lifetime measurement with the scanning electron microscope in a semiconductor containing a localized defect: transient analysis". J. Appl.Phys. 58(4). pp 1483-1488. 1985
- 6- K, L, Pey. D,S,H, Chan. and J,C,H, Phang. "Cathodoluminescence contrast of localized defects part1. Numerical model for simulation, part2. defect investigation", Scanning Mmicroscopy. Vol. 9, pp. 355-380, No. 2. 1995
- 7- S, Hildebrandt. J, Schreiber. W, Hergert. H, Uniewsk. and H, S, Leipner. "Theoretical fundamentals and experimental materials and defect studies using quantitative scanning electron microscopy-cathodoluminescence/electron beam induced current on compound semiconductors". Scanning microscopy. Vol. 12, No. 4 (pages 535-552). 1998
- 8- H, C, Casey. Jr, D, D, Sell. and K, W, Wecht "concentration dependence of the absorption coefficient for n-and p-type GaAs between 1.3 and 1.6 eV". Journal of applied physics, Vol. 46, No1. 1975
- 9- V, K, S, Ong. "A direct method of extracting surface recombination velocity from an electron beam induced current line scan. Review of scientific instruments", Vol. 69, No. 4. 1998
- 10- W, Hergert. P, Reck. L, Pasemann. And J, Schreiber "Cathodoluminescence measurements using the scanning electron microscope for the determination of semiconductor parameters". Phys. Stat. Sol. (a)101, 611. 1987
- 11- F, Koch. W, Hergert. G, Oelgart. and N, Puhlmann "determination of semiconductor parameters by electron beam induced current and cathodoluminescence measurements". phys. Stat. Sol. (a)109,261. 1989
- 12- S, Hildebrandt. J, Schreiber. W, Hergert. and V, I, Petrov, "Determination of the absorption coefficient and the luminescence spectrum of GaAs and GaAs_{1-x}

- xP_x ($x=0.375, 0.78$) from beam voltage dependent measurements of cathodoluminescence spectra in the scanning electron microscope". Phys. Stat. Sol. (a)110, 283. 1989
- 13- N, Puhlmann. G, Oelgart, "Semiconductor Characterization by means of EBIC, Cathodoluminescence, and photoluminescence". Phys. Stat. Sol. (a) 122,705. 1990
- 14- D, S, H, Chan. K, L, Pey. and J, C, H, Phang (1993) Semiconductor parameter extraction using cathodoluminescence in the scanning electron microscope. IEEE transactions on electron devices. Vol. 40. No. 8.
- 15- A, K, Hartmann. H, Rieger, "Optimization algorithms in Physics". Wiley-VCH Verlag Berlin GmbH. Berlin (Federal Republic of Germany) 1st edition. 2002.
- 16- R, L, Haupt. and S, E, Haupt, "Practical Genetic Algorithms". 2nd edition. Jhon Wiley and Sons. INC. publications. New Jersey. 2004.
- 17- G, L, Tan. S, W, Pan. W, H, Ku. and A, J, Shey, "ADIC-2.C: A General-Purpose Optimization program Suitable for Integrated Circuit Design Applications Using the Pseudo Objective Function Substitution Method (POSM)". IEEE Transactions on computer aided design. Vol. 1. No. 11. pp. 1150-1163. 1988.
- 18- M, I, Lourakis. A, A, Argyros, "Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment". Proceeding of the Tenth IEEE international conference on computer vision. Vol. 2, pp. 1526-1531. 2005.
- 19- S, Haykin, "Neural Networks; A Comprehensive Foundation". Prentice Hall, 2 edition. 1999

- 20- D, Bertsimas. and J, Tsitsiklis, "Simulated Annealing. Statistical Science".
Vol. 8. No. 1. 10-15. 1993
- 21- T, Sakurai. B, Lin. and A, R, Newton, "Fast Simulated Diffusion: An
Optimization Algorithm for Multimimum Problems and Its Application to
MOSFET Model Parameter Extraction". IEEE Transactions on computer
aided design. Vol 11. No 2. pp. 228-234. 1992
- 22- G, Antoun. El-Nozahi, M. Fikry, W. Abbas, H. "A hybrid genetic
algorithm for MOSFET parameter extraction". IEEE Canadian Conference on
Electrical and Computer Engineering. 4-7 . vol.2. pp: 1111- 1114. 2003
- 23- Y, Li. "An automatic parameter extraction technique for advanced CMOS
device modeling using genetic algorithm". Microelectronic Engineering. 84.
260–272. 2007
- 24- Y, S, Yu, "Parameter Extraction of Si-Based Single-Electron Transistor with
Electrical Tunnel Barriers for Circuit Simulation". Journal of the Korean
Physical Society. Vol. 47. pp. S547-S551. 2005
- 25- C, Scharff. J, C, Carter. and A, G, R, Evans, "New and fast MOSFET
parameter extraction method". Electronics letters. Vol. 28 No. 21 pp. a2006-
2008. 1992
- 26- P, R, Karlsson. K, O, Jeppson, "An efficient parameter extraction algorithm
for MOS transistor models". IEEE Transaction on Electron Devices. Vol. 39.
No. 9 2070-2076. 1992
- 27- Y, Hao. L, A, Yang. C, L, Yu, "An optimization technique for parameter
extraction of ultra_deep submicron LDD MOSFET's". Solid State Electronics
50.1540-1545. 2006

- 28-D, Busuioc, "Circuit model parameter extraction and optimization for microwave filters. Master thesis. University of Waterloo". Ontario Canada. 2002.
- 29-H, Wang. H, Z, Yang. and G, Z, Hu, "Combinatorial Algorithms for BJT Model Parameters Extraction". ICSE'96 Proc. pp. 144-147. Penang, Malaysia. 1996.
- 30-K, Garwacki, "Extraction of BJT model parameters using optimization method". IEEE transactions on computer-aided design. Vol. 7. No. 8. pp. 850-854. 1998.
- 31-B, Liu. J, Lu. Y, Wang. and Y, Tang, "An effective parameter extraction method based on memetic differential evolution algorithm". Microelectronics Journal. Volume 39. Issue 12.. Pages 1761-1769. 2008.
- 32-R, A, Thakker. M, B, Patil. K, G, Anil, "Parameter extraction for PSP MOSFET model using hierarchical particle swarm optimization". Engineering Applications of Artificial Intelligence. Volume 22. Issue 2. Pages 317-328. 2009.

Chapter 2

Scanning Electron Microscopy Fundamentals

Contents

2.1 Introduction	
2.2 Interaction of the incident electrons with irradiated sample	
2.3 Cathodoluminescence CL	
2.4 Electron Beam Induced Current	
Chapter references	

2.1 Introduction

The Electron Microscopy [1,2,3,4] was developed due to the limitations of light microscopy [5], it has been an important tool for the study and analysis of different specimens, it permits the observation and characterization on a nanometer to micrometer scale.

The Scanning electron microscopy SEM [1,2,3,4] is a type of Electron Microscopy, it is one of the most widely used techniques in research and industry. Basically a beam of highly accelerated electrons that typically has an energy ranging from 0.5 KeV to 50 KeV bombard a sample to be examined; When they penetrate into the sample they undergo interaction processes (figure2-1) that may result in the loss of energy, change of direction, and the creation of secondary signals (electron signals or photon signals).

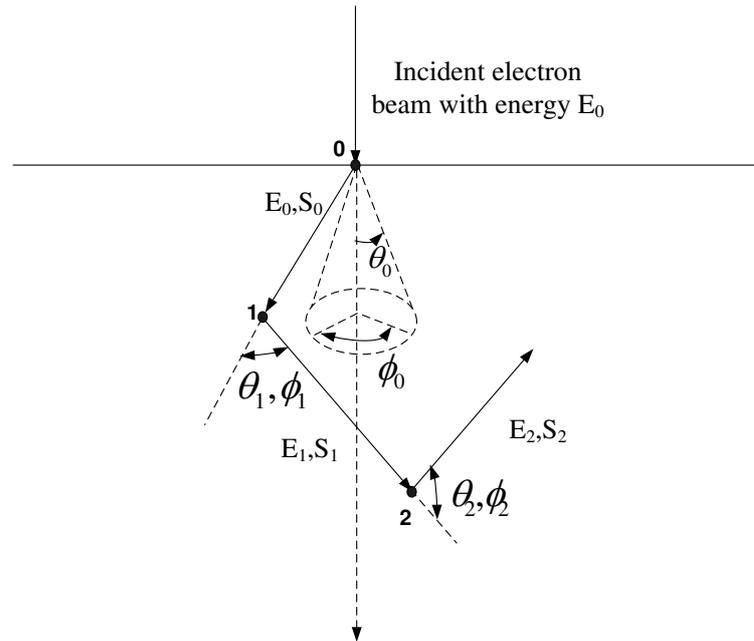


Figure 2-1. Schematic geometry of the initial steps of electron scatterings.

2.2 Interaction of the incident electrons with the irradiated matter

2.2.1 Physical process

Generally, the interaction processes that undergo the incident electrons can be grouped in two types [6]:

Elastic: in which only the trajectory of the incident electron changes while the kinetic energy and velocity remain constants. Elastic scattering results from the deflection of the incident electron by the positive charge of the nucleus of the atoms of the sample. It gives rise to high energy backscattered electrons. The cross section for elastic scattering obeys an inverse square dependence with the electron beam energy, and a squared proportional with the atomic number of the sample.

Inelastic: in which the incident electron lose significant and definite amount of energy, inelastic process occurs during the interactions of the incident electron with the electrons and atoms of the sample.

A schematic illustration of the elastic and inelastic interactions is illustrated in figure2-2.

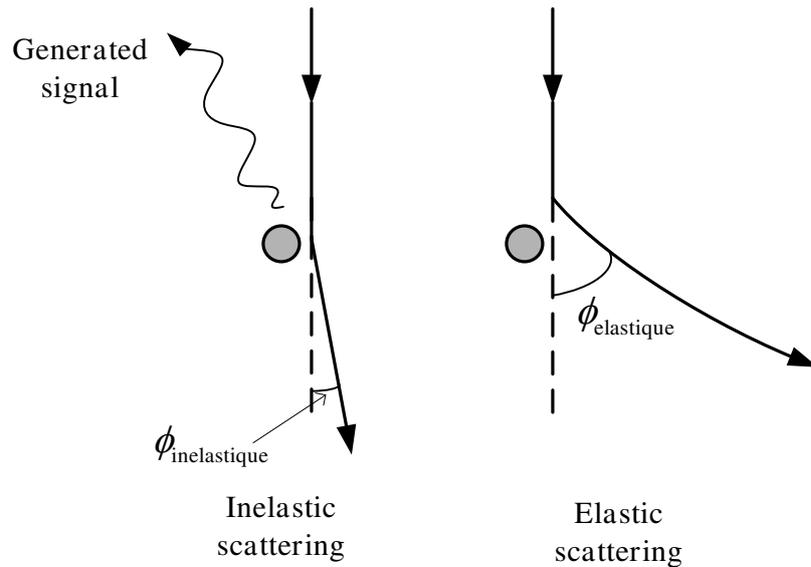


Figure2-2. A schematic illustration of elastic and inelastic scattering of an incident electron with a sample's atom.

As a consequence of the inelastic interactions a wide range of secondary signals are produced, Figure2-3 illustrates these signals (The directions shown doesn't represent the physical direction of the signal).

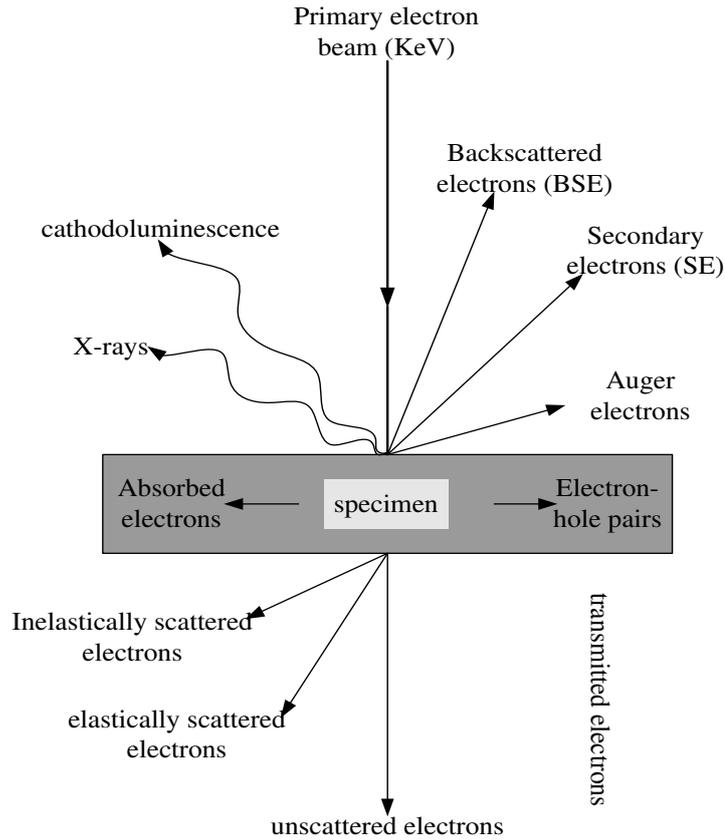


Figure 2-3. Schematic diagram of signals generated due to electron beam interaction with bombarded specimen.

2.2.2 Function of energy dissipation

Constantly the inelastic processes reduce the energy of a primary electron, finally to the point at which it is "captured". It is useful to study the energy loss of the incident electrons quantitatively as a function of the distance traveled and also the target composition (atomic number Z , atomic weight A , and density ρ). Bethe in 1933 derived the continuous energy loss approximation for all inelastic processes, the energy loss per unit of distance traveled by the incident electron dE/ds (the electron beam suffers a mean energy loss dE in penetrating a distance ds) is given by [7, 8]:

$$\frac{dE}{ds} = -\frac{2\pi e^4}{E} NZ \ln\left(\frac{aE \text{ (eV)}}{J}\right) \quad (2-1)$$

where E is the kinetic energy of the electrons, e is the electron charge, N is the number of atom's/cm³, and Z is the atomic number of the sample. Bethe gave for the constant $a=2$, J is a function of the atomic number and is given by [7, 8]:

$$J \text{ (eV)} = 11.5Z \quad (2-2)$$

(2-2) can be expressed as follows:

$$\frac{dE}{ds} = -\frac{2\pi e^4}{E} NZ \ln\left(\frac{2E \text{ (eV)}}{11.5Z}\right) \quad (2-3)$$

It is suitable to express the energy loss in the unit of KeV/ μm , and in terms of the atomic weight and density of the target material, (2-3) becomes [7, 8]:

$$\frac{dE}{ds} = -7.83 \left(\frac{\rho Z}{AE}\right) \ln\left(\frac{174E}{Z}\right) \quad \left(\frac{\text{KeV}}{\mu\text{m}}\right) \quad (2-4)$$

where, ρ (g/cm³) is the sample density, and A (g) is the atomic weight of the sample.

2.2.3 Electron range

As a result of the elastic and inelastic scatterings that the incident electrons undergo within the material, the original trajectories are randomized. The distance traveled between the point of entry and the final resting place of an incident electron undergoing inelastic scatterings is the Bethe range R_B , it is defined from the Bethe equation as follows [8,9]:

$$R_B \text{ (cm)} = \int_{E_0}^0 \frac{1}{dE/ds} dE \quad (2-5)$$

(2-5) overestimates the electron range because it neglects the effect of elastic scatterings. The effective depth to which energy dissipation extends is much smaller and it is known as Gruen range or penetration range R_e , it is a function of the primary energy beam E_0 and it is given by this general formula [10]:

$$R = (K / \rho)E_0^\alpha \quad (2-6)$$

where ρ is the material density in g/cm^3 , K and α depends on the atomic number and E_0 . Everhart and Hoff, for example, proposed for K , α and R the values [10]:

$$R(\mu\text{m}) = (0.0398 / \rho)E_0^{1.75} \quad (2-7)$$

where E_0 is in KeV.

Whereas Kanaya and Okayama proposed the following expression [10]:

$$R(\mu\text{m}) = (0.0276A / \rho Z^{0.889})E_0^{1.67} \quad (2-8)$$

where A is the atomic weight in (g/mol), Z is the atomic number, and E_0 is in KeV.

2.2.4 Generation of pairs

The generation of electrons-holes pairs is an inelastic process escorting the penetration of electrons inside the semiconductor sample; we can define a function $G(\vec{r}, R)$ describing the distribution of generated pairs in the volume. It corresponds to the number of generated pairs per unit volume and unit time ($\text{cm}^{-3} \cdot \text{s}^{-1}$) in the point spotted by the vector $\vec{r}(x, y, z)$. The parameter R here indicates the dependence on the energy beam.

Virtually, we don't need to know the explicit form of the function G but only its projection on a plan. If the projection of G on the plan (xy) , for example, is $g(z)$ it

corresponds to the number of pairs generated per unit depth per unit time. It is given by the following formula [8,11] :

$$g(z) = \frac{I_0}{qE_{e-h}} \frac{dE}{dz} \quad (2-9)$$

where I_0 is the incident electron beam current, q is electronic charge, E_{e-h} is the mean energy required to create an electron-hole pair, and dE is the mean energy loss per depth dz .

If we normalize the mean energy loss to the energy beam, and the depth to the electron range we can define the depth-dose function as follows:

$$\phi(z/R) = \frac{d(E/E_0)}{d(z/R)} \quad (2-10)$$

The depth-dose function represents the number of electron-hole pairs generated per electron of energy E per unit depth per unit time

Equation (2-9) becomes:

$$g(z) = \frac{I_0 E_0}{q E_{e-h} R} \phi(z/R) \quad (2-11)$$

In figure2-4 we reported the variation of the energy dissipation with the depth for different values of the incident energy beam using a Monte Carlo simulation (the code is detailed in Appendix A) in a silicon sample.

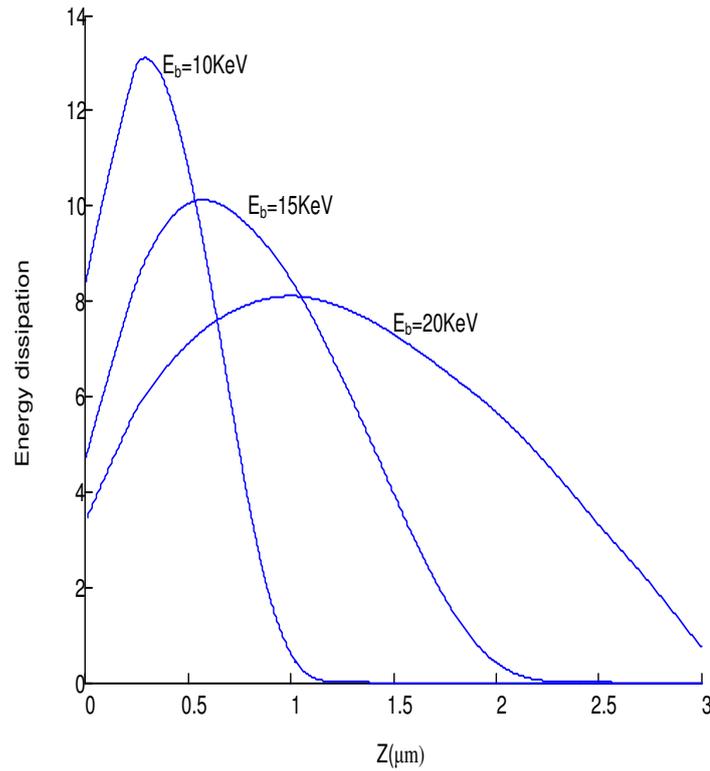


Figure2-4. Monte Carlo simulation of the energy dissipation in a Silicon sample using the slowing down approximation of Bethe for various values of the incident energy beam.

The choice of the generation function is important to get a good accuracy for semiconductor parameter measurements. Various analytical generation functions have been published; the most famous are:

- Everhart and Hoff have proposed for the depth-dose function a polynomial expression [10,12]:

$$\phi(z/R) = 0.60 + 6.21(z/R) - 12.40(z/R)^2 + 5.69(z/R)^3 \quad (2-12)$$

It is the most used because of its simplicity.

- Wittry proposed a Gaussian function [13]

$$\phi(z/R) = A \exp \left[- \left(\frac{z/R - u_0}{\Delta u} \right)^2 \right] \quad (2-13)$$

- Kyser proposed a modified Gaussian function [14]:

$$\phi(z/R) = A \exp \left[- \left(\frac{(z/R) - u_0}{\Delta u} \right)^2 \right] - B \exp \left(- \frac{b \times (z/R)}{u_0} \right) \quad (2-14)$$

where, A, u_0 , Δu , B, b are constants.

The number of pairs created in the sample per unit time is the total generation rate $G_0(s^{-1})$ [8]:

$$G_0 = \int_0^{\infty} g(z) dz \quad (2-15)$$

2.2.5 Generation volume

The generation volume or interaction volume is the 3-dimensional space in which primary incident electrons have enough energy to interact with the specimen. Its shape and its size are controlled by both elastic and inelastic interactions: if elastic scatterings are the dominant interaction, electrons tend to scatter away from their original direction which gives "width" for the interaction volume. However if inelastic scatterings are the dominant interactions, electrons are less deviated; they penetrate into the sample along their original trajectories and lose energy as they penetrate.

The two factors that can determine which interactions, elastic or inelastic, can dominate are the primary energy beam and the atomic number of the sample.

As the primary energy beam increases, the incident electrons penetrate into the sample along a path close to their incident direction, as they lose energy by inelastic scatterings the probability of elastic scatterings increase so they begin deflected into

the sample. Figure2-5 shows a Monte Carlo simulation (see Appendix A for the MATLAB code) of the generation volume as a function of the energy beam.

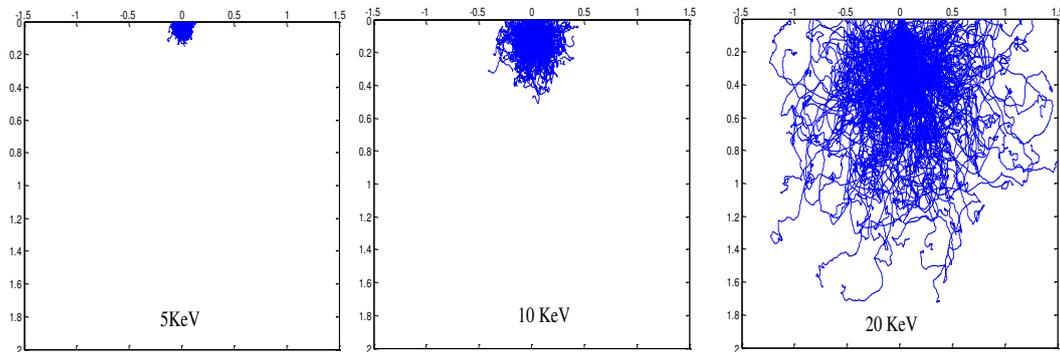


figure2-5. Monte Carlo simulation showing the size of the interaction volume as a function of the energy beam (number of electrons is 250).

Electrons entering a high atomic number sample are scattered away from their original directions. However in a low atomic number sample electrons penetrate into the sample and can lose energy by inelastic scatterings till their energy is such that the elastic scatterings can dominate. Figure2-6 shows a Monte Carlo simulation (see Appendix A for MATLAB codes) of the generation volume as a function of the atomic number of the target.

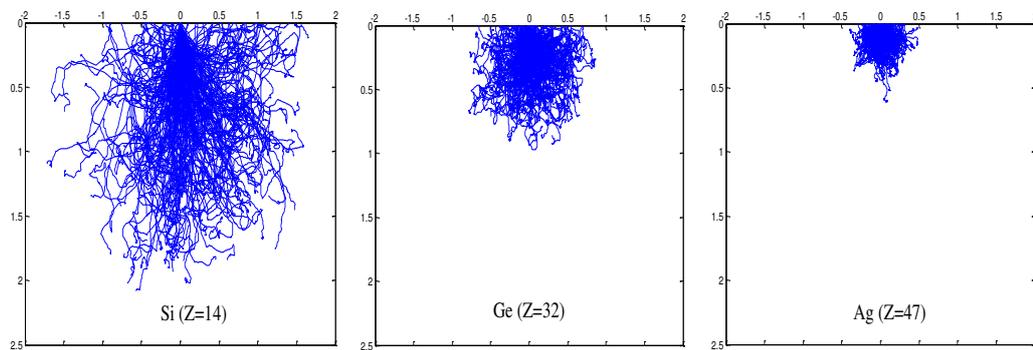


figure2-6: Monte Carlo simulation showing the size of the interaction volume as a function of the atomic number (number of electrons is 250 with initial energy of 15KeV).

The tilt angle, which is the angle between the sample surface and the horizontal plane, determines the symmetry of the interaction volume; as the sample is tilted away from the horizontal the interaction volume appears asymmetric. Figure2-7 shows a Monte Carlo simulation (see Appendix A for MATLAB codes) of the generation volume as a function of tilt angle.

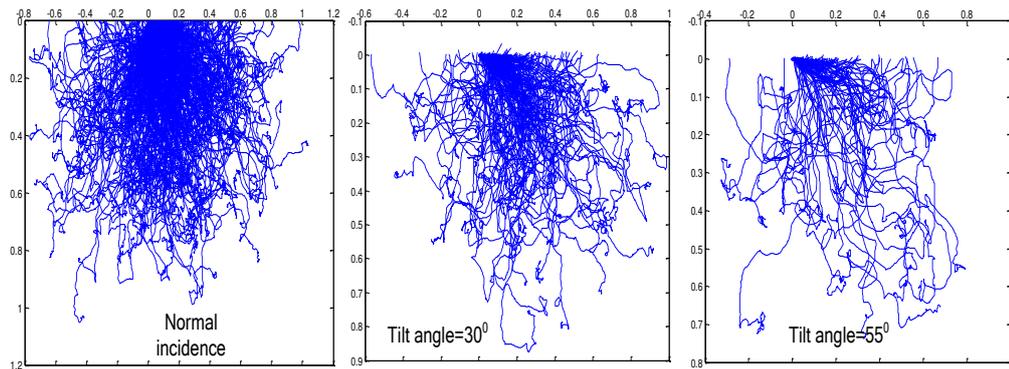


figure2-7. Monte Carlo simulation showing the interaction volume as a function of the tilt angle (number of electrons is 250 with initial energy of 15KeV).

From figure2-5,2-6 and 2-7 we can see that the boundaries of the generation volume can not be well defined but only approximated. Generally the shape of the generation volume is approximated to one of three main shapes: pear, spherical, or hemispherical shape (figure2-8).

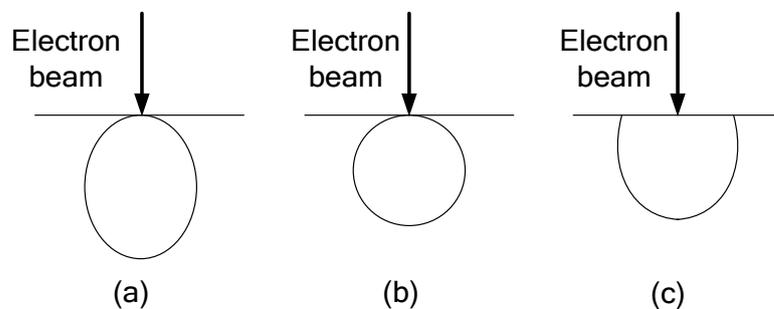


Figure2-8. The three main approximations of the generation volume shape: (a) pear, (b) spherical, and (c) hemispherical.

2.3 Cathodoluminescence (CL)

The cathodoluminescence (CL) analysis permits the evaluation of different properties of materials. For example; the CL spectroscopy can be used in the identification and measurement of luminescent center concentrations and distribution, the dependence of CL intensity on electron beam voltage is used to determine electronic properties (carrier diffusion length, surface recombination velocity...), CL maps can be used to find the concentration and distribution of defects (dislocations...).

2.3.1 Physical phenomena

The cathodoluminescence is the emission of photons from a solid supplied with electron excitation or cathode rays. As mentioned earlier, the inelastic processes occurring during the penetration of the incident energetic electrons in the bombarded sample generate electron-hole pairs, the excess carriers diffuse inside the material and recombine, either by radiative or nonradiative processes.

The radiative recombination may be an intrinsic or extrinsic mechanism [10,15,16] :

Intrinsic luminescence results from the recombination of electrons and holes across the fundamental energy gap; it is an intrinsic property of the material. In indirect gap semiconductors the recombination must be accompanied by the simultaneous emission of a photon and a phonon to conserve the momentum. A schematic illustration of the intrinsic luminescence in direct and indirect semiconductors is presented in Figure2-9.

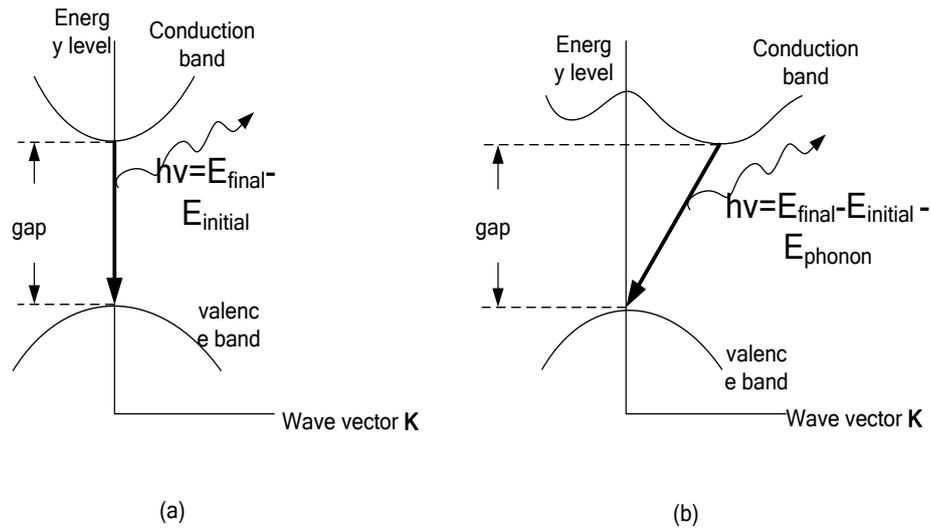


Figure2-9 –Schematic illustration of intrinsic luminescence in direct (a) and indirect (b) semiconductors.

Extrinsic luminescence results from the radiative transitions involving shallow or deep energy states located within the forbidden band gap in both direct and indirect semiconductors. A schematic illustration of the extrinsic emission is reported in figure2-10.

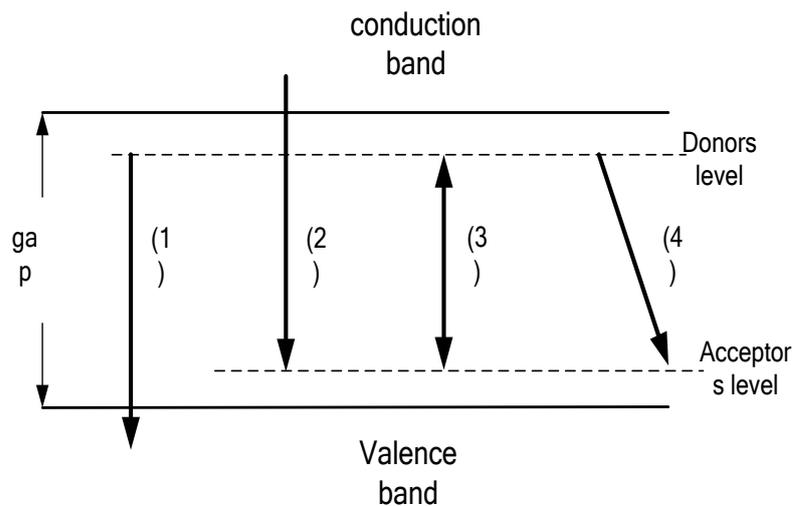


Figure2-10 –Schematic illustration of extrinsic luminescence: (1), (2) band-donor or acceptor, (3), (4) donor-acceptor.

2.3.2 Formation of the CL signal

Let us consider a semi infinite homogenous n-type semiconductor, bounded by the top surface (figure2-11), having minority carrier (holes) diffusion coefficient and life time D_p and τ , respectively, and surface recombination velocity v_s . bombarded on the top surface by an electron beam of energy E_0 .

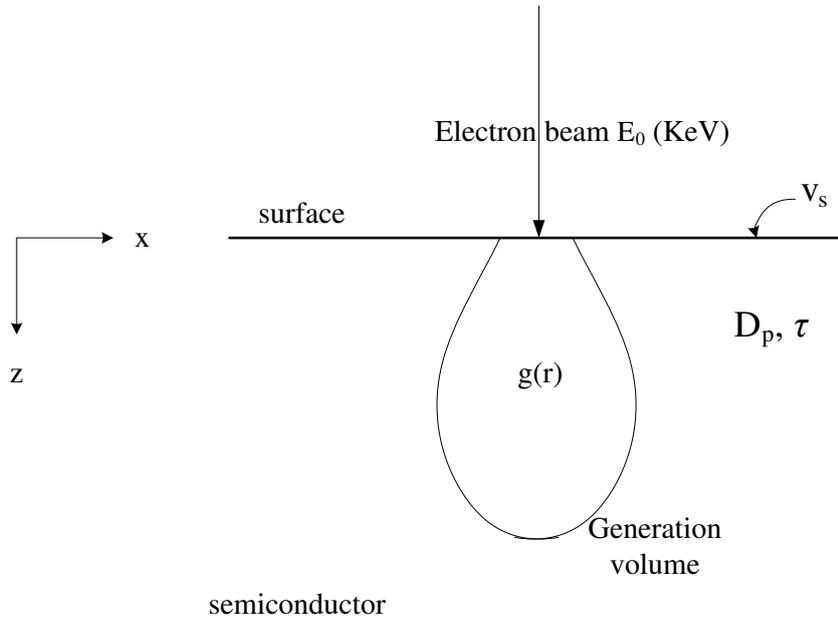


Figure2-11. Schematic illustration of the model employed for deriving CL signal.

The generation of photons is essentially governed by the stationary excess minority carriers concentration $p(r)$; which in the steady state obeys the differential equation given by [16,17,18]:

$$D_p \nabla^2 p(r) - \frac{p(r)}{\tau} + g(r) = 0 \quad (2-16)$$

where, $g(r)$ is the generation rate per unit volume and unit time.

With the boundary condition at the top surface, that takes into account the surface recombination:

$$D_p \left(\frac{dp(r)}{dz} \right) = v_s p(r) \quad (2-17)$$

The generation of photons is essentially governed by the stationary excess minority carriers' concentration $p(r)$.

$$I_{CL} \sim \int_V \frac{p(r)}{\tau_r} dV \quad (2-18)$$

After radiative recombination occurs the emitted photons propagate inside the semiconductor, due to optical absorption and reflection losses not all generated photons are emitted; only a fraction of them is coming out from the material. Thus the total CL intensity; which is the number of photons emitted per unit time; is given by [10, 19, 20]:

$$I_{CL} = \int_V F(z) \frac{p(r)}{\tau_r} dV \quad (2-19)$$

where, τ_r is the radiative recombination lifetime of the minority carriers, and F is the correction function for optical absorption and reflection losses.

2.3.3 Calculation of the CL signal using Hergert et al model

Many models were proposed for the calculation of the cathodoluminescence signal, using analytical approach [10, 20, 21, 22] or using Monte Carlo simulation [25]. We choose the model of Hergert et al [22-23-24] because of its simplicity. The details of that model are given now.

In this model the equation (2-19); which is the current of photons modified by the internal absorption of the luminescence radiation; is written as follows:

$$I_{CL} = 2\pi \int_0^{\theta_c} \sin \theta d\theta \int_V e^{-\alpha z / \cos \theta} \frac{p(r)}{\tau_r} dV \quad (2-20)$$

where θ_c is the critical angle of the total reflection, and α is the absorption coefficient of the semiconductor.

The minority carrier density in (2-20) is obtained from the continuity equation in cylindrical coordinates:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial p}{\partial r} \right) + \frac{\partial^2 p}{\partial z^2} - \frac{1}{L^2} p = -\frac{1}{D} g(r, z) \quad (2-21)$$

The boundary conditions for the geometry of semi infinite semiconductor of figure2-11 are as follows:

$$D_p \left. \frac{\partial p}{\partial z} \right|_{z=0} = v_s p(z=0) \quad (2-22-a)$$

$$p(z \rightarrow \infty) = 0 \quad (2-22-b)$$

The electron-hole pairs due to absorption of internal luminescence are not taken into account.

To solve (2-21) a Hankel transform of order 0 is used; the Hankel transform and the inverse Hankel transform of minority carrier density are given by:

$$\tilde{p}(\lambda, z) = \int_0^{\infty} p(r, z) J_0(\lambda r) r dr \quad (2-23-a)$$

$$p(r, z) = \int_0^{\infty} \tilde{p}(\lambda, z) J_0(\lambda r) \lambda d\lambda \quad (2-23-b)$$

where, J_0 is the Bessel function of first kind of order 0.

This gives the following equation:

$$\frac{d^2}{dz^2} \tilde{p}(\lambda, z) - \left(\lambda^2 + \frac{1}{L^2} \right) \tilde{p}(\lambda, z) = -\frac{1}{D} \tilde{g}(\lambda, z) \quad (2-24)$$

(2-24) is solved using Green's function, which leads to:

$$\tilde{p}(\lambda, z) = \frac{1}{4\pi D} \int_0^\infty G(z, z', \lambda) \tilde{g}(\lambda, z) \quad (2-25)$$

where $G(z, z', \lambda)$ is the Greens function of (2-24).

with:

$$\int_0^\infty r dr J_0(\lambda r) = \delta(\lambda) \frac{1}{\lambda} \quad (2-26)$$

where δ is the Dirac delta function.

Equation (2-20) becomes as follows:

$$I_{CL} = \frac{\pi}{\tau_r D} \int_0^{\theta_c} \sin \theta d\theta \int_0^\infty dz' \int_0^\infty G(z, z', 0) \tilde{g}(0, z') e^{-\alpha z / \cos \theta} dz \quad (2-27)$$

The CL signal was then expressed by a universal function which depends only on the depth distribution $g(z)$ as follows:

$$\Phi(x, z_1) = \int_0^\infty e^{-xz} g(z + z_1) dz \quad (2-28)$$

and the CL signal is then given by:

$$I_{CL} = \frac{2\pi}{\tau_r} \int_0^\infty \sin \theta d\theta F(\hat{\alpha}) \quad (2-28)$$

with:

$$\hat{\alpha} = \alpha / \cos \theta \quad (2-29-a)$$

$$F(x) = \frac{\tau G_0 e^{-xZt}}{(1-x^2L^2)} \left[\Phi(x, Zt) - \frac{Lx+S}{1+S} \Phi\left(\frac{1}{L}, Zt\right) \right] \quad (2-29-b)$$

where:

- G_0 is the total generation rate of the minority carriers.
- $\tau = \tau_{nr} \tau_r / (\tau_{nr} + \tau_r)$ is the total life time with τ_{nr} and τ_r are the non radiative and radiative life time of the semiconductor, respectively.
- L is the diffusion length of the excess minority carriers.
- g is the depth distribution function of the generation rate.
- Z is the distance from the surface.
- S is the normalized surface recombination velocity given by: $S = V_s/V_d$. where V_s is the surface recombination velocity and V_d is the diffusion velocity given by: $V_d = L/\tau$.

The depth distribution of the energy dissipation is of Wu and Wittry, it is the modified Gaussian approximation of Kyser[14] reported in (2-14). It can be written as:

$$\phi(u) = A \exp\left[-\left(\frac{u-u_0}{\Delta u}\right)^2\right] - B \exp\left(-\frac{bu}{u_0}\right) \quad (2-30)$$

where: $u = \rho Z/R$, ρ (g/cm³) is the density of the semiconductor material and Z is the depth.

The electron range R is:

$$R(\text{cm}) = \frac{2.56 \times 10^{-3}}{\rho} (E_0(\text{KeV})/30)^{1.7} \quad (2-31)$$

For GaAs: $u_0=0.125$, $\Delta u=0.350$, $b=4$, and $B/A=0.4$, the atomic number is 32, and the critical angle is 16° . Figure 2-12 illustrates CL signals calculated using the approach described before.

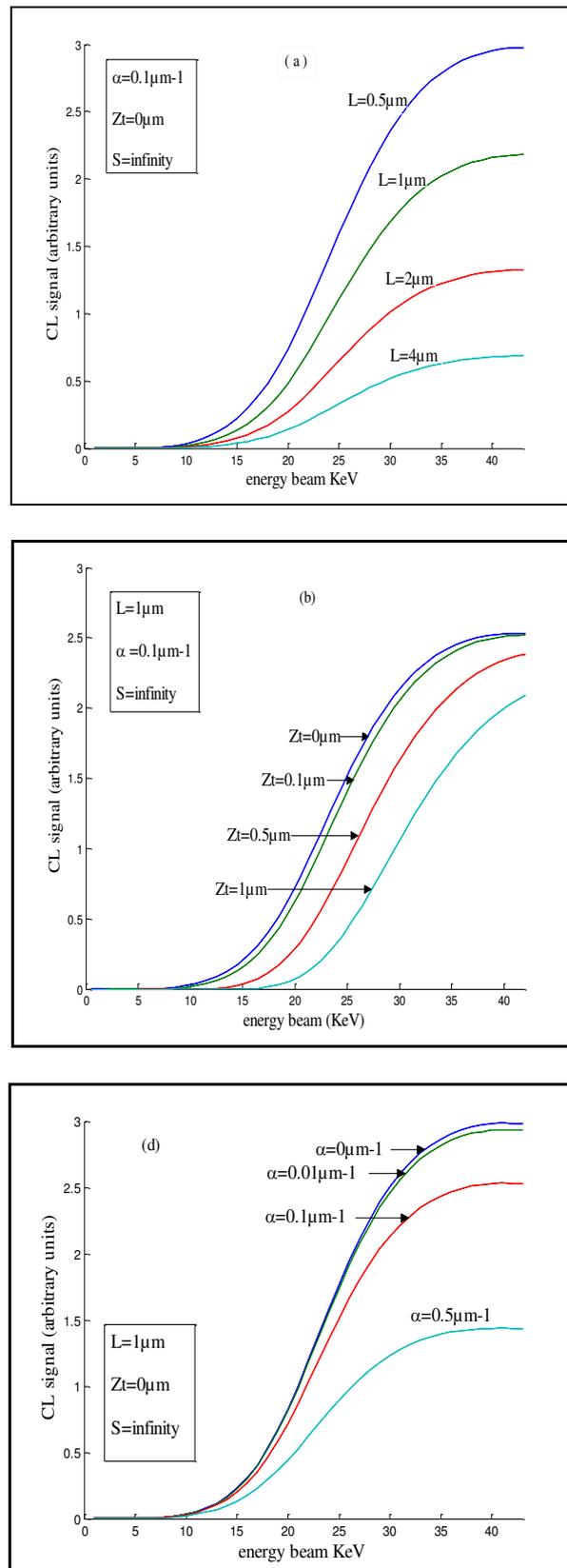


Figure 2-12. CL signal calculated using the model of Hergert et al with a Wu and Wittry generation rate for a GaAs semiconductor: (a) dependence on diffusion length L , (b) dependence on dead layer thickness Z_t , and (c) dependence on absorption coefficient α .

2.4 Electron beam induced current EBIC

The electron beam induced current (EBIC) technique allows examining semiconductor materials and looking for features such as crystallographic defects, as well as giving information about the semiconductor parameters that characterize the material such as the diffusion length and the surface recombination velocity.

2.4.1 Physical phenomena

As mentioned earlier, the inelastic processes occurring during the penetration of the incident energetic electrons in the bombarded sample generate electron-hole pairs, the excess carriers diffuse inside the material, when the semiconductor sample contains internal electric field (p-n junction or Schottky junction) the charge carriers are separated by that field and minority carriers can therefore reach the junction by diffusion this results in a charge-collection current or electron beam induced current EBIC, which can be amplified and measured externally when an external circuit is connected to the ends of the junction.

The EBIC can be performed in one of two modes:

- **Planar configuration** in which the junction (p-n junction or Schottky junction) is perpendicular to the electron beam.
- **Cross sectional or normal configuration** in which the junction (p-n junction or Schottky junction) is parallel to the electron beam.

These configurations are illustrated in figure2-13.

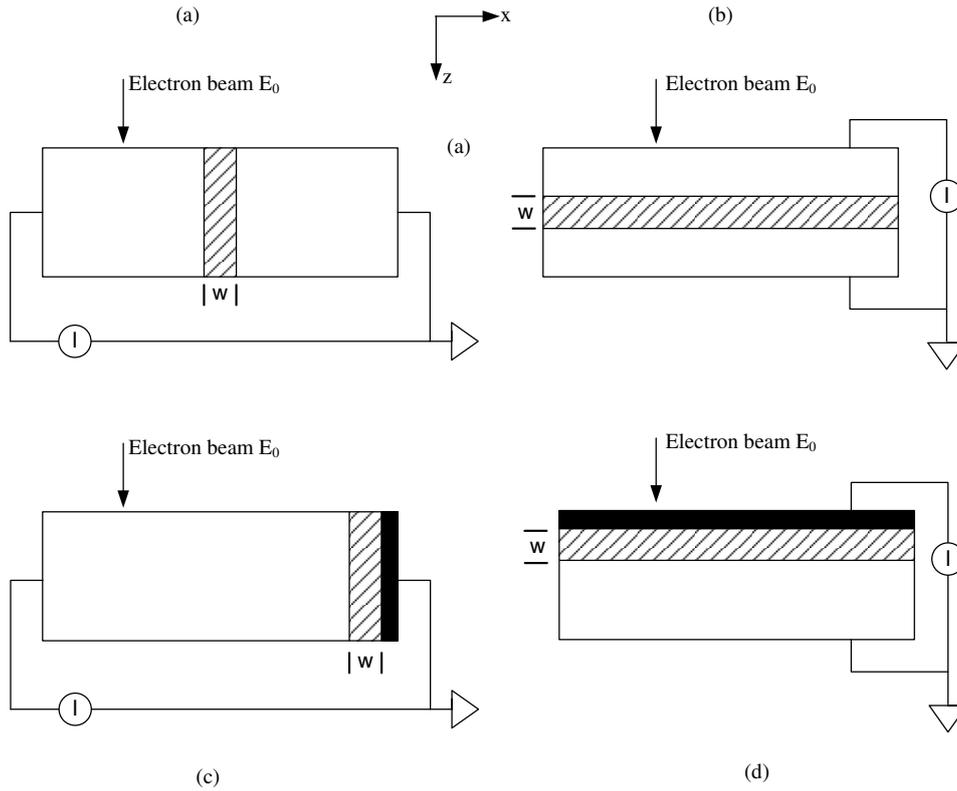


Figure13. Different configurations used for EBIC measurements, the shades area represents the built-in potential: (a) and (c) are normal configurations, (b) and (d) are planar configurations.

2.4.2 Calculation of the EBIC current in a normal-collector p-n junction configuration

One of the most used configurations is the normal p-n configuration, the electron beam scanned over the surface and the steady state EBIC is measured as a function of the distance x_b between the junction and the beam position (figure2-14.).

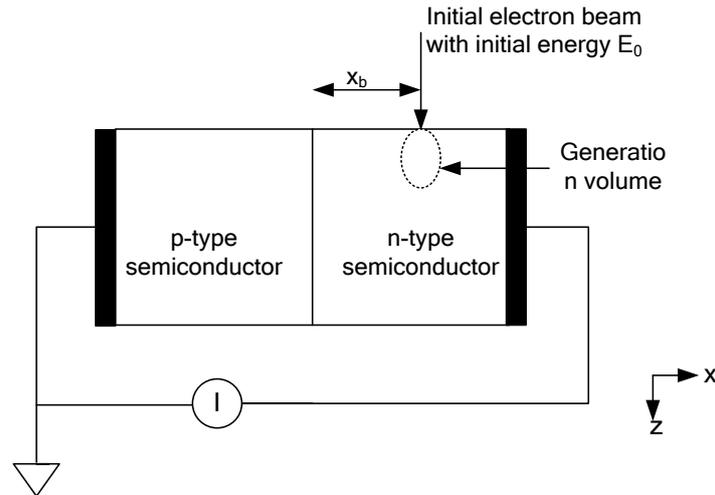


Figure2-14. Normal collector configuration of a p-n junction, with x distance between the beam and the junction.

2.4.2.1 The charge collection probability of Donolato

The model of Donolato [27] for the charge collection probability which is the EBIC due to a point source is detailed in this section.

In this model the presence of the back surface of the diode is neglected (sample thickness considered infinite) and the transport of the minority carriers generated by the electron beam in the neutral material (n type) is described by the steady state diffusion equation (2-16) [27].

According to [27], the configuration of figure2-14 has translational invariance along the y axis, in the sense that the contribution to the collected current of any source element doesn't depend on its y coordinate. Thus the collected current depends only on the projected generation on the xz plane:

$$g(x, z) = \int_{-\infty}^{+\infty} g(x, y, z) dy \quad (2-32)$$

Equation (2-16) becomes:

$$D_p \nabla^2 p(x, z) - \frac{p(x, z)}{\tau} + g(x, z) = 0 \quad (2-33)$$

The boundary conditions on the surface and at the junction plane are:

$$\begin{cases} p = 0 & \text{at } x = 0 \\ \frac{\partial p}{\partial z} = S \cdot p & \text{at } z = 0 \end{cases} \quad (2-34-a)$$

$$(2-34-b)$$

with S the normalized surface recombination velocity.

The solution of (2-33) is then:

$$p(x, z) = \int_0^{\infty} dx' \int_0^{\infty} g(x', z') G(x, x', z, z') dz' \quad (2-35)$$

where $G(x, x', z, z')$ is the Green's function for equation (2-33) satisfying the boundary conditions (2-34).

The charge collection probability $i(x', z')$ at a point (x', z') is :

$$i(x', z') = D \int_0^{\infty} \frac{\partial G}{\partial x} \Big|_{x=0} dz \quad (2-36)$$

Finally Donolato calculated the charge collection probability as follows:

$$i(x, z) = \exp(-\lambda x) - \frac{2S}{\pi} \int_0^{\infty} \frac{k}{\mu^2(\mu + S)} \exp(-\mu z) \sin(kx) dk \quad (2-37)$$

where $\lambda = 1/L$, L is the diffusion length of minority carriers, k is a constant, and $\mu = (k^2 + \lambda^2)^{1/2}$.

For more details about calculation of (2-37) the reader is referred to [27].

2.4.2.2 Calculation of the EBIC current

The approach proposed in [28,29] is used for the calculation of the EBIC profiles:

First, the current is divided into three regions as proposed by Czaja [28]: the n-region, the depletion region, and the p-region. In each region the current is a convolution of the generation volume distribution $g(x,z)$ and the charge collection probability $i(x,z)$ with the assumption that the charge collection within the depletion region is unity. The EBIC current when the beam is located at a point x' is then given by:

$$I(x') = \int_0^{\infty} \int_{-\infty}^{x_n} g(x-x',z)i(x,z)dx dz + \int_0^{\infty} \int_{x_n}^{x_p} g(x-x',z)dx dz + \int_0^{\infty} \int_{x_p}^{\infty} g(x-x',z)i(x,z)dx dz \quad (2-38)$$

where: x_n and x_p are the edges location of the depletion region and the x axis in (2-37) starts from the depletion region edge.

Then, a generation volume distribution proposed by Donolato is used to calculate the EBIC profiles, it is given by [28,29,30]:

$$g(r) = \frac{\phi(z/R)}{2\pi R \sigma^2(z,R)} \exp\left(\frac{-r^2}{2\sigma^2(z,R)}\right) \quad (2-39)$$

where:

R is the electron range related to the energy beam as follows:

$$R(\mu\text{m}) = 4 \times 10^{-2} \frac{E_0^{1.75}(\text{KeV})}{\rho} \quad (2-40)$$

- $\rho(\text{g/cm}^3)$ is the density of the sample.
- σ^2 is the standard deviation of the Gaussian distribution given by:

$$\sigma^2(z,R) = 0.36d^2 + 0.11z^3/R \quad (2-41)$$

- d is the beam diameter.

- $\phi(z/R)$ is the depth distribution, it follows the polynomial of Everhart and

Hoff:

$$\phi(z/R) = \begin{cases} 0.6 + 6.21(z/R) - 12.4(z/R)^2 + 5.69(z/R)^3 & 0 \leq z/R \leq 1.1 \\ 0 & z/R > 1.1 \end{cases} \quad (2-42)$$

According to (2-32) the equation (2-39) becomes:

$$g(x, z) = \frac{\phi(z/R)}{\sqrt{2\pi R \sigma}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (2-42)$$

Figure 2-15 illustrates EBIC profiles in a normal collector configuration calculated using the approach described above. The sample material is silicon (density of 2.33 g/cm^3), and the beam radius is of 10 nm.

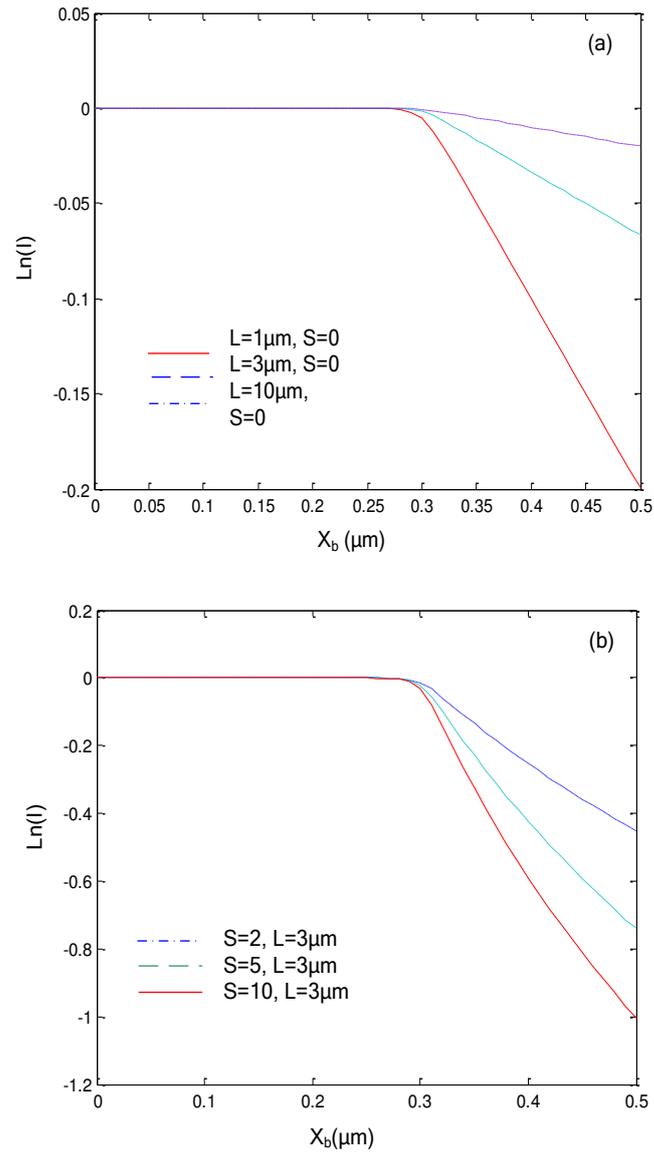


Figure 2-15- EBIC profiles for the region around the edges of a depletion layer located at $\pm 0.3\mu\text{m}$: (a) different values of the diffusion length L ($S=0$), (b) different values of the surface recombination velocity ($L=3\mu\text{m}$).

Chapter references

- 1- I. M. Watt, "The principles and practice of electron microscopy", Cambridge University Press, 2nd edition 1997.
- 2- D. B. Williams, C. B. Carter, "Transmission electron microscopy a textbook for materials science", Springer 2nd edition 2009.
- 3- S. J. B. Reed, "Electron Microprobe Analysis and Scanning Electron Microscopy in Geology ", Cambridge University Press 2nd edition 2005.
- 4- L. Reimer, H. Kohl, "Transmission Electron Microscopy, physics of image formation", Springer 5th edition 2008.
- 5- R. Oldfield, "Light Microscopy an illustrated guide", Wolfe Publishing 1994.
- 6- D. A. Newbury, D. C. Joy, P. Echlin, C. E. Fiori, and J. I. Goldstein, "Advanced scanning electron microscopy and X-ray microanalysis", Plenum Press, New York, 1986.
- 7- W. Williamson, Jr. and G. C. Duncan, "Monte Carlo simulation of nonrelativistic electron scattering" Am. J. Phys, 54(3), March 1986.
- 8- N. Tabet, "Contribution a l'étude des propriétés électriques de volume et des joints de grains dans le germanium application de la méthode du courant induit par faisceau d'électrons EBIC", a thesis submitted for the degree of Doctor of Philosophy, university of Paris Sud, Centre d'Orsay, 1988.
- 9- Course note:"Electron Microprobe Analysis by wavelength dispersive X-ray spectrometry", MIT Electron Microprobe Facility.
- 10- B. G. Yakobi, D. B. Holt, "Cathodoluminescence microscopy of inorganic solids", Plenum Press, New York, 1990.
- 11- G. Thomas, "Electron microscopy and structure of materials: proceedings", University of California Press, 1972.

- 12- J. Jimenez, "microprobe characterization of optoelectronic material", Taylor & Francis, 2003.
- 13- S. Tanuma, K. Nagashima, "Evaluation of an improved absorption correction based on the Gaussian ionization distribution model for quantitative Electron Probe Microanalysis", Mikrochimica Acta [Wien], 299-313, 1983.
- 14- E. Napchan, "Electron and Photon matter interaction :energy dissipation and injection levels", Revue de Physique Appliqué, colloque C6, supplément au n^o 6, tome 24, juin 1989.
- 15- C. Lamberti, "Characterization of semiconductor heterostructures and nanostructures", Elsevier, Oxford, UK 2008.
- 16- S. M. Sze, and K. K. Ng, "Physics of semiconductor devices", Wiley, 3rd edition, 2007.
- 17- D. Donolato, "On the theory of SEM charge-collection imaging of localized defects in semiconductors", Optik, 52(1978/79), No.1, 19-36.
- 18- F. Berz, and H. K. Kuiken, "Theory of life time measurements with the scanning electron microscope: steady state", Solid State Electronics, vol.19, pp.437-445.
- 19- A. Jakubowicz, "Transient cathodoluminescence of semiconductors in a scanning electron microscopy", Journal of applied physics. 58(11), pp 4354-4359, Dec 1985.
- 20- A. Jakubowicz, "Theory of cathodoluminescence contrast from localized defects in semiconductors", Journal of applied physics. 59(6), pp 2205-2209, March 1985.

- 21- T. S. Rao-Sahib and D. B. Wittry, "Measurement of diffusion lengths in p-type Galium Arsenide by electron beam excitation", Journal of applied physics. 40(9), pp 3745-3750, 1969.
- 22- W. Hergert, S. Hildebrandt, L. Pasemann, "Theoretical investigations of combined EBIC, LBIC, CL and PL experiments", Phys. Status. Solidi, (a) 102, 819(1987).
- 23- W. Hergert, and L. Pasemann, "Theoretical study of the information depth of the cathodoluminescence signal in semiconductor materials", Phys. Status. Solidi, (a) 85, 641(1984).
- 24- W. Hergert, P. Reck, L. Pasemann, and J. Schreiber, "Cathodoluminescence measurements using the scanning electron microscope for the determination of semiconductor parameters", Phys. Status. Solidi, (a) 101, 611(1987).
- 25- J. C. H. Phang, K. L. Pey, D. S. H. Chan, "A simulation model for cathodoluminescence in the Scanning Electron Microscopy". IEEE transactions on electron devices, vol 39(4), Apr 1992.
- 26- C. J. Wu, D. B. wittry, "investigation of minority-carrier diffusion lengths by electron bombardment of Schottky barriers". J. App. Phys, Vol. 49. (5). May. 1978.
- 27- C. Donolato, "On the analysis of diffusion length measurements by SEM", Solid State Electronics, Vol. 25, No. 11, pp. 1077-1081, 1982.
- 28- Kurniawan. O, Ong. V. K. S, "Choice of generation volume models for electron beam induced current computation", IEEE Transactions on Electron Devices 56(5), 1094-1099, 2009.

29- O. Kurniawan , "device parameters characterization with the use of EBIC",
Phd thesis (supervisor Pr O. V. Ong), Nanyang Technological University,
2008.

30- Nouar. Tabet, "Contribution a l'étude des propriétés électriques de volume et
des joints de grains dans le germanium application de la méthode du courant
induit par faisceau d'électrons EBIC", Phd thesis, University of Paris Sud
centre d'Orsay.

Chapter 3

Artificial Neural Networks

Contents

3.1 Introduction	
3.2 The neuron	
3.3 Neural network topology.....	
3.4 Feedforward network	
3.5 The learning process.....	
3.6 The learning algorithm	
3.7 ANN for function approximation.....	
Chapter references.....	

3.1 Introduction

Artificial neural networks ANN are mathematical creations inspired by the structure and performance of our biological neural networks, but they can not reach anywhere near there performance [1]. They have seen an explosion of interest over the last few years and received a lot of attention and are being successfully applied across a huge range of problems. They gained a wide spread into many applications including parameter extraction of semiconductor devices and integrated circuits. This is mainly due to many desirable characteristics they exhibit and which can be roughly summarized into the following points:

1. ANNs are able to learn by examples through training, that is, given a set of input/output training data, the ANN is able to extract the functional relationship between the input and the output data.

2. ANNs have the ability to generalize and predict, that is, the ANN is able to respond to input that have not been seen before as well as responding to incomplete or partial input data.
3. Due to the internal structure of the ANN, the computations are performed in parallel in each layer which offers significant reduction in computation time.

An ANN is defined as a massively parallel distributed mathematical model or computation model that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two aspects: knowledge is acquired by the network through a learning process, and interneuron connection weights known as synaptic weights are used to store the knowledge.

3.2 The Neuron

A biological neuron is the fundamental unit of our biological nervous system it consists of [1]:

- A cell body or soma.
- Nerve fibers called dendrites associated with the cell body; they receive signals from other neurons.
- A single long fiber called axon that branches into connections to other neurons.

We can consider an artificial network as a very simplified model of the biological neural network, the basic building block of an ANN is the neuron. It is the information-processing unit which is fundamental to the operation of an ANN. A typical artificial neuron has more than one inputs and only one output as shown in figure3-1 [2].

A typical artificial neuron, as described by figure3-1 consists of [1,2,3]:

- Input signals x_i .
- Synaptic weights w_i : they are factors that weight each input x_i .
- Summing junction in which the linear combiner output is calculated as [2]:

$$a = \sum_{\text{all inputs}} w_i x_i \quad (3-1)$$

- An activation or transfer function φ .
- A threshold or bias θ : it has the effect of lowering (negative bias) or increasing (positive bias) the net input of the activation function.
- The neuronal output is then:

$$y = \varphi(a + \theta) \quad (3-2)$$

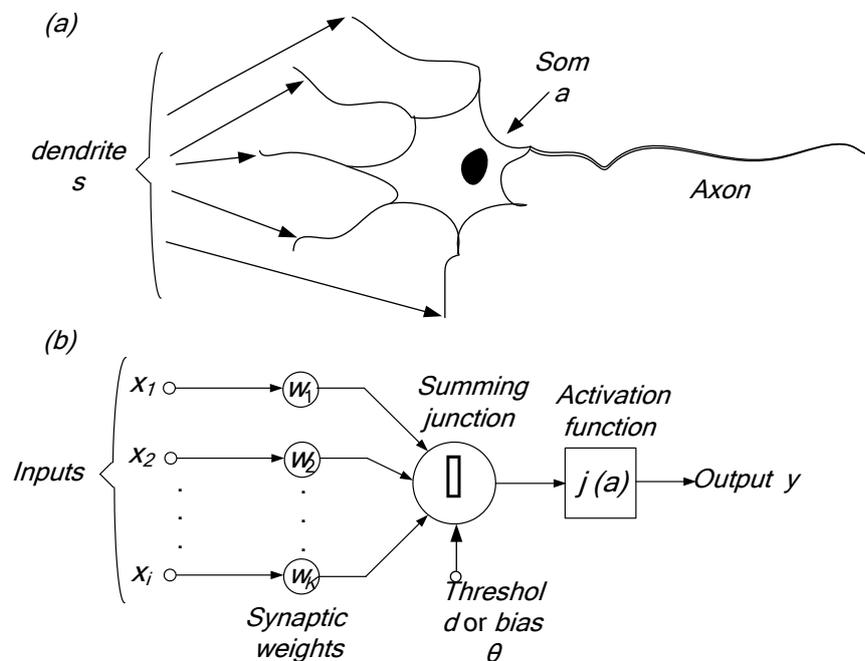


Figure3-1 – Biological neuron (a), and an artificial neuron (b).

The pair of equations (3-1) and (3-2) is the mathematical description of the neuron.

There are several types of activation functions, the basic types are [2]:

1. **Threshold function;** for this type shown in figure3-2-a we have:

$$\varphi(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ 0 & \text{if } a < 0 \end{cases} \quad (3-3)$$

2. **Piecewise-linear function.** This type, depicted in figure3-2-b, is defined as [2]:

$$\varphi(a) = \begin{cases} 0 & \text{if } a \leq a_{\min} \\ m \times a + b & \text{if } a_{\min} < a < a_{\max} \\ 1 & \text{if } a \geq a_{\max} \end{cases} \quad (3-4)$$

3. **Sigmoid function.** This type which has an "S" shaped graph is the most used one. An example of the sigmoid function is depicted in figure3-2-c it is the logistic function [2] defined by:

$$\varphi(a) = \frac{1}{1 + \exp(\beta \times a)} \quad (3-5)$$

where β is the slope parameter of the sigmoid function.

The activation functions above range from 0 to +1, they can range from -1 to +1.

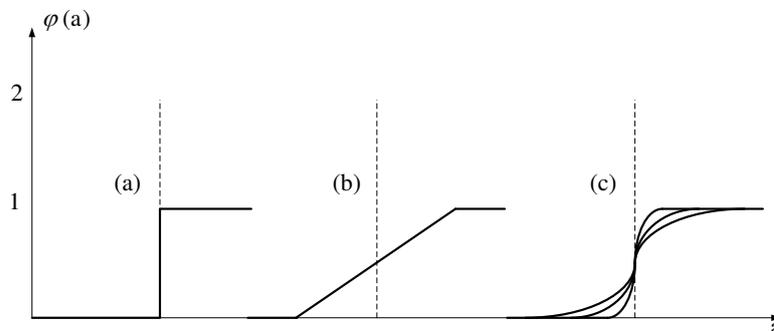


Figure3-2 – Types of activation functions: (a) Threshold function, (b) Piecewise-linear function, (c) Sigmoid function.

3.3 Neural Network topology

Artificial neural networks are practical only when the processing units (neurons) are structured in appropriate manner to accomplish a given task.

ANNs are in general organized into layers of processing units; the units of a layer are similar in the sense of having all the same activation dynamics and output function.

Connections can be made from the units of one layer to the units of another layer it is an "interlayer connections", or among the units within the same layer it is an "intralayer connections", or both.

Further, the pattern of connections and the propagation of data can be in a "feed-forward" or "feedback" manner:

Feed-Forward networks, for this type, neurons are organized into layers that have connections between them only in one direction from input units to output units.

Feedback or recurrent networks, contrary to the feed-forward networks, the data can travel in both directions, all possible connections between neurons are allowed.

We are interested in our work to the first type: the feedforward networks, its description is detailed below.

3.4 Feedforward networks

3.4.1 Single layer feedforward network

It is the simplest form; we have an input layer of neurons (source nodes) that project directly in an output layer of neurons (computation nodes) as illustrated in figure3-3. The input layer is not counted because no computation is performed there.

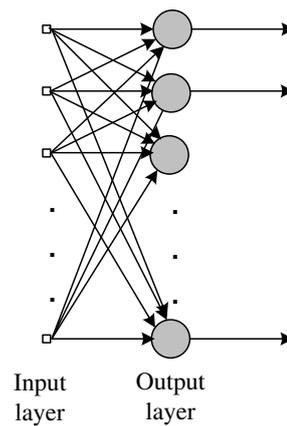


Figure3-3- Feedforward network with a single layer.

3.4.2 Multilayer feedforward network

The multilayer feedforward network (figure3-4) is an important class of neural networks. We have an input layer of neurons (source nodes), one or more layers of neurons called hidden layers, and an output layer. The signal or the data propagates through the network in a forward direction, on a layer by layer basis [2].

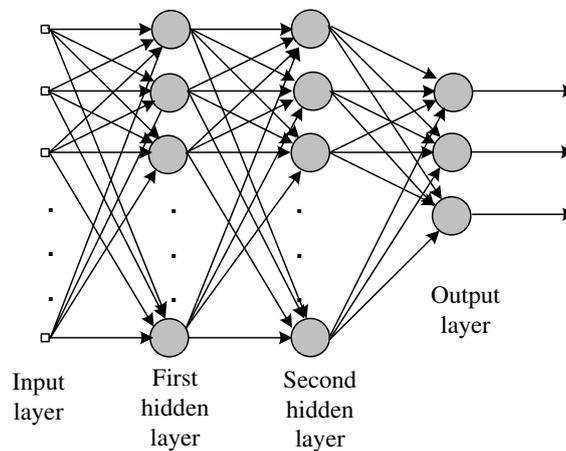


Figure3-4- Typical feedforward network with two hidden layers and an output layer.

3.5 The Learning Process

The ability of ANN to learn is a fundamental characteristic, the learning process in ANN can be viewed as the problem of updating the connection weights and the biases so that the network can efficiently perform a task. To understand or design a learning process we need to know: first, learning paradigm or learning set which is the information available to the network, second, learning rules which is how the weights are updated. Third, the learning algorithm which is the procedure in which learning rules is used for adjusting the weights.

The learning can be:

A supervised learning; it is like learning with a teacher in which the network is provided with a correct answer (output) for every input pattern in the training data set.

An unsupervised learning; it is like learning without a teacher the network doesn't require a correct answer (output) for every input pattern in the training data set.

A hybrid learning; this is a combination of the supervised and unsupervised learning.

3.6 The Learning algorithm

As revealed above, the learning algorithm is the procedure in which learning rules are used to adjust the weights in an orderly way so as to attain a desired objective or task. Different types of learning algorithms have been developed for supervised learning (least mean squares LMS algorithm and Backpropagation (BP) algorithm) and unsupervised learning (Hebbian learning and competitive learning).

For the supervised learning we will talk about two algorithms that are the most popular and the most used [3,4]: the BP algorithm and the Levenberg Marquardt (LM) algorithm.

Since we used the LM algorithm in this work, only a brief introduction to the BP will be provided here, for more details the reader is referred to [3,4],

3.6.1 The Backpropagation Algorithm

The Backpropagation algorithm (BP) appears to be a popular algorithm used for feedforward networks. The BP is an iterative gradient based algorithm that minimizes an error between the actual output of the network and the desired output. It can be summarized into the following steps:

Step1- The network is initialized by setting up all the weights to small random numbers.

Step2- The input pattern is applied and the output is calculated this is called forward pass, since the weights are random, the calculation gives an output completely different to the desired one (the target).

Step3- The error of each neuron is then calculated as: desired output – actual output.

Step4- This error is then used mathematically to change the weights in such a way that the error will get smaller, i.e. the output of each neuron will get closer to its target this is called the reverse pass.

Step5- The process is repeated until the error is minimized

The BP has some many disadvantages; the most known ones are:

- Slow convergence: compared to other training algorithms, the BP exhibits a slow convergence behavior. This is mainly because it is gradient descent algorithm and doesn't exploit the second derivative information to accelerate the convergence speed.
- Convergence to a local minimum: because it is a gradient descent algorithm, it will always follow the direction of the negative of the gradient and therefore this will not guarantee finding the global minimum of the error surface which is in general a multimimima function..

3.6.2 The Levenberg Marquardt algorithm

The Levenberg Marquardt (LM) algorithm is an approximation to the Newton method, the last approximates the error of the network using a second order derivative expression (in contrast with the BP that does it with a first order derivative expression). The LM is the most popular algorithm for solving non linear least square optimization problems.

For ANN training the objective function is an error function given by:

$$E = \sum_{k=1}^p \frac{1}{2} (\tilde{y}_k - y_k)^2 \quad (3-6)$$

where: \tilde{y}_k and y_k are the actual and the desired output, respectively, for the k^{th} pattern. And p is the total number of training patterns.

The training using an LM algorithm consists of the following steps [3]:

Step1- Presenting all the inputs to the network and computing the corresponding network outputs and errors. Then calculate the mean square error from (3-6)

Step2- Calculating the Jacobian matrix $J(w)$ of the weights.

Step3- Solving the LM weight update equation as:

$$\Delta w = [J^T(w)J(w) + \mu I]^{-1} J^T(w)E_r \quad (3-7)$$

where:

$J^T(w)J(w)$ is the Hessian matrix.

I is the identity matrix.

μ is the learning parameter.

E_r is the error vector of size p given by: $E_r = [\tilde{y}_1 - y_1, \dots, \tilde{y}_p - y_p]^T$.

Step4- Calculating the error using the updated weights $w + \Delta w$:

- If the new error is smaller than that calculated in **step 1** → reduce the training parameter μ and → go back **step 3**.
- If the error is not reduced → increase the training parameter μ and → go back **step 3**.

Step5- The algorithm is converged when the norm of the gradient is less than some predetermined value, or when the error has been reduced to some error goal.

3.7 ANN for function approximation

The function approximation is the task of interest in this work. Let's consider a nonlinear functional relationship:

$$\mathbf{Y} = F(\mathbf{X}) \quad (3-8)$$

where the vector \mathbf{X} is the input and the vector \mathbf{Y} is the output.

The function F is assumed to be unknown, what is known is a set of input/output data pairs obtained by applying a set of data inputs to this unknown function and recording the outputs, that is:

$$\mathfrak{T} = \{(x_i, y_i)\} \quad i = 1, p \quad (3-9)$$

The target is to design a neural network that approximates enough the unknown function F . Two main applications of function approximation are system identification and inverse modeling, these applications are detailed below [2]:

3.7.1 System identification

The ANN is trained by the set of examples (3-9), lets \tilde{y}_i be the actual output of the neural network calculated in response to the input x_i . The difference between y_i (associated with x_i) and \tilde{y}_i gives the error signal e_i (figure3-5). This error signal is used to minimize the squared difference between the output of the unknown system Y and the ANN output \tilde{Y} in a statistical sense over the entire training set.

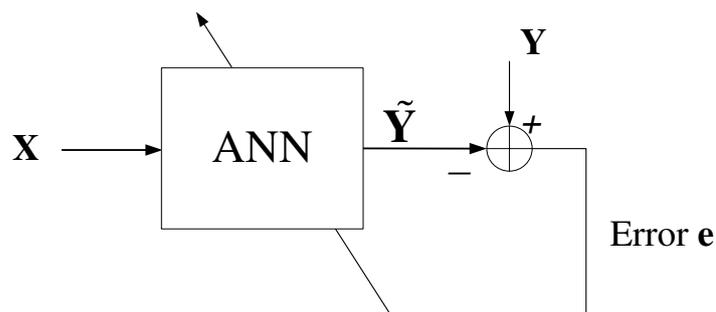


Figure3-5- ANN system identification

3.7.2 Inverse modeling

The requirement here is to build an inverse model that produces \mathbf{X} in response to \mathbf{Y} as:

$$\mathbf{X} = \mathbf{F}^{-1}(\mathbf{Y}) \quad (3-10)$$

where \mathbf{F}^{-1} denotes the inverse of \mathbf{F} .

The roles of x_i and y_i are interchanged: y_i used as input, and x_i as desired output and e_i is the error between x_i and \tilde{y}_i . e_i is used to minimize the squared error between the output of the unknown inverse system and the ANN output in statistical sense over the entire training set (figure3-6).

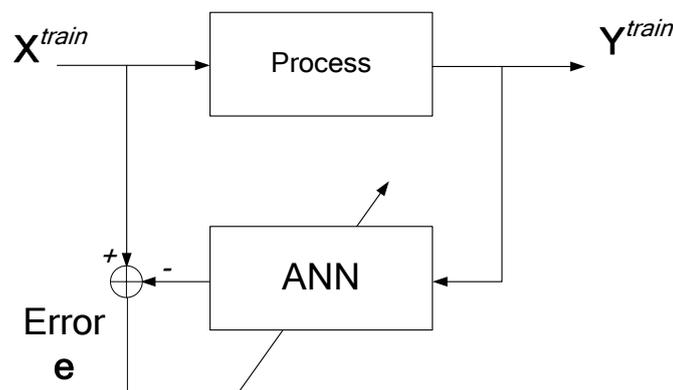


Figure3-6- ANN inverse modeling

Chapter references

- 1- B. Yegnanarayana, "Artificial Neural Networks", Prentice Hall of India, New Delhi. 2006.
- 2- S. Haykin, "Neural Networks; A Comprehensive Foundation", Prentice Hall; 2nd edition, 1999.

- 3- R. M. Hristev, "The ANN book".1st edition 1998.
- 4- Tommy W. S. Chow, and Siu-Yeung Cho " Neural networks and computing: learning algorithms and applications", Imperial College Press. London, 2007.

Chapter 4

Genetic Algorithms

Contents

4.1 Introduction	
4.2 The GA operators	
4.3 The GA parameters	
4.4 The continuous GA algorithm.....	
Chapter references.....	

4.1 Introduction

Recently, genetic algorithms (GA) [1] have appeared as an effective search method to find good near-optimal solutions to complicated problems. GA are widely used in business, science and engineering. GA is an optimization search technique based on the principles of genetics and natural selection, initially developed by John Holland in 1970s. GA optimization starts with a set of solutions called population, each solution in the population is presented by a vector called chromosome (or individual), and each variable in the chromosome is called a gene. The population evolves under specified rules to a state that minimizes the cost function.

A GA can find good solutions in vast search spaces with a reasonable computational cost due to its characteristics such as [1,2]:

1. GA doesn't require derivative information.
2. Simultaneously searches over a wide search space.
3. Well suited for parallel computers due to its inherent parallelism.

4. Can escape from local minima.

4.2 The GA operators

The simplest form of a GA involves three main operators [1,2,3]:

4.2.1 Selection

The selection is the first operator applied on a population to make the decision of which chromosomes are fit enough to survive and reproduce new chromosomes or offspring in the next generation. There are different methods (functions) how to select the best chromosomes [1,2]: roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection, just to name a few.

4.2.2 Crossover

This operator is the process that combines or mates two chromosomes "parents" to produce a new "child chromosome" or offspring which is some combination of them. The new chromosome may be better than both the parents if it takes the best characteristic from each of the parents. The crossover occurs during evolution according to a crossover probability defined by the user. There are many types of crossovers: one point, two points, uniform, arithmetic, and others.

4.2.3 Mutation

This operator is the random change in some of the genes in order to force the GA to explore other areas of the cost surface and avoid the problem of early convergence at local minimum. This operator introduces diversity in the population whenever the population tends to become homogenous due to repeated use of reproduction and crossover. The probability that a gene may become mutated is called the mutation probability.

4.3 The GA parameters

The GA parameters related to the GA operators and stopping criteria used in this work can be summarized as:

4.3.1 Population options

- **The chromosome length:** specifies the number of variables used to optimize the objective function.
- **Initial population size:** specifies the number of chromosomes in the initial generation. Using a large population size the GA searches the cost surface more carefully which reduce the chance of returning a local minimum, however with a large population size the GA runs slowly.
- **Creation function:** specifies the function that is used to generate the initial population. In our work the creation function is a Uniform distribution one; it creates a random initial population with a uniform distribution.

4.3.2 Fitness scaling options

These options permit the conversion of the raw fitness scores that are returned by the fitness function to values in a range that is suitable for the selection function.

- **Fitness scaling function:** in our work the fitness scaling function was a Rank one; this function scales the raw scores based on the rank of each individual instead of its score (The rank of an individual is its position in the sorted scores). Rank fitness scaling removes the effect of the spread of the raw scores.

4.3.3 Selection options

Selection options specify how the genetic algorithm chooses parents for the next generation

- **Selection function:** In our work, the selection function is the roulette wheel; it selects parents by simulating a roulette wheel (the area of the section of the wheel corresponding to an individual is proportional to the individual's expectation). The algorithm uses a random number to select one of the sections with a probability equal to its area.
- **Elite count:** specifies the number of individuals with the best fitness values in the current generation that are passed to the next generation without any modification.

4.3.4 Crossover options

- **Crossover function:** specifies the function used to perform crossover, in our work the function scattered is used; it creates a random binary vector, selects gene from the first parent if the vector is 1 and from the second parent if the vector is 0, then combines the genes to form the child.
- **Crossover fraction:** specifies the fraction of the next generation, other than elite individuals, that are produced by crossover.

4.3.5 Mutation options

Mutation options specify how the genetic algorithm makes small random changes in the individuals in the population to create mutation children. Mutation

provides genetic diversity and enables the genetic algorithm to search a larger space.

- **Mutation function:** specifies the function that is used to perform mutation. In our work the Gaussian function is used; it adds a random number taken from a Gaussian distribution with mean 0 to each entry of the parent vector. The standard deviation of this distribution is determined by two parameters: the scale parameter, which determines the standard deviation at the first generation and the shrink parameter which controls how the standard deviation shrinks.

4.3.6 Stopping criteria options

The stopping criteria determine the conditions that should be satisfied to terminate the algorithm. For genetic algorithms, the most important stopping criteria are:

- **Number of generations:** specifies the maximum number of generations the GA is allowed to achieve.
- **Stall generation:** specifies the maximum number of generations for which the GA stops if the weighted average change in the fitness function value is less than another parameter called **function tolerance**.
- **Function tolerance:** specifies the cumulative change in the fitness function over stall generation under which the GA algorithm stops running.

For more details about the selection of the GA parameters, the reader is referred to [1]

4.4 The continuous GA algorithm

Depending on the variables (genes) representation, a GA is [1]:

- Binary when the variables are represented by an encoded binary string and works with the binary strings to minimize the cost function.
- Continuous when no binary encoding is used, and the variables themselves are used to minimize the cost function.

Since we used in this work a simple continuous GA, a detailed description of this type is provided here. For other types of Gas, the reader is referred to [1,3].

Solving an optimization problem using a simple continuous GA is performed through the following steps [1]:

1. We start by defining a chromosome as an array of variables. If the chromosome has N variables it is written as an array of $1 \times N$ elements:

$$\text{chromosome} = [p_1, p_2, \dots, p_N] \quad (4-1)$$

each chromosome has a cost found by evaluating the cost function f at the variables p_1, p_2, \dots, p_N .

$$\begin{aligned} \text{cost} &= f(\text{chromosome}) \\ &= f(p_1, p_2, \dots, p_N) \end{aligned} \quad (4-2)$$

Equations (4-1) and (4-2) constitute the problem to be solved.

2. Defining an initial population of N_{pop} chromosomes which is a $N_{\text{pop}} \times N$ matrix of random uniformly distributed values between 0 and 1.

The chromosomes are passed to the cost function for evaluation.

3. The chromosomes in the initial population that are fit enough can survive and reproduce offspring in the next generation. The N_{pop} costs and associated chromosomes are ranked from lowest cost to highest cost; the rest die off. This process of "selection" occurs at each iteration of the

algorithm and permits the evolution of the population of chromosomes to the most-fit elements as defined by the cost function.

The fraction of N_{pop} that survives for the next step is the selection rate X_{rate} , and the number of chromosomes that are kept in each generation is:

$$N_{\text{keep}} = X_{\text{rate}} \times N_{\text{pop}} \quad (4-3)$$

of the N_{pop} chromosome in a generation only N_{keep} survive and $N_{\text{pop}} - N_{\text{keep}}$ are discarded to make place for the new offspring.

4. The most-fit selected chromosomes N_{keep} pair or mate in some random way as parents and each pair produces two offsprings that are in fact some combination of them.
5. To avoid the fast convergence of the GA algorithm before sampling the whole cost surface, which can end up in a local minimum; the algorithm is forced to explore other areas of the cost surface by randomly introducing changes in some of the variables through the "mutation" operator
6. The process described previously is iterated until a satisfactory solution is found

A summary of a simple continuous GA is provided below:

- **Step1:** define variables, GA parameters and options.
- **Step2:** define the cost function or objective function.
- **Step3:** a set of initial solutions (population of chromosomes) is generated randomly over the search space.
- **Step4:** each chromosome in the population is passed to the cost function for calculating its fitness.
- **Step5:** a new population of chromosomes (next generation) is then created using the three operators:

- Selection.
- Crossover.
- Mutation.
- **Step6:** replace the current population with the new one.
- **Step7:** go to step3.
- This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

Chapter references

1. R. L. Haupt and S. E. Haupt, "Practical Genetic Algorithms", 2nd edition, Jhon Wiley and sons.
2. M. Mitchell, "An Introduction to Genetic Algorithms", The MIT press. 1999.
3. S. N. Sivanandam, S. N. Deepa "Introduction to Genetic Algorithms", Springer. 2007.

Chapter 5

Semiconductor parameter extraction using Artificial Neural Networks and exhaustive search

Contents

5.1 Introduction	
5.2 Parametre extraction based on ANN and exhaustive searsh	
5.3 Application to cathodoluminescence	
5.4 Application to EBIC	
5.5 Conclusion.....	
Chapter references.....	

5.1 Introduction

To the best of our knowledge, no work has been reported regarding the use of Artificial Neural Networks (ANN) in semiconductor parameter extraction from EBIC/CL signals. In this chapter a new technique based on ANN is presented and discussed; The ANN is used here for function approximation or in other words for the identification of the EBIC/CL signal generation process given a set of input parameters characterizing the EBIC/CL signal generation (see chapter3 paragraph 3-7). After the identification of the EBIC/CL signal generation process using an ANN, the latter is used to generate extra EBIC/CL signals by exploiting the generalization property of ANNs and finally an exhaustive search algorithm is used to perform parameter extraction.

This parameter extraction algorithm is first applied to the simultaneous extraction of four related semiconductor parameters, that is: diffusion length L , absorption coefficient α , dead layer thickness Z_t , and relative quantum efficiency Q , from any steady state CL signal of a free defect semi-infinite semiconductor. The effect of noise measurement is discussed as well. Then, the parameter extraction algorithm is applied to the simultaneous extraction of two related semiconductor parameters that are diffusion length L and normalized surface recombination velocity S from any EBIC line scan in a normal collector configuration.

5.2 Parameter Extraction based on ANN and exhaustive search

Suppose that we have a nonlinear functional relationship (process) given by: $\mathbf{Y} = F(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)$; where \mathbf{X}_i $i=1, \dots, n$ represents the input vector and \mathbf{Y} is the output vector (signal). The parameter extraction algorithm based on a feedforward ANN, and an exhaustive search technique can be summarized in the following phases:

5.2.1 Preparation of the training and test data sets:

To train the ANN, the training data set is prepared by sampling and stacking the parameters (X_1, X_2, \dots, X_n) into vectors as follows: $\mathbf{X}_1^{\text{train}} = [X_{1,1}^{\text{train}}, X_{1,2}^{\text{train}}, \dots, X_{1,d_1}^{\text{train}}]$, $\mathbf{X}_2^{\text{train}} = [X_{2,1}^{\text{train}}, X_{2,2}^{\text{train}}, \dots, X_{2,d_2}^{\text{train}}]$... $\mathbf{X}_n^{\text{train}} = [X_{n,1}^{\text{train}}, X_{n,2}^{\text{train}}, \dots, X_{n,d_m}^{\text{train}}]$, where there dimensions are given by $\{1\text{-by-}d_1\}$, $\{1\text{-by-}d_2\}$... $\{1\text{-by-}d_m\}$ respectively. All possible combinations of the vectors $\mathbf{X}_1^{\text{train}}, \mathbf{X}_2^{\text{train}} \dots \mathbf{X}_n^{\text{train}}$ form the input training data set $\mathbf{X}^{\text{train}}$ of dimension $\{n\text{-by-}d_1 \times d_2 \times \dots \times d_m\}$. For each column of $\mathbf{X}^{\text{train}}$ the \mathbf{Y} signal is calculated/measured. The results are stored in another matrix $\mathbf{Y}^{\text{train}}$ (output training data set).

For the test data set the parameters (X_1, X_2, \dots, X_n) are sampled and stacked into vectors by taking the mid-value between each two successive values of the training

set: $\mathbf{X}_1^{\text{test}} = [X_{1,1}^{\text{test}}, X_{1,2}^{\text{test}}, \dots, X_{1,d_1-1}^{\text{test}}]$, $\mathbf{X}_2^{\text{test}} = [X_{2,1}^{\text{test}}, X_{2,2}^{\text{test}}, \dots, X_{2,d_2-1}^{\text{test}}] \dots$

$\mathbf{X}_n^{\text{test}} = [X_{n,1}^{\text{test}}, X_{n,2}^{\text{test}}, \dots, X_{n,d_m-1}^{\text{test}}]$. The dimensions of the vectors $\mathbf{X}_1^{\text{test}}$, $\mathbf{X}_2^{\text{test}}$... $\mathbf{X}_n^{\text{test}}$ are given by {1-by-(d₁-1)}, {1-by-(d₂-1)}... {1-by-(d_m-1)}, respectively. All their possible combinations form the input test data set \mathbf{X}^{test} of dimension {n-by-(d₁-1)×(d₂-1)... ×(d_m-1)}. For each column of the matrix \mathbf{X}^{test} , the \mathbf{Y} signal is calculated/measured and the results are stored in a matrix \mathbf{Y}^{test} (output test data set).

5.2.2 Training the ANN algorithm:

The ANN is trained to learn the functional relationship f between the input data set $\mathbf{X}^{\text{train}}$ and the output data set $\mathbf{Y}^{\text{train}}$ (see § 3-7 in chapter 3), that is:

$$\mathbf{X}^{\text{train}} \xrightarrow{f} \mathbf{Y}^{\text{train}} \quad (5-1)$$

where the ANN receives the input data set $\mathbf{X}^{\text{train}}$ and calculates the actual output data set $\mathbf{Y}^{\text{actual}}$ which is then subtracted from the desired output data set $\mathbf{Y}^{\text{train}}$ resulting in an error \mathbf{e} . The latter is then used to update the ANN weights. This process is repeated a number of epochs till the error (sum squared error SSE) goes beyond a predefined threshold, the SSE is given by the following formula [1] :

$$\text{SSE} = \sum_{j=1}^{k^{\text{train}}} (Y_j^{\text{train}} - Y_j^{\text{actual}})^2 \quad (5-2)$$

where k^{train} represents the number of samples of $\mathbf{Y}^{\text{train}}$.

5.2.3 Testing the ANN algorithm

Once the ANN has been trained, its ability to generalize is tested by applying a new set of input values \mathbf{X}^{test} not seen before. The signal $\tilde{\mathbf{Y}}^{\text{test}}$ calculated by the ANN is then compared to the desired data set \mathbf{Y}^{test} and the sample-by-sample

percentage error between \mathbf{Y}^{test} and $\tilde{\mathbf{Y}}^{\text{test}}$ is calculated using the following formula [1]:

$$e_{i,j} = \left| \frac{\mathbf{Y}_{i,j}^{\text{test}} - \tilde{\mathbf{Y}}_{i,j}^{\text{test}}}{\mathbf{Y}_{i,j}^{\text{test}}} \right| \times 100 \quad (5-3)$$

where i and j indicate the i^{th} row, j^{th} column of the matrices \mathbf{Y}^{test} and $\tilde{\mathbf{Y}}^{\text{test}}$, respectively.

5.2.4 Oversampling of the signal using ANN

The main feature of ANNs that will be exploited is their ability to generalize, hence this property is used to obtain more samples (oversampling) of the \mathbf{Y} signal. This comes with a low computation load compared to the case of obtaining these extra samples by experimentation.

In this phase, the output $\tilde{\mathbf{Y}}^{\text{over}}$ calculated by ANN is compared to the desired data output \mathbf{Y}^{over} and the sample-by-sample percentage error between \mathbf{Y}^{over} and $\tilde{\mathbf{Y}}^{\text{over}}$ is then calculated using (5-3). Note that the input data matrix \mathbf{X}^{over} here is obtained by oversampling the vectors of the training data set $\mathbf{X}_1^{\text{train}}, \mathbf{X}_2^{\text{train}} \dots \mathbf{X}_n^{\text{train}}$ where more values are taken between each two successive values.

5.2.5 Parameter Extraction through exhaustive search

After oversampling the \mathbf{Y} signal, the parameter extraction of a set of input parameters is now straightforward. The procedure is as follows: the oversampled input data set \mathbf{X}^{over} and the oversampled data set output $\tilde{\mathbf{Y}}^{\text{over}}$ are used as a database for the exhaustive search. A randomly selected value of the signal \mathbf{Y} : y^* obtained for example from an experiment, is selected and the Euclidian distance between the latter and all other columns of matrix $\tilde{\mathbf{Y}}^{\text{over}}$ is calculated. The values of

the parameters $(X_1, X_2 \dots X_n)$ corresponding to the column of matrix $\tilde{\mathbf{Y}}^{\text{over}}$ with the smallest distance are selected and considered as the extracted parameters, that is:

$$\arg \min_{j=1,2,\dots,K^{\text{over}}} \left\| \mathbf{y}^* - \tilde{\mathbf{Y}}_j^{\text{over}} \right\|_2 \quad (5-4)$$

where K^{over} is the number of columns of $\tilde{\mathbf{Y}}^{\text{over}}$.

In order to evaluate and qualify the performance of the parameter extraction algorithm, a large set of randomly selected signals for which the input parameters $(X_1, X_2 \dots X_n)$ are known, is generated and used to test the parameter extraction algorithm. The percentage error between the true input parameter (nominal value) and the parameter determined using the proposed parameter extraction algorithm is given by:

$$e_{i,j} = \left| \frac{\mathbf{X}_{i,j}^{\text{rand}} - \mathbf{X}_{i,j}^{\text{over}}}{\mathbf{X}_{i,j}^{\text{rand}}} \right| \times 100 \quad (5-5)$$

We can summarize the different steps of the parameter extraction algorithm in table 5.1

Table 5.1. Summary of different steps of the parameter extraction algorithm based on ANN and exhaustive search

Step	Description
1- Preparation of the training and test data sets	Obtain the input ($\mathbf{X}^{\text{train}}$) and output ($\mathbf{Y}^{\text{train}}$) training data sets and the input (\mathbf{X}^{test}) and output (\mathbf{Y}^{test}) test data sets, respectively, through theoretical calculation simulation or experimentation.
2- Training the ANN	Train the ANN using the input/output training data sets ($\mathbf{X}^{\text{train}}, \mathbf{Y}^{\text{train}}$).
3- Testing the ANN	Test the ANN's ability to generalize by applying a new input/output data set ($\mathbf{X}^{\text{test}}, \mathbf{Y}^{\text{test}}$) not seen before.
4- Oversampling the signal using ANN	Obtain more signal samples $\tilde{\mathbf{Y}}^{\text{over}}$ using the ANN trained and tested in steps 1 and 2, respectively.
5- Exhaustive search and parameters extraction	Given the signal y^* for which the parameters (X_1, X_2, \dots, X_n) are to be extracted, choose the values of the parameters (X_1, X_2, \dots, X_n) corresponding to the column of the matrix $\tilde{\mathbf{Y}}^{\text{over}}$ with the smallest distance that is : $\arg \min_{j=1,2,\dots,K^{\text{over}}} \ \mathbf{y} - \tilde{\mathbf{Y}}_j^{\text{over}}\ _2$ where K^{over} is number of columns of $\tilde{\mathbf{Y}}^{\text{over}}$

5.3 Application to cathodoluminescence

In this section we present the application of the parameter extraction algorithm described above to the simultaneous extraction of: diffusion length L , absorption coefficient α , dead layer thickness Z_t , and relative quantum efficiency Q

from a steady state CL signal of a free defect semi-infinite semiconductor (Hergert's et al model; see chapter2). The effect of noise measurement is discussed here as well.

In all the following, we consider an n-type GaAs semiconductor its critical angle is equal to: 16° and its atomic number is equal to: 32, and the electron beam is perpendicular to the surface of the semiconductor sample.

5.3.1 Training the ANN

The first step in the proposed algorithm is to collect the input/output data pairs used to train the ANN. The data were collected from the evaluation of Hergert's et al model described in chapter 2. However, Monte Carlo simulations and experimental data can be used as well to form the input/output data pairs for the training of ANN. The CL signal was calculated for each combination of the parameters α^{train} , L^{train} , Zt^{train} , Q^{train} for different values of the energy beam. Table5.2 details the values of the semiconductor parameters used to generate the CL signal and consequently form the input/output data pairs used for training the ANN. For α^{train} , L^{train} , Zt^{train} , Q^{train} , the values in [2] were used here as well, the value of S is fixed to (S=10).

Table5.2. Semiconductor parameters used for training (CL)

Material Parameter	Minima l value	Maxima l value	Samplin g step size	Number of samples
Energy beam E (KeV)	10	50	10	5
Diffusion length L^{train} (μm)	0.73	0.78	0.01	6
Absorption coefficient α^{train} (μm^{-1})	0.785	0.825	0.01	5
Dead layer thickness Zt^{train} (μm)	0.03	0.08	0.01	6
Relative quantum efficiency Q^{train} (arbitrary units)	7.15	7.45	0.05	7
Total number of samples used for training equals: number of samples of L^{train} \times number of samples of α^{train} \times number of samples of Zt^{train} \times number of samples of Q^{train} ($6 \times 5 \times 6 \times 7 = 1260$).				

As illustrated in Table5.2, taking all the possible combinations of α^{train} , L^{train} , Zt^{train} , Q^{train} results in a total number of samples of 1260 samples, therefore, 1260 input/output pairs are used for training the ANN.

A feed forward neural network consisting of one hidden layer is used, the hidden layer comprises 5 neurons and uses logarithmic sigmoid transfer functions whereas the output layer consists of 5 neurons (corresponds to the number of samples of the energy beam vector) and uses linear activation functions. Table5.3 illustrates the ANN parameters used here. The learning algorithm used here is the Levenberg Marquardt (see chapter 3).

Table5.3- ANN parameters used for training (CL)

ANN parameters		
Number of epochs	600	
Desired goal	10^{-6}	
Number of hidden layers	1	
Number of neurons in the hidden layer	5	uses logarithmic sigmoid transfer functions
Number of neurons in the output layer	5 (number of samples of the energy beam)	uses linear transfer functions

The training curve of the ANN is shown in figure 5.1; the desired goal was achieved after 434 epochs.

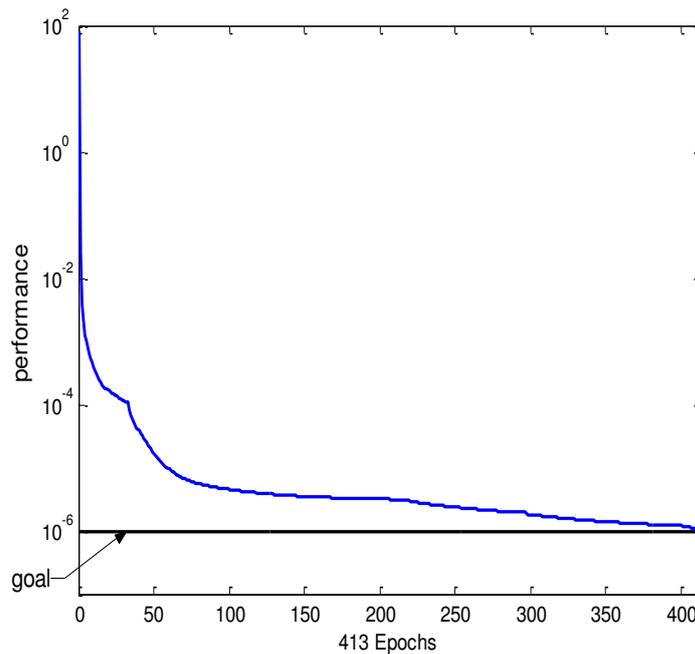


Figure5.1 Training curve of the ANN (CL)

Figure5.2 shows the percentage error between the theoretical training CL signal and the CL signal calculated by the ANN for five values of the energy beam. The maximal

percentage error for all the parameters is 0.25%, which indicates that the training is satisfactory.

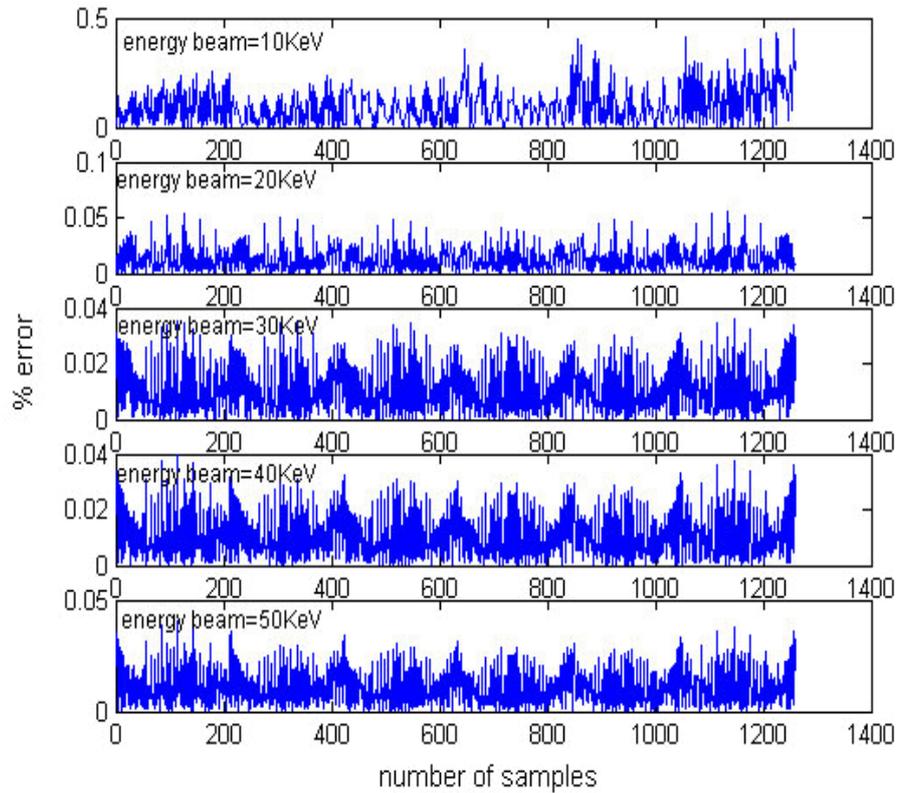


Figure 5.2 - Percentage error between theoretical training CL signal and the training CL signal calculated by ANN

It is also instructive to observe the histogram plot of the error, as in Figure5.3.

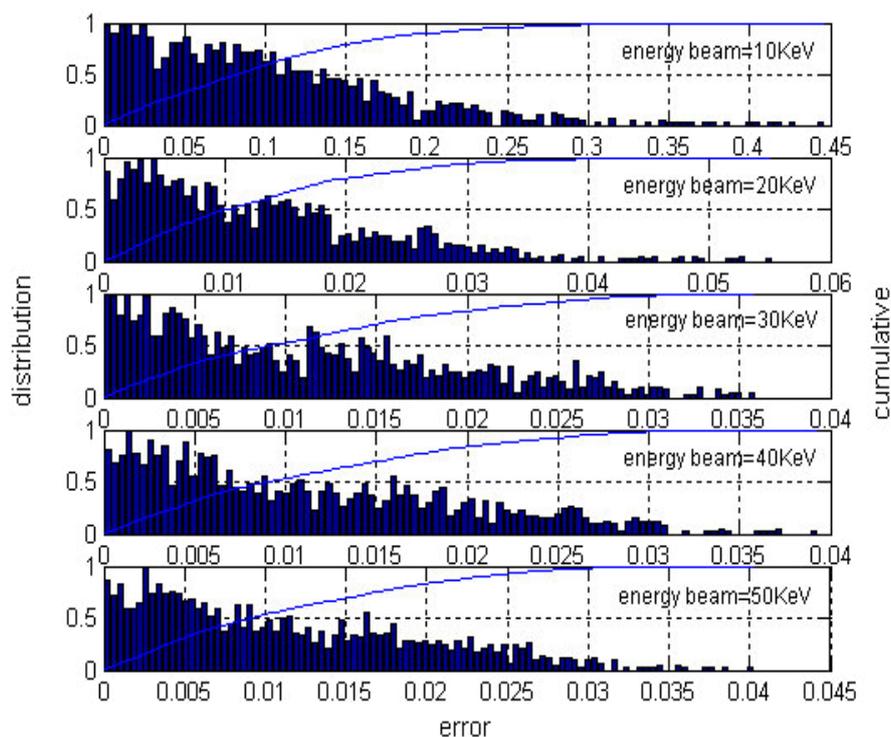


Figure 5.3 Probability density function PDF (bars) and cumulative density function CDF (solid line) of the percentage error for the training set (CL).

We notice from figure 5.3 that most of the errors are clustered around zero, indicating that the ANN is able to learn the input/output relationship effectively. Table 5.4 details the value of the error in 95% and 100% of the samples for different energy beam values.

Table 5.4- Error for 95% and 100% of the training samples (CL).

Energy beam value (KeV)	For 95% of the samples, error is below	For 100% of the samples, error is below
10	0.15%	0.25%
20	0.01%	0.016%
30	0.009%	0.014%
40	0.008%	0.014%
50	0.008%	0.014%

From figure5.3 and table5.4 we notice that in 95% of the training cases the maximal error is small: 0.15% and it is 0.25% in 100% of the training cases.

5.3.2 Testing the algorithm

Our ANN is tested by using values of CL signal not seen before as illustrated in Table5.5 (taking the mid-value between each two successive values of the training set).

Table5.5 -Semiconductor parameters used for testing the ANN (CL)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Energy beam (KeV)	10	50	10	5
Diffusion length $L^{test}(\mu m)$	0.73+(0.01/2)	0.78	0.01	5
Absorption coefficient $\alpha^{test}(\mu m^{-1})$	0.785+(0.01/2)	0.825	0.01	4
Dead layer thickness $Zt^{test}(\mu m)$	0.03+(0.01/2)	0.08	0.01	5
Relative quantum efficiency Q^{test} (arbitrary units)	7.15+(0.05/2)	7.45	0.05	6
Total number of samples for the test equals: number of samples of L^{test} × number of samples of α^{test} × number of samples of Zt^{test} × number of samples of Q^{test} (5×4×5×6=600)				

The total number of samples used for test is 600 samples.

Figure5.4 illustrates the percentage error between the samples of the test CL signal and the CL signal calculated by ANN. The maximal percentage error for all the parameters is: 0.18%.

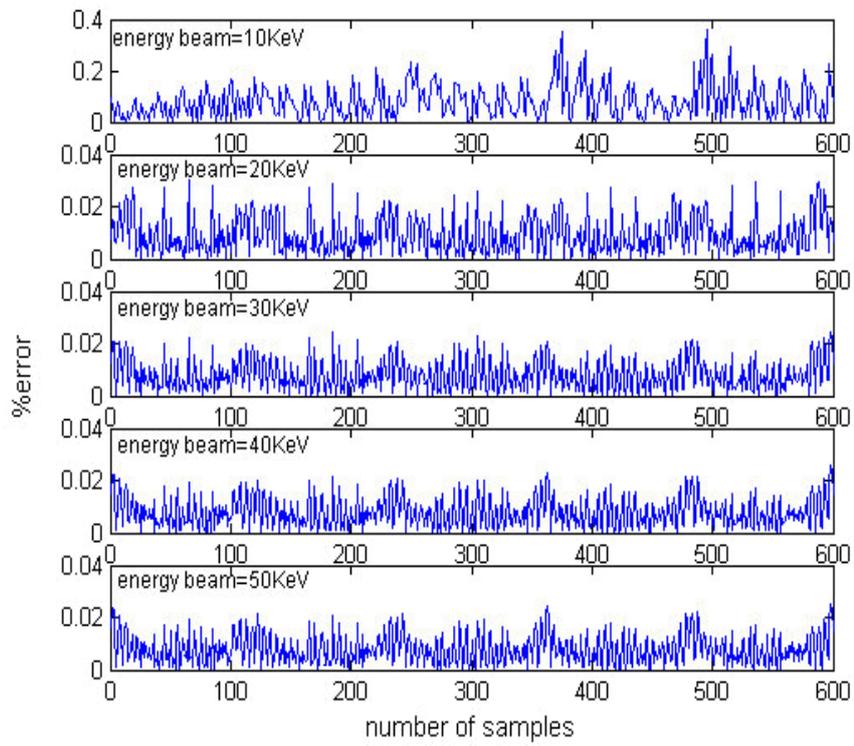


Figure5.4 Percentage error between the theoretical test CL signal and the test CL signal calculated by ANN.

The histogram plot of the error and the CDF is shown in Figure5.5.

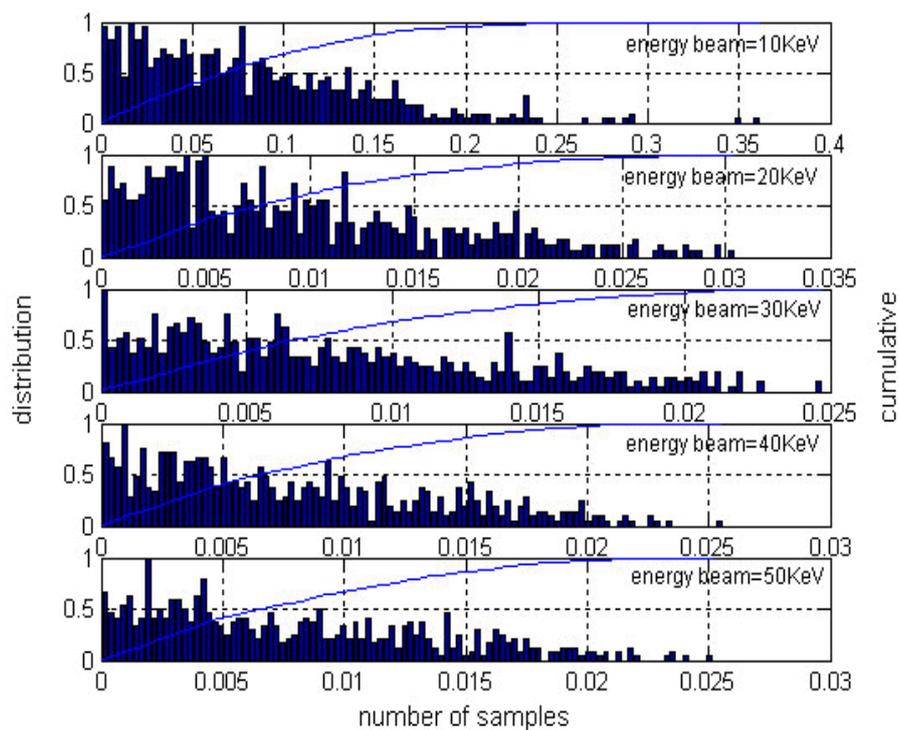


Figure5.5. PDF and CDF of the percentage error for the test set (CL).

All the errors are clustered around zero in figure 5.5.

The values of the error in 95% and 100% of the test samples for different energy beam values can be extracted from figure5.5, the results are reported in table5.6.

Table5.6- Error for 95% and 100% of the test samples (CL).

Energy beam value (KeV)	For 95% of the samples, error is below	For 100% of the samples, error is below
10	0.1%	0.18%
20	0.007%	0.014%
30	0.006%	0.01%
40	0.006%	0.01%
50	0.006%	0.012%

We notice that in 95% of the test cases the maximal error is 0.1% and it is of 0.18% in all the test cases, both values are small.

5.3.3 Oversampling of CL signal

The input parameters ranges are oversampled by taking more samples between each two successive training samples and used to generate the oversampled signal using the ANN (trained and tested before) the values of the oversampled set are detailed in table5.7.

Table5.7- Semiconductor parameters used to obtain the oversampled CL signal

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Energy beam (KeV)	10	50	10	5
Diffusion length L^{over} (μm)	0.73	0.78	0.01/4	21
Absorption coefficient α^{over} (μm^{-1})	0.785	0.825	0.01/4	17
Dead layer thickness Zt^{over} (μm)	0.03	0.08	0.01/4	21
Relative quantum efficiency Q^{over} (arbitrary units)	7.15	7.45	0.05/4	25
Total number of samples for oversampling the CL equals: number of samples of L^{over} \times number of samples of α^{over} \times number of samples of Zt^{over} \times number of samples of Q^{over} ($21 \times 17 \times 21 \times 25 = 187425$).				

The total number of input data samples used for oversampling the CL signal is 187425. Figure5.6 represents the percentage error; the maximal percentage error for all parameters is of 0.25%.

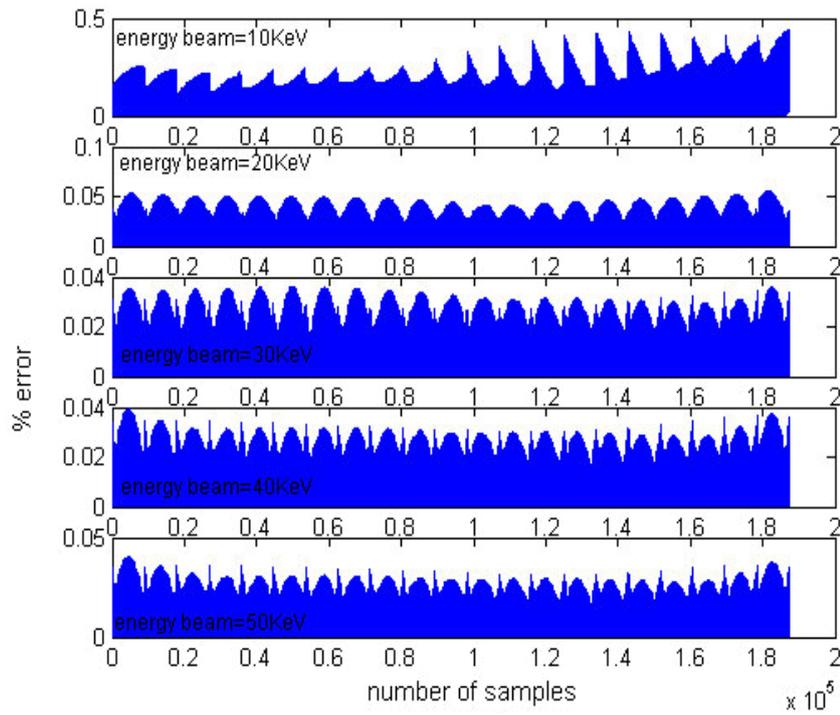


Figure5.6 Percentage error between the theoretical oversampled CL signal and the oversampled CL signal calculated by ANN

The histogram plot of the error is also shown in Figure5.7.

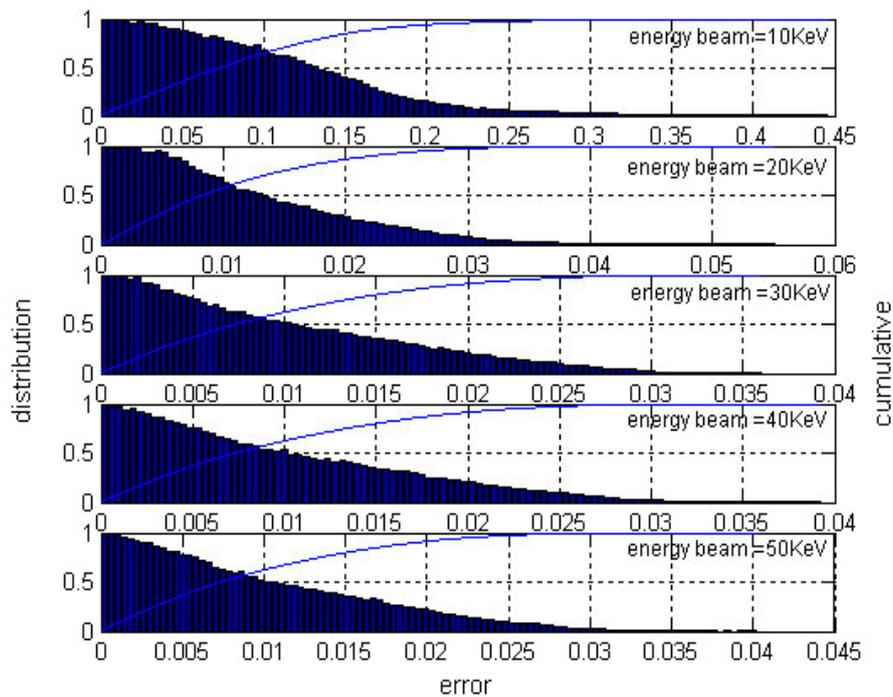


Figure5.7. PDF and CDF of the percentage error for the oversampled set (CL).

Figure 5.7 shows that all the errors are clustered around zero, the smooth Gaussian shape is clear here because of the large number of samples.

From figure 5.7 we detail in table 5.8 the values of the error in 95% and 100% of the samples for different energy beam values.

Table 5.8- Error for 95% and 100% of the oversampled samples (CL).

Energy beam value (KeV)	For 95% of the samples, error is below	For 100% of the samples, error is below
10	0.125%	0.25%
20	0.008%	0.018%
30	0.008%	0.016%
40	0.008%	0.016%
50	0.008%	0.015%

In 95% of the oversampled set the maximal error is of 0.125%, and in all cases it is only 0.25%.

5.3.4 Exhaustive search

After obtaining more outputs using the ANN trained and tested before, an exhaustive search procedure is used to determine the value of the oversampled CL signal that is the closest (in terms of Euclidian distance) to the CL signal for which we want to determine the input parameters (α , L , Z_t , Q).

In order to evaluate and qualify the performance of the parameter extraction algorithm, a large set of randomly selected CL curves for which the input parameters (α , L , Z_t , Q) are known, is generated and used to test the parameter extraction algorithm. The percentage error between the true input parameter (nominal value) and

the parameter determined using the proposed parameter extraction algorithm is calculated from (5-5), and is calculated for each randomly selected CL curve.

Figure5.8 shows the histogram plot and the CDF of the percentage error for each parameter. Note that 153600 randomly selected CL curves are used for testing the parameter extraction algorithm.

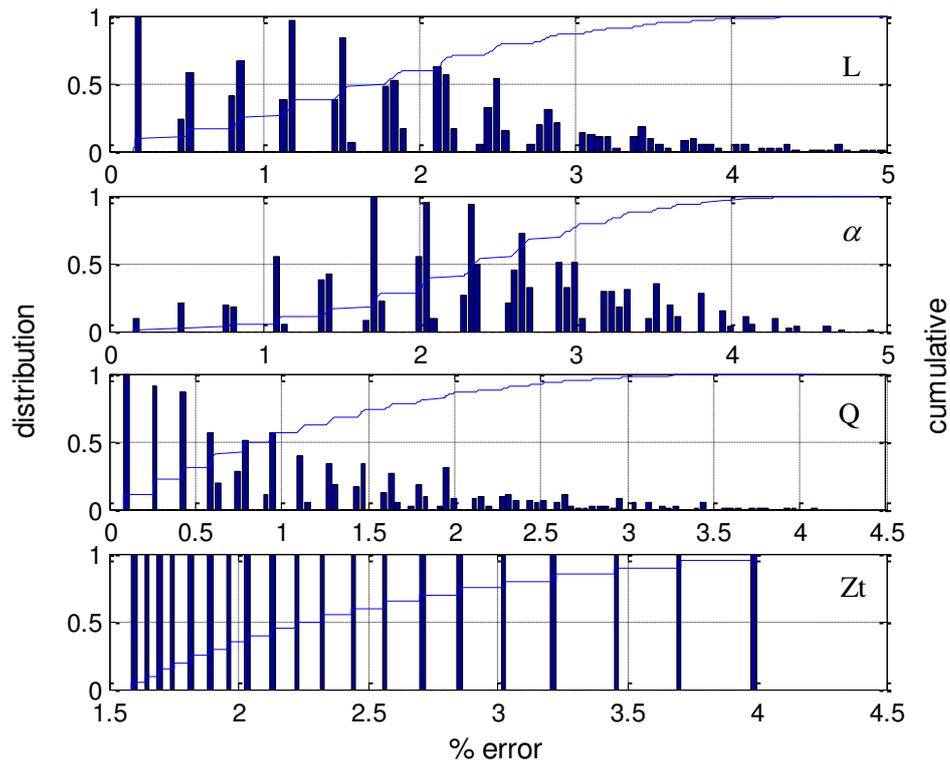


Figure5.8 PDF and CDF of percentage error of the parameters α , L , Z_t , Q

From figure5.8, the value of the error for each parameter for 95% and 100% of the cases is detailed in table5.9.

Table5.9- Error of the four parameters α , L, Zt, Q for 95% and 100% of the cases.

Semiconductor parameter extracted	For 95% of the cases, error is below	For 100% of the cases, error is below
L	3.5%	5.5%
α	4%	5%
Q	3%	4.5%
Zt	4%	4.5%

Table 5.9 shows that the proposed model is very satisfactory since the maximal error in extracting the parameters from 95% of the cases is just 4% and it is 5.5% for all cases.

5.3.5 Effect of measurement noise

In the following, we consider the case of noisy measurements of the noisy CL signal by adding an additive white Gaussian noise (AWGN) signal with different signal to noise ratios (SNRs) to the CL signal used for training the ANN, thus the ANN is trained with a noisy CL signal instead of the exact CL signal generated using the Hergert's model. The effect of noise on the performance of the parameter extraction algorithm is depicted in figures 5.9, 5.10 and 5.11 for the SNRs (dB): 10, 20 and 30, respectively. These figures show that the performance of the parameter extraction algorithm in terms of the percentage error of each parameter, improves as the SNR increases.

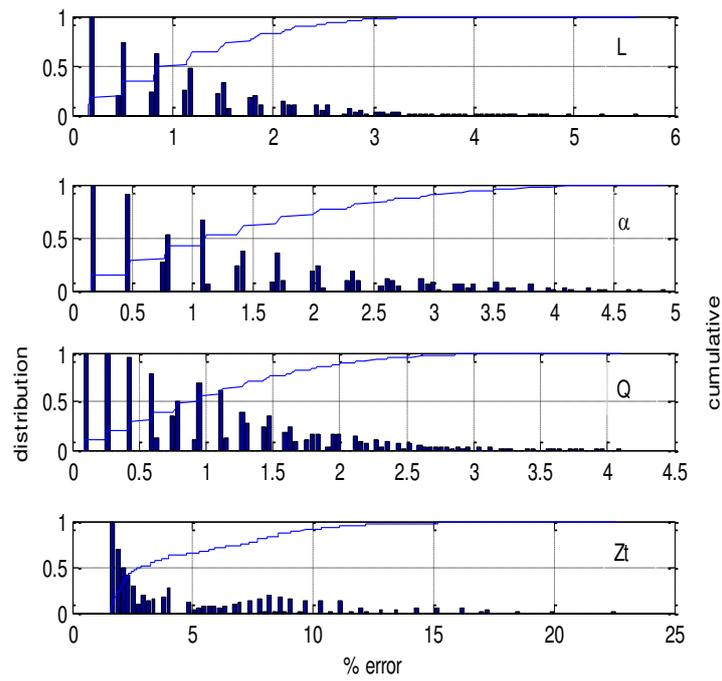


Figure 5.9. PDF and CDF of percentage error of the parameters α , L , Z_t , Q with a SNR of 10 dB.

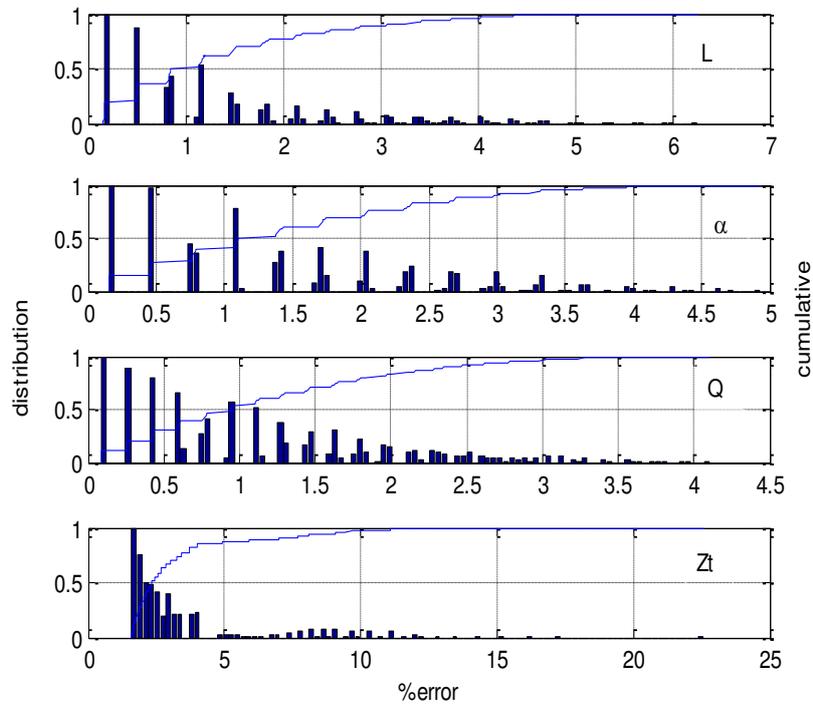


Figure 5.10. PDF and CDF of percentage error of the parameters α , L , Z_t , Q with a SNR of 20 dB.

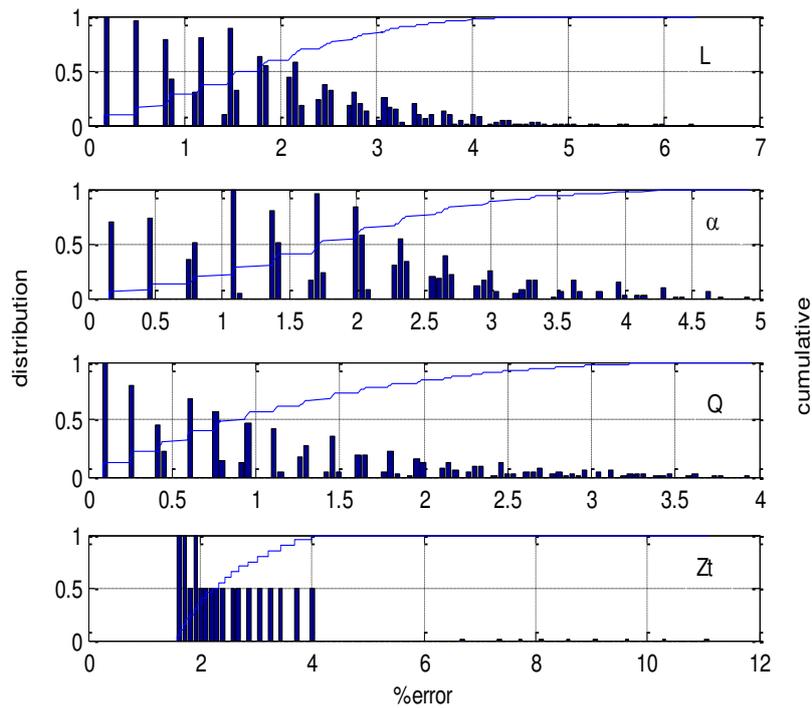


Figure 5.11. PDF and CDF of percentage error of the parameters α , L , Zt , Q with a SNR of 30 dB.

5.4 Application to EBIC

We present here the application of the parameter extraction algorithm described above to the simultaneous extraction of the diffusion length L and the normalized surface recombination velocity S from an EBIC line scan in a normal collector configuration (Donolato's model see chapter 2).

The material sample is chosen to be Silicon (a density of 2.33 g/cm^3); the electron beam is perpendicular to the surface of the semiconductor sample. The line scans the region outside the junction, and the energy beam equals to 14 KeV which results in an electron range of $1.7 \mu\text{m}$.

5.4.1 Training the Algorithm

The first step is to collect the input/output data pairs used to train the ANN. The data were collected from the evaluation of Donolato's model described previously (see chapter 2). However, Monte Carlo simulations and experimental data can be used as well to form the input/output data pairs for the training of ANN.

To train the ANN, the EBIC signal was calculated for each combination of the parameters $\mathbf{L}^{\text{train}}$, $\mathbf{S}^{\text{train}}$. Table 5.10 details the values of the parameters used to generate the EBIC signal and consequently form the input/output data pairs used for training the ANN. For $\mathbf{L}^{\text{train}}$, $\mathbf{S}^{\text{train}}$, the values in [3,4] were used here as well.

Table 5.10-Semiconductor parameters used for training (EBIC)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Beam position(μm)	6	6.9	0.2	5
Diffusion length $\mathbf{L}^{\text{train}}$ (μm)	3	3.9	0.03	31
Normalized surface recombination velocity	2	4	0.03	67
Total number of samples used for training	31×67=2077			

As illustrated in Table 5.10, taking all the possible combinations of $\mathbf{L}^{\text{train}}$, $\mathbf{S}^{\text{train}}$ results in a total number of samples of 2077 samples, therefore, 2077 input/output pairs are used for training the ANN.

A feed forward neural network consisting of one hidden layer is used, the hidden layer comprises 5 neurons and uses logarithmic sigmoid transfer functions whereas the output layer consists of 5 neurons (corresponds to the number of samples of the beam position vector) and uses linear transfer functions. Table 5.11 illustrates the ANN parameters used. The training algorithm used to train the ANN is the Levenberg Marquardt (see chapter 3).

Table 5.11- ANN parameters used for training (EBIC)

ANN parameters		
Number of epochs	300	
Desired goal	10^{-6}	
Number of hidden layers	1	
Number of neurons in the hidden layer	5	uses logarithmic sigmoid transfer functions
Number of neurons in the output layer	5 (number of samples of the beam position)	uses linear transfer functions

The training curve of the ANN is shown in figure5.12; the desired goal was achieved after 40 epochs.

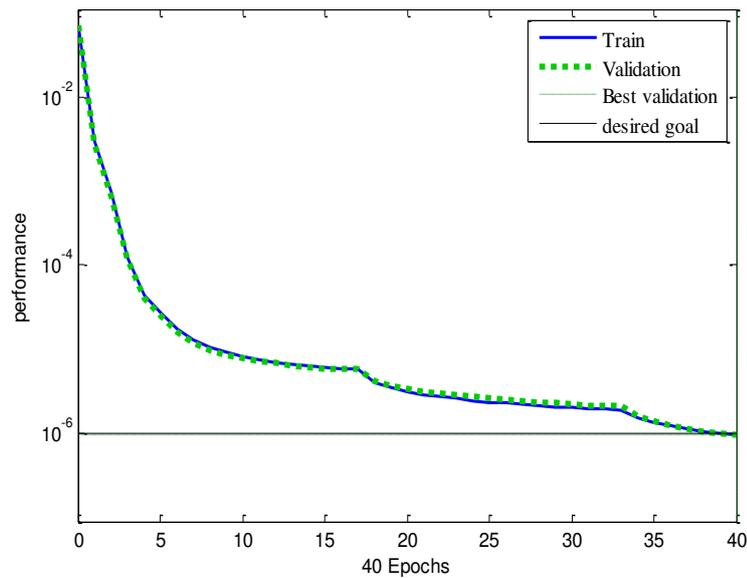


Figure5.12 Training curve of the ANN (EBIC).

Figure5.13 shows the percentage error between the theoretical training EBIC and the EBIC calculated by the ANN for five values of the beam position. The maximal percentage error for the two parameters is 0.9037%. This indicates that the training is very satisfactory.

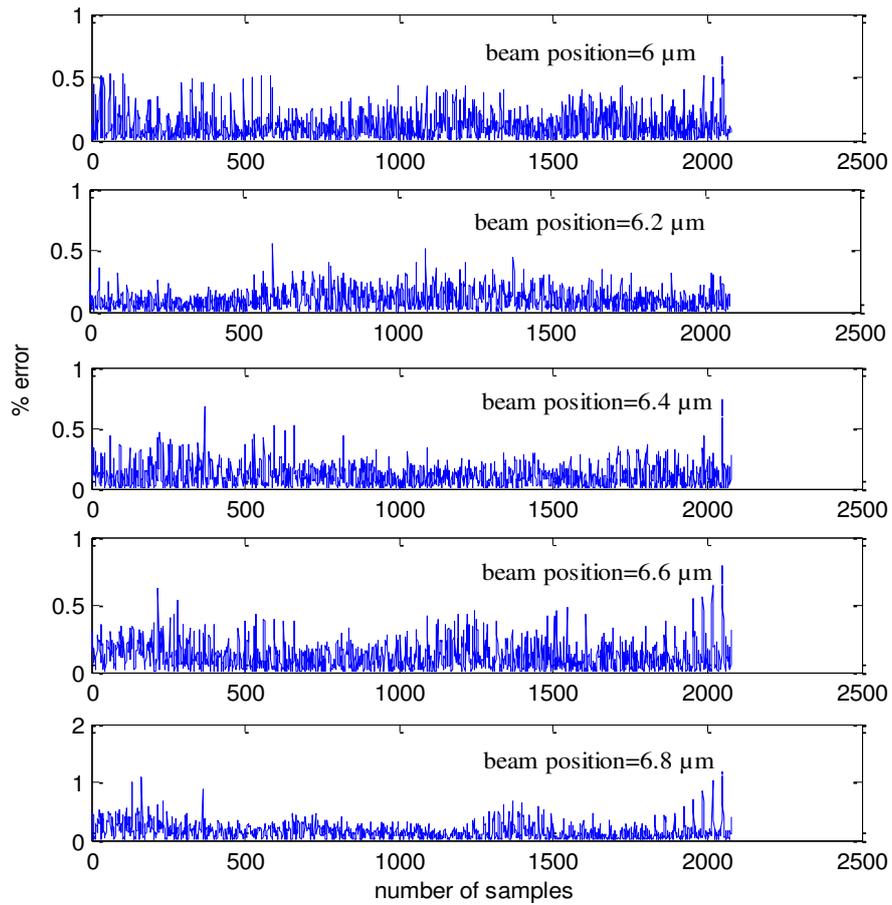


Figure 5.13 - Percentage error between theoretical training EBIC and the training EBIC calculated by ANN

The histogram plot of the error is depicted in Figure 5.14.

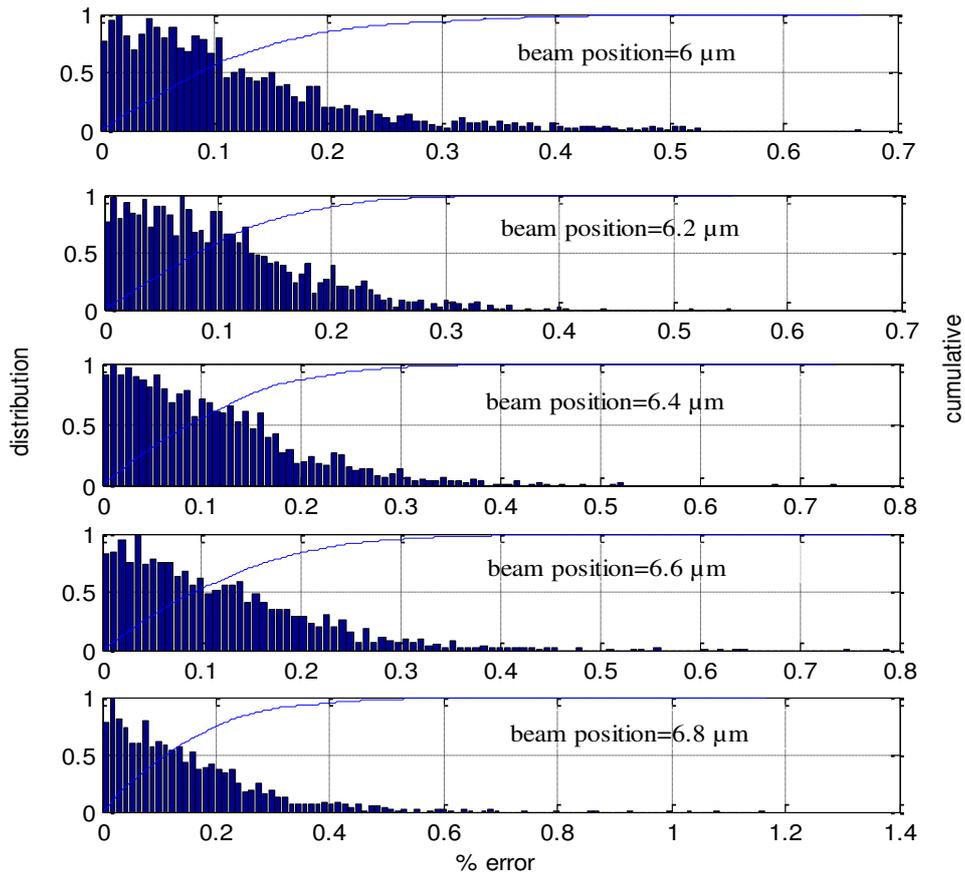


Figure 5.14 Probability density function PDF (bars) and cumulative density function CDF (solid line) of the percentage error for the training set (EBIC).

In figure 5.14 most of the errors are clustered around zero, this indicates that the learning is satisfactory. Table 5.12 details the value of the error in 95% and 100% of the samples for different beam position values.

Table5.12- Error for 95% and 100% of the training samples (EBIC).

beam position value(μm)	For 95% of the samples, error is below	For 100% of the samples, error is below
6	0.3%	0.65%
6.2	0.25%	0.55%
6.4	0.3%	0.75%
6.6	0.3%	0.8%
6.8	0.4%	1.2%

The maximal error in 955 of the training set is only 0.4% and for all the training set it is only 1.2%.

5.4.2 Testing the algorithm

Our ANN is tested using values of EBIC not seen before (mid-value between each two successive values of the training set) as illustrated in Table5.13.

Table5.13-Semiconductor parameters used for testing the ANN (EBIC)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Beam position(μm)	6	6.9	0.2	5
Diffusion length L^{train} (μm)	$3+(0.03/2)$	3.9	0.03	30
Normalized surface recombination velocity S^{train}	$2+(0.03/2)$	4	0.03	67
Total number of samples used for training	$30 \times 67 = 2010$			

The total number of samples used for test is 2010 samples. Figure5.15 illustrates the percentage error between the samples of the test EBIC and the EBIC calculated by ANN, the maximal percentage error for the two parameters is 1.5571%.

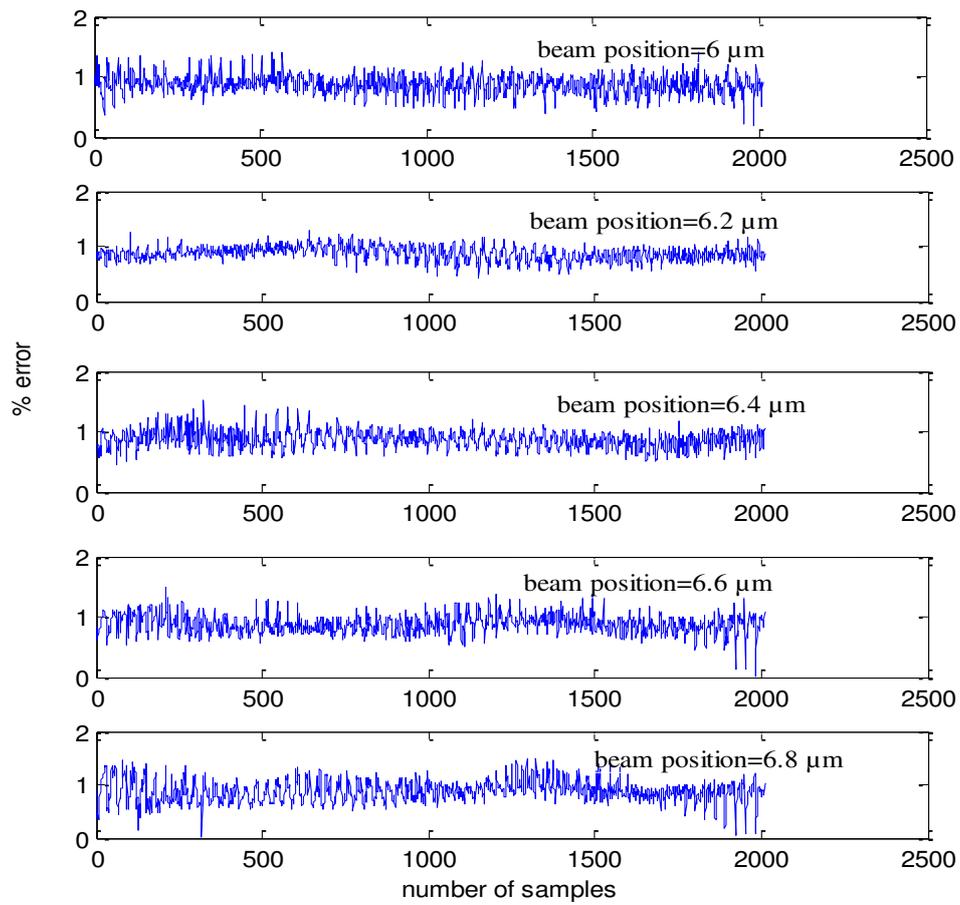


Figure5.15 Percentage error between the theoretical test EBIC and the test EBIC calculated by ANN.

The histogram plot of the error and the CDF is shown in Figure5.16.

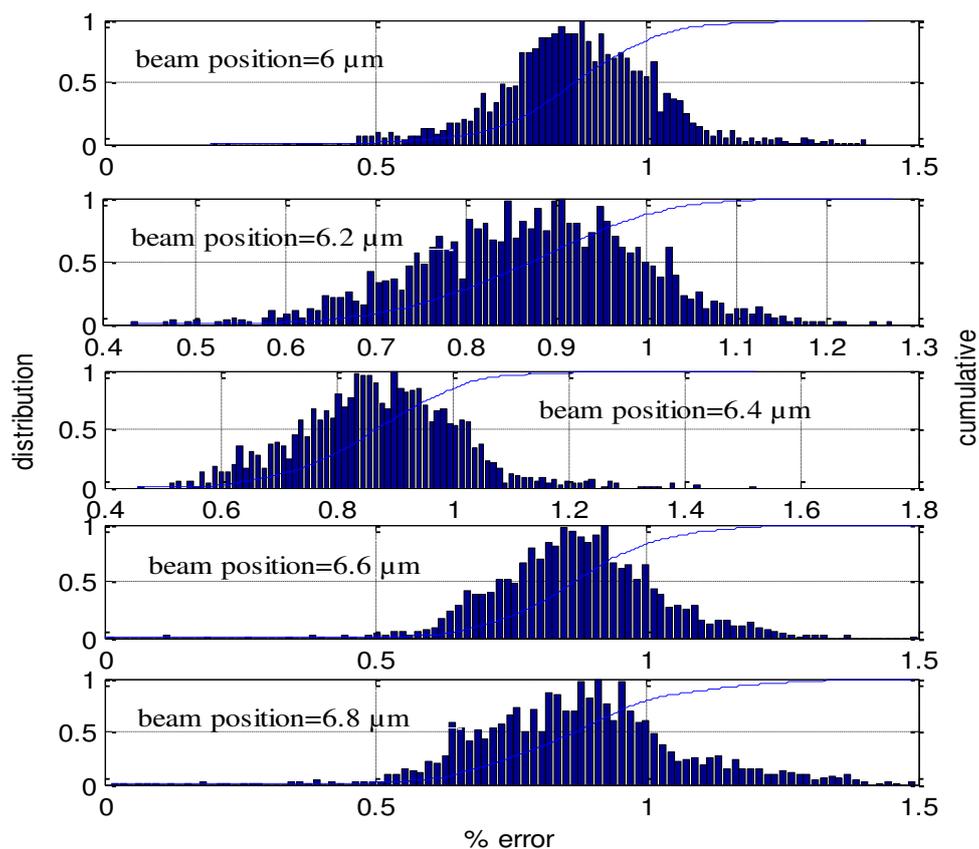


Figure 5.16. PDF and CDF of the percentage error for the test set (EBIC).

From figure 5.16, we detail in table 5.14 the values of the error in 95% and 100% of the samples for different beam position values.

Table 5.14- Error for 95% and 100% of the test samples (EBIC).

Beam position value (μm)	For 95% of the samples, error is below	For 100% of the samples, error is below
6	1.08%	1.4%
6.2	1.04%	1.3%
6.4	1.1%	1.5%
6.6	1.1%	1.5%
6.8	1.2%	1.5%

The maximal error in 95% of the test set is 1.2% and is only 1.5% in all the test set.

5.4.3 Oversampling of EBIC

The input parameters ranges are oversampled by taking more samples between each two successive training samples and used to generate the oversampled signal using the ANN (trained and tested before) are detailed in table5.15.

Table 5.15- Semiconductor parameters used to obtain the oversampled EBIC

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Beam position(μm)	6	6.9	0.2	5
Diffusion length L^{train} (μm)	3	3.9	0.03/4	121
Normalized surface recombination velocity	2	2.66	0.03/4	89
Total number of samples used for training	121 \times 89=10769			

The total number of input data samples used for oversampling the EBIC is 10769.

Figure5.17 represents the percentage error; the maximal percentage error for the two parameters is 0.9654%.

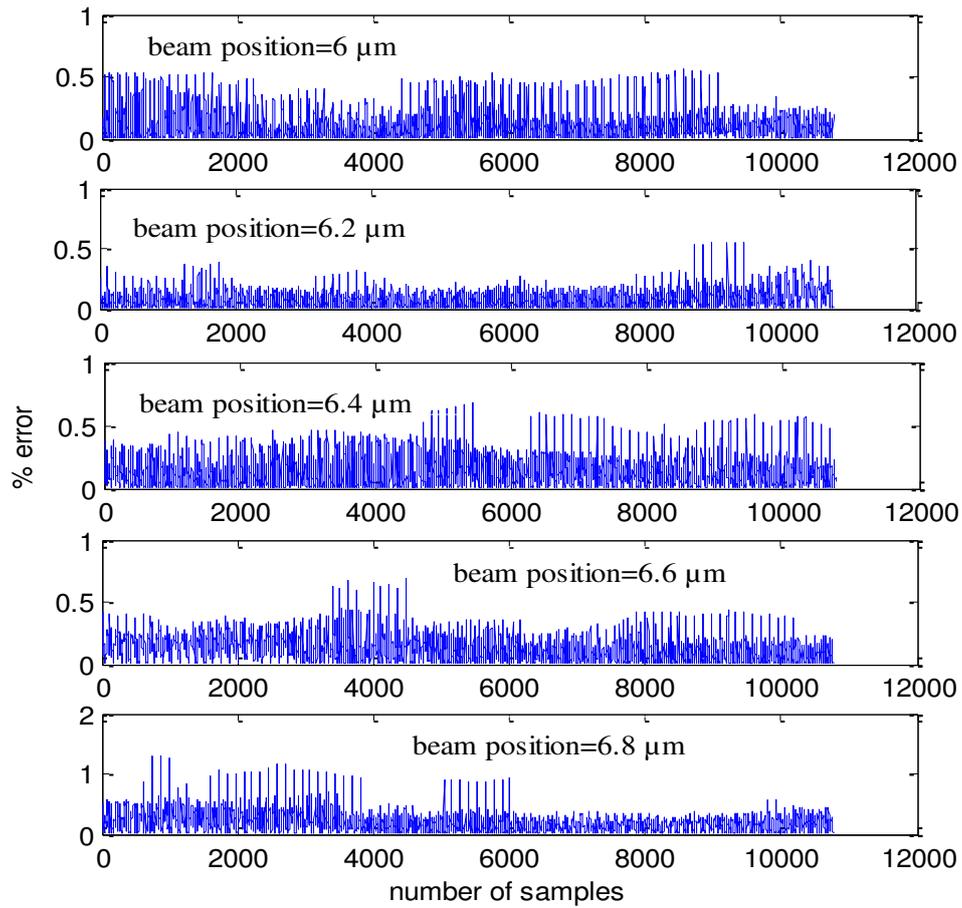


Figure5.17 Percentage error between the theoretical oversampled EBIC and the oversampled EBIC calculated by ANN

The histogram plot of the error is also shown in Figure5.18.

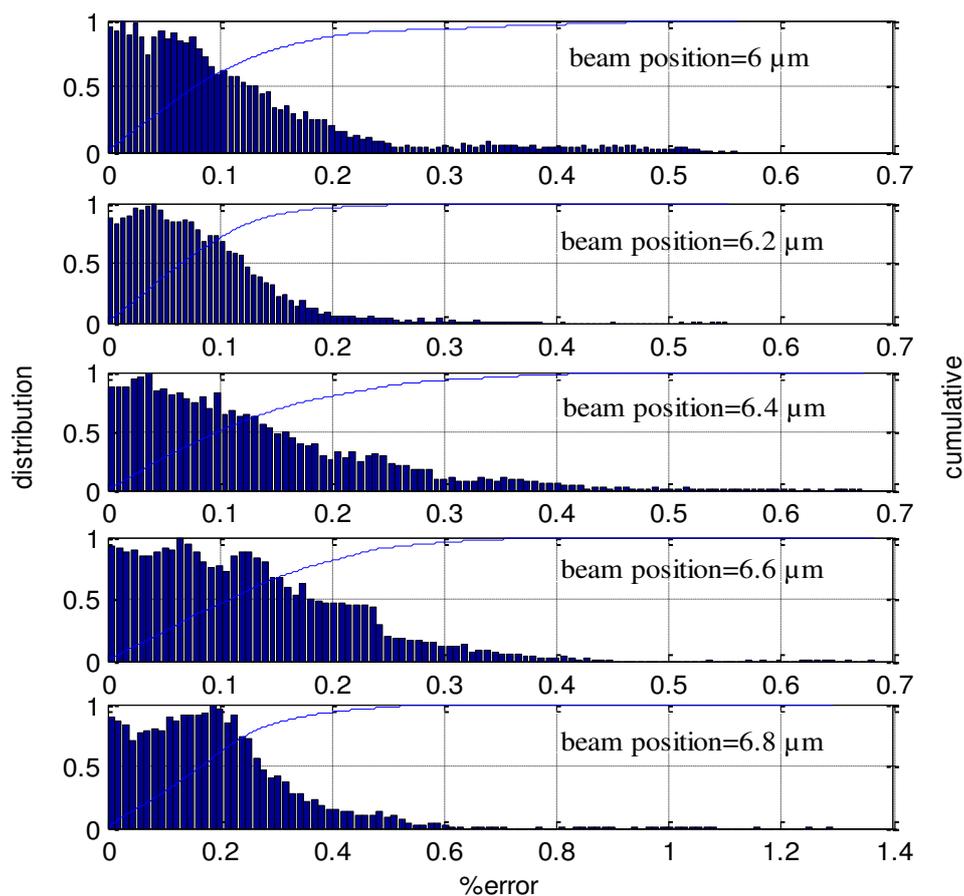


Figure5.18. PDF and CDF of the percentage error for the oversampled set (EBIC).

Using figure5.18 we detail in table 5.16 the values of the error in 95% and 100% of the samples for different beam position values.

Table5.16- Error for 95% and 100% of the oversampled samples (EBIC).

beam position value (μm)	For 95% of the samples, error is below	For 100% of the samples, error is below
6	0.35%	0.55%
6.2	0.2%	0.55%
6.4	0.35%	0.7%
6.6	0.3%	0.7%
6.8	0.45%	1.26%

The maximal error in 95% of the set is 0.45% and it is 1.26% in 100% of the set.

5.4.4 Exhaustive search

After obtaining more outputs using the ANN trained and tested before, an exhaustive search procedure is used to determine the value of the oversampled EBIC that is the closest (in terms of Euclidian distance) to the EBIC for which we want to determine the input parameters.

Figure5.19 shows the histogram plot and the CDF of the percentage error calculated for each parameter using (5-5), and is calculated for each randomly selected EBIC. Note that 10680 randomly selected EBIC currents are used for testing the parameter extraction algorithm.

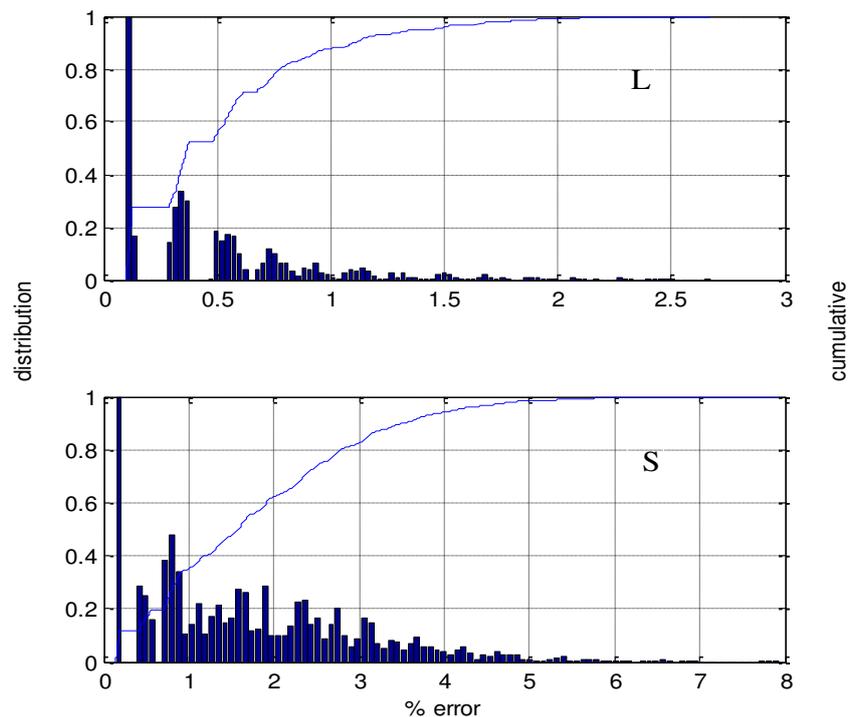


Figure5.19 PDF and CDF of percentage error of the parameters L, S

From figure 5.19, the value of the error for each parameter for 95% and 100% of the cases is detailed in table 5.17.

Table 5.17- Error of the two parameters L, S for 95% and 100% of the cases (EBIC).

Semiconductor parameter extracted	For 95% of the cases, error is below	For 100% of the cases, error is below
L(μm)	1.5%	2.7%
S	4%	8%

The maximal error in extracting simultaneously the two parameters L and S in 95% of the cases is 4% and is 8% for all the cases; this is a good result compared to other results found in the literature [3-4].

5.5 Conclusion

In this chapter a new model for the simultaneous extraction of related semiconductor parameters is presented it is based on artificial neural networks (ANN) and an exhaustive search technique.

The results derived from the application of our parameter extraction algorithm to the extraction of semiconductor parameters using CL/EBIC signals can be summarized into the following points:

- For the simultaneous extraction of the four related parameters (L, α , Zt, Q) from CL signal a unique set of parameter values is obtained with errors less than: 3.5% for L, 4% for α , 3% for Q, and 4% for Zt, in 95% of the cases used, and an error less than 5.5% for L, 5% for α , 4.5% for Q, and 4.5% for Zt, in 100% of the cases. This clearly indicates that the proposed strategy is very successful.

- The training, test and oversampling errors are significantly larger for low energy beam values. This is remarkable at 10KeV. This is due the fact that the variation of CL signal for low beam energies is very small when these parameters are varied (see figure 2.3.a and 2.3.b in chapter2). The sensitivity of the CL signal to the variation of parameters at low beam energies is small. This makes the modeling of the CL generation process using the ANN difficult for low beam energies and relatively the error is high for low beam energies.
- The performance of algorithm in terms of the percentage error of each parameter, improves as the SNR increases.
- For the simultaneous extraction of the two parameters (L, S) from EBIC a set of parameter values is obtained with errors less than 1.7% for the diffusion length and less than 4% for the normalized surface recombination velocity in 95% of the cases and an error less than 2.7% for the diffusion length and less than 8% for the normalized surface recombination velocity in 100% of the cases. These results confirm that our strategy is very successful.

Chapter references

1. L. Cazzanty, M. Khan, F. Cerrina, "Parameter extraction with neural networks", Proceeding SPIE, 3332, 654. 1998
2. Puhlmann, N. Oelgart, G. "Semiconductor Characterization by means of EBIC, Cathodoluminescence, and photoluminescence". Physica Status Solidi. (a) 122,705. 199031.

3. Oka. Kurniawan, "device parameters characterization with the use of EBIC", a thesis submitted for the degree of Doctor of Philosophy, Nanyang Technological University, 2008, supervised by Professor V. K. S. Ong.
4. Kurniawan. O, Ong. K. S, "Choice of generation volume models for electron beam induced current computation", IEEE Transactions on Electron Devices 56(5), 1094-1099, 2009.

Chapter 6

Semiconductor parameter extraction using Artificial Neural Networks and inverse modeling

Contents

6.1 Introduction	
6.2 Parametre extraction using ANN and inverse modeling.....	
6.3 Application to cathodoluminescence	
6.4 Application to EBIC	
6.5 Conclusion.....	
Chapter references.....	

6.1 Introduction

In this chapter a new parameter extraction algorithm based on ANN is presented and discussed, the ANNs are used here to model the inverse process of CL/EBIC generation process (see chapter3 paragraph 3-7).

The parameter extraction algorithm is applied to the simultaneous extraction of the diffusion length L , absorption coefficient α , dead layer thickness Z_t , and relative quantum efficiency Q , from any steady state CL signal of a free defect semi-infinite semiconductor. Similarly, the parameter extraction algorithm is also applied to the simultaneous extraction of two related semiconductor parameters, that is, the diffusion length L and the normalized surface recombination velocity S from any EBIC line scan in a normal collector configuration.

6.2 Parameter Extraction based on ANN and inverse modeling

Assume we have a nonlinear functional relationship given by: $Y = F(X_1, X_2, \dots, X_n)$; where X_i $i=1, \dots, n$ represents the input vector and Y is the output vector (signal). The parameter extraction algorithm based on a feedforward ANN, and inverse modeling technique is summarized into the following steps:

6.2.1 Preparation of the training and test data sets:

To train the ANN, the training data set is prepared by sampling and stacking the parameters (X_1, X_2, \dots, X_n) into vectors as follows: $\mathbf{X}_1^{\text{train}} = [X_{1,1}^{\text{train}}, X_{1,2}^{\text{train}}, \dots, X_{1,d_1}^{\text{train}}]$, $\mathbf{X}_2^{\text{train}} = [X_{2,1}^{\text{train}}, X_{2,2}^{\text{train}}, \dots, X_{2,d_2}^{\text{train}}]$... $\mathbf{X}_n^{\text{train}} = [X_{n,1}^{\text{train}}, X_{n,2}^{\text{train}}, \dots, X_{n,d_m}^{\text{train}}]$, where there dimensions are given by $\{1\text{-by-}d_1\}$, $\{1\text{-by-}d_2\}$... $\{1\text{-by-}d_m\}$ respectively. All the possible combinations of the vectors $\mathbf{X}_1^{\text{train}}, \mathbf{X}_2^{\text{train}} \dots \mathbf{X}_n^{\text{train}}$ form the input training set $\mathbf{X}^{\text{train}}$ of dimension $\{n\text{-by-}d_1 \times d_2 \times \dots \times d_m\}$. For each column of $\mathbf{X}^{\text{train}}$ the Y signal is evaluated using. The results are stored in another matrix $\mathbf{Y}^{\text{train}}$ (output training set).

For the test data set the two parameters (X_1, X_2, \dots, X_n) are sampled and stacked into vectors by taking the mid-value between each two successive values of the training set:

$$\mathbf{X}_1^{\text{test}} = [X_{1,1}^{\text{test}}, X_{1,2}^{\text{test}}, \dots, X_{1,d_1-1}^{\text{test}}], \quad \mathbf{X}_2^{\text{test}} = [X_{2,1}^{\text{test}}, X_{2,2}^{\text{test}}, \dots, X_{2,d_2-1}^{\text{test}}] \dots$$

$\mathbf{X}_n^{\text{test}} = [X_{n,1}^{\text{test}}, X_{n,2}^{\text{test}}, \dots, X_{n,d_m-1}^{\text{test}}]$. The dimensions of the vectors $\mathbf{X}_1^{\text{test}}, \mathbf{X}_2^{\text{test}} \dots \mathbf{X}_n^{\text{test}}$ are given by $\{1\text{-by-}(d_1-1)\}$, $\{1\text{-by-}(d_2-1)\}$... $\{1\text{-by-}(d_m-1)\}$, respectively. All their possible combinations form the input test data set \mathbf{X}^{test} of dimension $\{n\text{-by-}(d_1-1) \times (d_2-1) \dots \times (d_m-1)\}$. For each column of the matrix \mathbf{X}^{test} , the Y signal is calculated and the results are stored in a matrix \mathbf{Y}^{test} (output test data set).

6.2.2 Training the ANN algorithm:

The ANN is trained to model the inverse of the functional relationship f between the input process data set $\mathbf{X}^{\text{train}}$ and the output data set $\mathbf{Y}^{\text{train}}$ (see §3-7 in chapter 3), that is, the output process data set $\mathbf{Y}^{\text{train}}$ is used as an input to the ANN and the input process data set $\mathbf{X}^{\text{train}}$ is used as an output of the ANN, that is:

$$\mathbf{Y}^{\text{train}} \xrightarrow{f^{-1}} \mathbf{X}^{\text{train}} \quad (6-1)$$

Here the ANN receives the input data set $\mathbf{Y}^{\text{train}}$ and calculates the actual output data set $\mathbf{X}^{\text{actual}}$ which is then subtracted from the desired output data set $\mathbf{X}^{\text{train}}$ resulting in an error \mathbf{e} . The latter is then used to update the ANN weights. This process is repeated a number of epochs till the sum squared error (SSE) goes beyond a predefined threshold. The SSE is given by the following formula [1] :

$$\text{SSE} = \sum_{j=1}^{k^{\text{train}}} (Y_j^{\text{train}} - Y_j^{\text{actual}})^2 \quad (6-2)$$

where k^{train} represents the number of samples of $\mathbf{Y}^{\text{train}}$.

6.2.3 Testing the ANN algorithm

Once the ANN has been trained, its ability to generalize is tested by applying a new set of input values \mathbf{Y}^{test} that have not seen before. The output vector $\tilde{\mathbf{X}}^{\text{test}}$ calculated by the ANN is then compared to the desired data set \mathbf{X}^{test} and the sample-by-sample percentage error between \mathbf{X}^{test} and $\tilde{\mathbf{X}}^{\text{test}}$ is calculated using the following formula [1]:

$$e_{i,j} = \left| \frac{\mathbf{Y}_{i,j}^{\text{test}} - \tilde{\mathbf{Y}}_{i,j}^{\text{test}}}{\mathbf{Y}_{i,j}^{\text{test}}} \right| \times 100 \quad (6-3)$$

where i and j indicate the i^{th} row, j^{th} column of the matrices \mathbf{Y}^{test} and $\tilde{\mathbf{Y}}^{\text{test}}$, respectively.

6.3 Application to cathodoluminescence

We present here the application of the parameter extraction algorithm described above to the simultaneous extraction of diffusion length L , normalized surface recombination velocity S , absorption coefficient α , dead layer thickness Z_t , and relative quantum efficiency Q from a steady state CL signal of a free defect semi-infinite semiconductor (Hergert's et al model; see chapter2).

We consider an n-type GaAs semiconductor its critical angle is equal to: 16° and its atomic number is equal to: 32, and the electron beam is perpendicular to the surface of the semiconductor sample.

6.3.1 Training the Algorithm

First, the input/output data pairs used to train the ANN are collected. The data was collected from the evaluation of Hergert's model. However, Monte Carlo simulations and experimental data can be used as well to form the input/output data pairs for the training of ANN. The CL signal was calculated for each combination of the parameters α^{train} , L^{train} , S^{train} , Z_t^{train} and Q^{train} for different values of the energy beam. Table6.1 details the values of the semiconductor parameters used to generate the CL signal and consequently form the input/output data pairs used for training the ANN. For α^{train} , L^{train} , S^{train} , Z_t^{train} and Q^{train} , the values in [2] were used here as well.

Table6.1 Semiconductor parameters used for training (CL)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Energy beam (KeV)	10	50	10	5
Diffusion length $L^{\text{train}}(\mu\text{m})$	0.73	0.77	0.01	5
Absorption coefficient $\alpha^{\text{train}}(\mu\text{m}^{-1})$	0.78	0.82	0.01	5
Normalized surface recombination velocity S^{train}	5	10	1	6
Dead layer thickness $Zt^{\text{train}}(\mu\text{m})$	0.03	0.08	0.01	6
Relative quantum efficiency Q^{train}	7.15	7.45	0.05	7
Total number of samples used for training	$5 \times 5 \times 6 \times 6 \times 7 = 6300$			

As illustrated in Table6.1, taking all the possible combinations of α^{train} , L^{train} , S^{train} , Zt^{train} and Q^{train} , results in a total number of 6300 input/output pairs used to train the ANN.

A feedforward neural network consisting of one hidden layer is used, the hidden layer comprises 30 neurons and uses logarithmic sigmoid transfer functions whereas the output layer consists of 5 neurons (corresponds to the number of the rows in $\mathbf{X}^{\text{train}}$) and uses linear transfer functions. Table6.2 illustrates the ANN parameters used in this simulation. The learning algorithm is a Levenberg Marquardt (see chapter 3).

Table6.2 ANN parameters used for training (CL)

ANN parameters		
Number of epochs	3000	
Desired goal	10^{-4}	
Number of hidden layers	1	
Number of neurons in the hidden layer	30	uses logarithmic sigmoid transfer functions
Number of neurons in the output layer	5 (number of rows of $\mathbf{X}^{\text{train}}$)	uses linear transfer functions

The training curve of the ANN is shown in figure6.1; the desired goal is achieved after 1984 epochs.

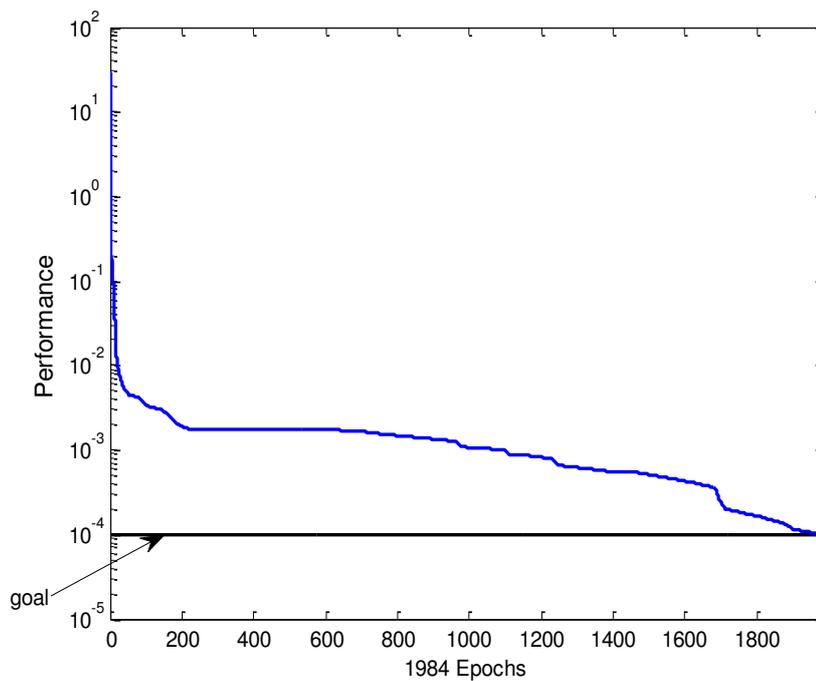


Figure6.1 Training curve of the ANN (CL)

Figure6.2 shows the percentage error between the theoretical parameters L , α , S , Q , and Z_t and parameters calculated by the ANN.

The maximal percentage error for each parameter L , α , S , Q , and Z_t is: 1.5011%, 1.2883%, 0.9727%, 1.0843%, and 1.8617% respectively.

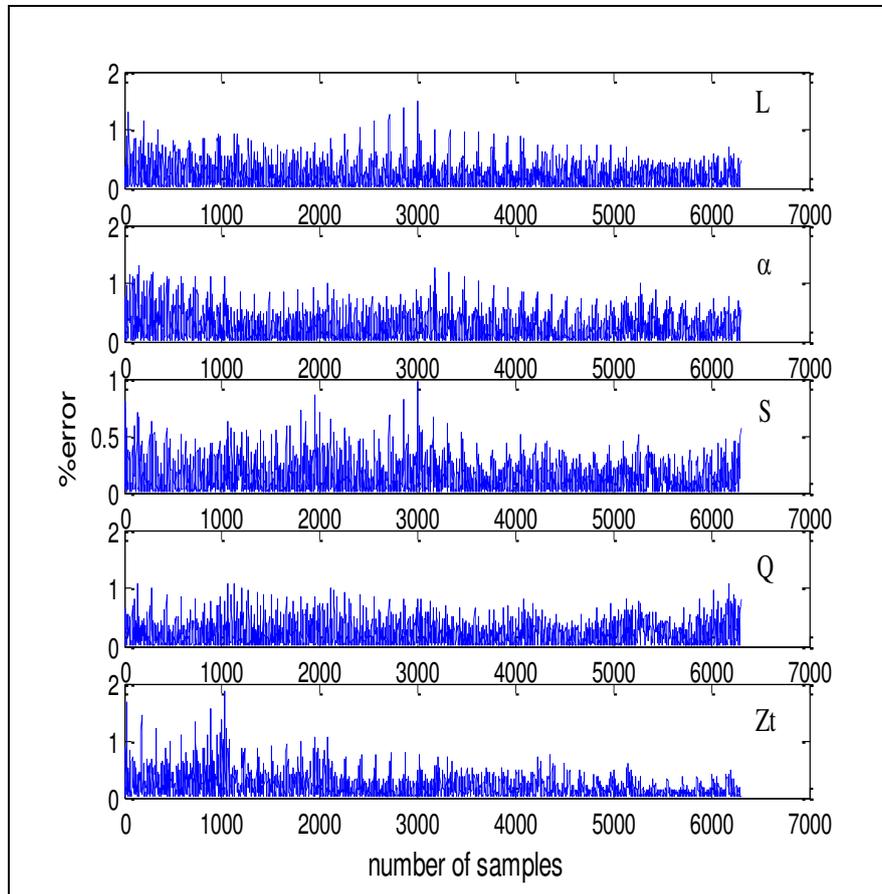


Figure6.2 Percentage error between theoretical parameters and the parameters calculated by ANN for the training set (CL).

It is also instructive to observe the histogram plot of the error, as in Figure6.3.

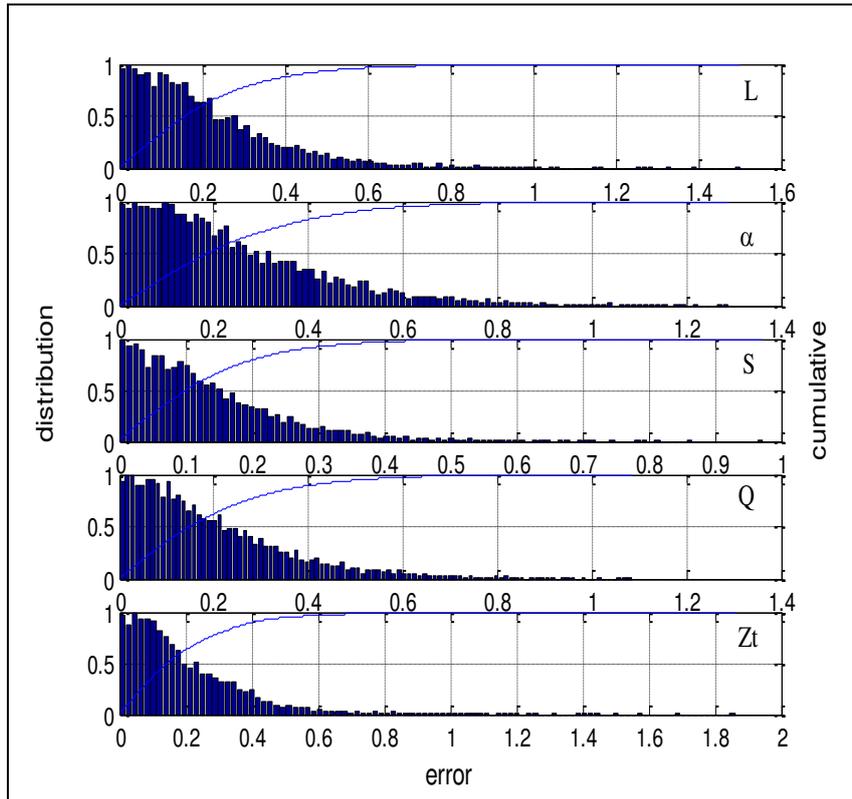


Figure6.3 Probability density function PDF (bars) and cumulative density function CDF (solid line) of the percentage error of L , α , S , Q , and Z_t for the training set.

We notice from figure6.3 that most of the errors are clustered around zero, indicating that the ANN is able to learn the input/output relationship effectively. Table6.3 details the value of the percentage error in 95% and 100% of the training samples for each parameter.

Table6.3 Percentage error of the five parameters L , α , S , Q , and Z_t for 95% and 100% of the training samples (CL).

Semiconductor parameter	For 95% of the training samples, the percentage error is below	For 100% of the training samples, the percentage error is below
L (μm)	0.6%	1.6%
α (μm^{-1})	0.6%	1.4%
S	0.4%	1%
Q (arbitrary units)	0.6%	1.4%
Z_t (μm)	0.6%	2%

We note from table6.3 that the maximal error in 95% of the training cases is of 0.6% and in all the training cases the maximal error is of 2%.

6.3.2 Testing the algorithm

A data not seen before used for testing the ANN is illustrated in Table6.4 (taking the mid-value between each two successive values of the training set).

Table6.4 Semiconductor parameters used for testing the ANN (CL)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Energy beam (KeV)	10	50	10	5
Diffusion length L^{test} (μm)	$0.73+(0.01/2)$	0.77	0.01	4
Absorption coefficient α^{test} (μm^{-1})	$0.78+(0.01/2)$	0.82	0.01	4
Normalized surface recombination velocity S^{train}	$5+(1/2)$	10	1	5
Dead layer thickness Z_t^{test} (μm)	$0.03+(0.01/2)$	0.08	0.01	5
Relative quantum efficiency Q^{test}	$7.15+(0.05/2)$	7.45	0.05	6
Total number of samples for the test	$4 \times 4 \times 5 \times 5 \times 6 = 2400$			

The total number of samples used for test is 2400 input/output pairs.

Figure6.4 illustrates the percentage error between the samples of the test CL signal and the CL signal calculated by ANN. The maximal percentage error for each parameter L, α , S, Q, and Zt is: 0.815%, 0.7847%, 0.5390%, 0.8544%, and 0.7472%, respectively.

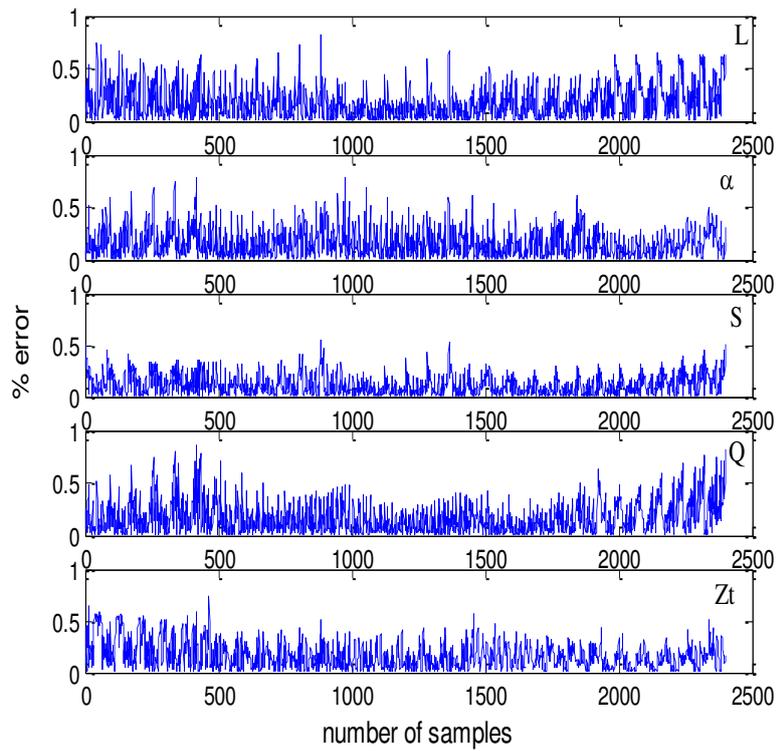


Figure6.4 Percentage error between the theoretical parameters and the parameters calculated by ANN for the test set (CL).

The PDF and the CDF of the percentage error is shown in Figure6.5.

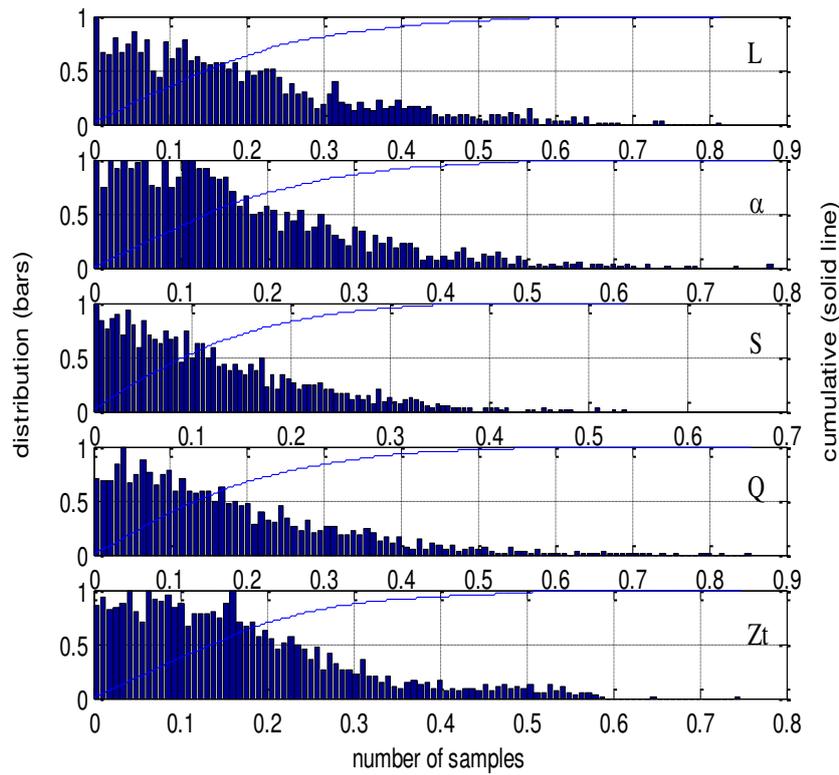


Figure 6.5 PDF and CDF of the percentage error of L , α , S , Q , and Z_t for the test set.

Using figure 6.5, we detail in table 6.5 the values of the error in 95% and 100% of the test samples for each parameter.

Table 6.5 Percentage error of the five parameters L , α , S , Q , Z_t for 95% and 100% of the test samples

Semiconductor parameter	For 95% of the test samples, percentage error is below	For 100% of the test samples, percentage error is below
L (μm)	0.5%	0.9%
α (μm^{-1})	0.4%	0.8%
S	0.3%	0.7%
Q (arbitrary units)	0.4%	0.9%
Z_t (μm)	0.4%	0.8%

The results obtained and presented in table6.5 give a maximal error of 0.5% in 955 of the test cases and a maximal error of 0.9% of all the test cases.

6.4 Application to EBIC

We present now the application of the parameter extraction algorithm described above to the simultaneous extraction of the diffusion length L and the normalized surface recombination velocity S from an EBIC line scan in a normal collector configuration (Donolato's model see chapter2).

The material sample is chosen to be Silicon (a density of 2.33 g/cm^3); the electron beam is perpendicular to the surface of the semiconductor sample. The line scans the region outside the junction, and the energy beam is equals to 14KeV which gives for the energy range a value of $1.7\mu\text{m}$.

6.4.1 Training the Algorithm

The first step is to collect the input/output data pairs used to train the ANN. The data were collected from the evaluation of Donolato's model described in chapter 2. However, Monte Carlo simulations and experimental data can be used as well to form the input/output data pairs for the training of ANN.

To train the ANN, the EBIC signal was calculated for each combination of the parameters $\mathbf{L}^{\text{train}}$, $\mathbf{S}^{\text{train}}$. Table6.6 details the values of the parameters used to generate the EBIC signal and consequently form the input/output data pairs used for training the ANN. For $\mathbf{L}^{\text{train}}$, $\mathbf{S}^{\text{train}}$, the values in [3,4] were used here as well.

Table6.6-Semiconductor parameters used for training (EBIC)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Beam position(μm)	6	6.9	0.2	5
Diffusion length L^{train} (μm)	3	3.9	0.03	31
Normalized surface recombination velocity S^{train}	2	4	0.03	67
Total number of samples used for training	31 \times 67=2077			

As illustrated in Table6.6, taking all the possible combinations of L^{train} , S^{train} results in a total number of samples of 2077 samples, therefore, 2077 input/output pairs are used for training the ANN.

A feed forward neural network consisting of one hidden layer is used, the hidden layer comprises 15 neurons and uses logarithmic sigmoid transfer functions whereas the output layer consists of 2 neurons (corresponds to the number of rows in $\mathbf{X}^{\text{train}}$) and uses linear transfer functions. Table6.7 illustrates the ANN parameters used. The training algorithm is the Levenberg Marquardt (see chapter 3).

Table6.7- ANN parameters used for training (EBIC)

ANN parameters		
Number of epochs	300	
Desired goal	10^{-3}	
Number of hidden layers	1	
Number of neurons in the hidden layer	15	uses logarithmic sigmoid transfer functions
Number of neurons in the output layer	2 (number of rows of $\mathbf{X}^{\text{train}}$)	uses linear transfer functions

The training curve of the ANN is shown in figure6.6; the desired goal was achieved after 8 epochs.

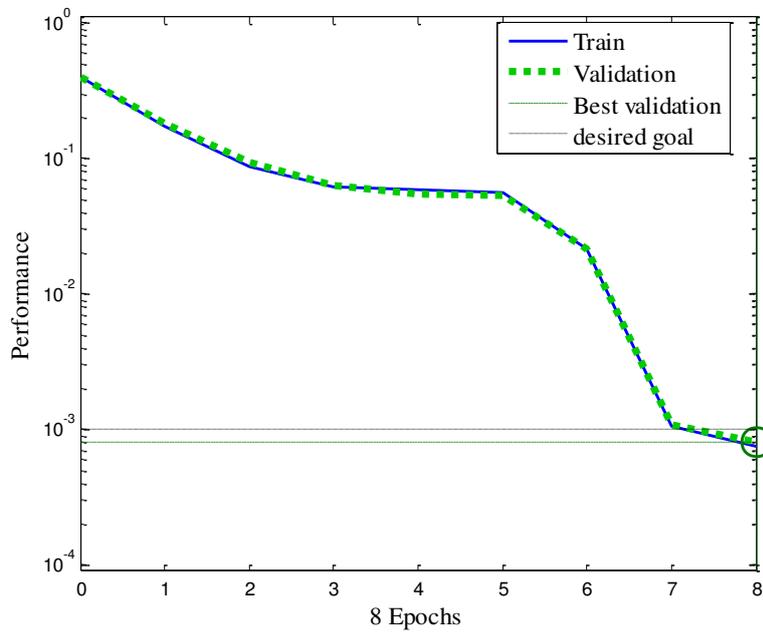


Figure6.6- Training curve of the ANN (EBIC)

Figure6.7 shows the percentage error between the theoretical training parameters and the parameters calculated by the ANN. The maximal percentage error for L and S is 2.1141% and 5.4334% respectively. This indicates that the training is very satisfactory.

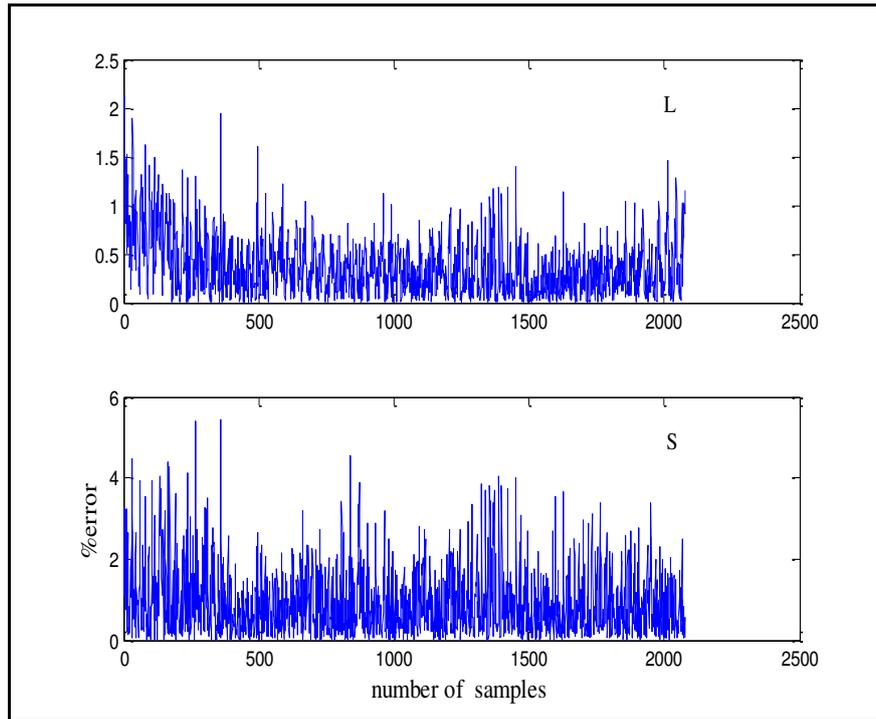


Figure6.7- Percentage error between theoretical training parameters and the parameters calculated by ANN (EBIC).

It is also instructive to observe the histogram plot of the error, is presented in Figure6.8.

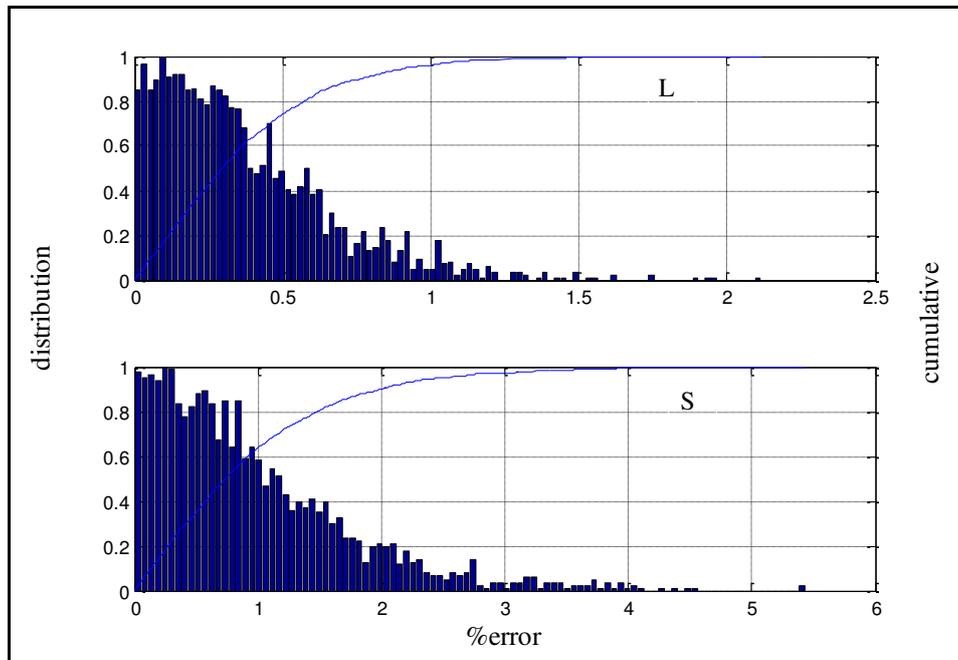


Figure 6.8 –PDF (bars) and CDF (solid line) of the percentage error for the training set (EBIC).

In figure 6.8 most of the errors are gathered around zero, this means that the ANN is able to learn the input/output relationship successfully.

From figure 6.8 we can detail the values of the error in 95% and 100% of the training samples for each parameter, the results are reported in table 6.8.

Table 6.8- percentage error of two parameters for 95% and 100% of the samples (EBIC).

Semiconductor parameter	For 95% of the samples, error is below	For 100% of the samples, error is below
L (μm)	1.5%	2.5%
S	3%	5.5%

The maximal error obtained in 95% of the training set is of 3% and for all the training set the maximal error is of 5.5%.

6.4.2 Testing the algorithm

Our ANN is tested using values of EBIC not seen before as illustrated in Table 6.9.

Table 6.9-Semiconductor parameters used for testing the ANN (EBIC)

Material Parameter	Minimal value	Maximal value	Sampling step size	Number of samples
Beam position (μm)	6	6.9	0.2	5
Diffusion length L^{test} (μm)	3+(0.01/2)	3.9	0.03	30
Relative quantum efficiency S^{test}	2+(0.05/2)	4	0.03	67
Total number of samples for the test	30×67=2010			

The total number of samples used for test is 2010 samples.

Figure6.9 illustrates the percentage error between the samples of the test EBIC and the EBIC calculated by ANN. The maximal percentage error for L and S is 2.3593% and 4.8562%, respectively.

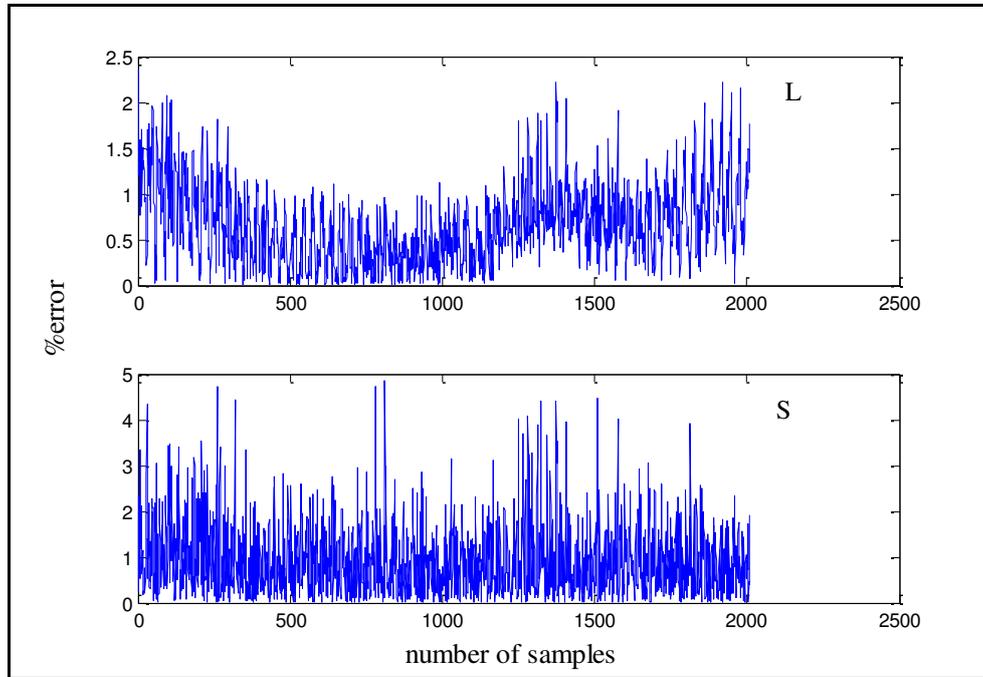


Figure6.9- Percentage error between the theoretical parameters and the parameters calculated by ANN for the test set (EBIC).

The histogram plot of the error and the CDF is shown in Figure6.10.

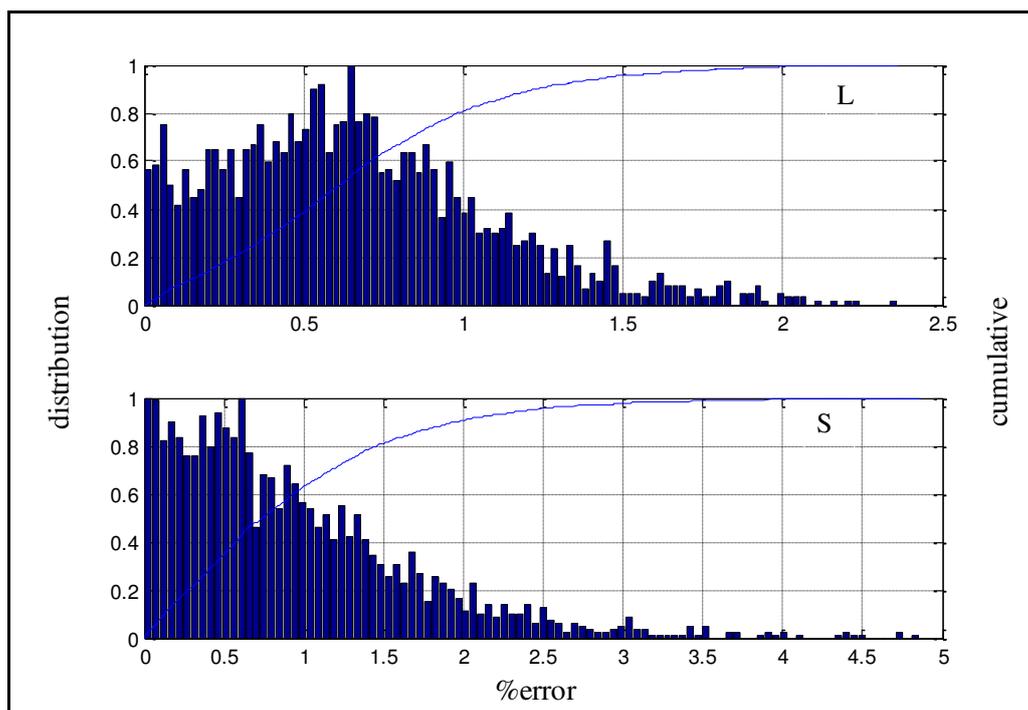


Figure6.10- PDF and CDF of the percentage error of L, S of the test set.

From figure6.10, we detail in table6.10 the values of the error in 95% and 100% of the samples.

Table6.10- Percentage error of the two parameters L and S for 95% and 100% of the samples

Semiconductor parameters	For 95% of the samples, error is below	For 100% of the samples, error is below
L (μm)	1.75%	2.5%
S	3%	5%

The maximal error obtained in 95% of the test cases is of 3% and is of 5% in 100% of the test cases.

6.5 Conclusion

A new parameter extraction algorithm based on ANNs as inverse modeling is presented and discussed in this chapter for the joint extraction of the semiconductor

related parameters. The algorithm is applied to the semiconductor parameter extraction using CL/EBIC signals. The results obtained can be summarized into the following points:

- For the simultaneous extraction of the five related parameters (L , α , Z_t , S , Q) from CL signal a set of parameter values is obtained with errors less than: 0.5% for L , 0.4% for α , 0.3% for S , 0.4% for Q and 0.4% for Z_t , in 95% of the cases, and an error less than 0.9% for L , 0.8% for α , 0.7% for S , 0.9 for Q , and 0.8% for Z_t , in 100% of the cases. The maximal error is 0.9% for all the cases, a result that clearly shows that the strategy proposed in this work is very successful in extracting the semiconductor parameters.
- For the simultaneous extraction of the two parameters (L , S) from EBIC a set of parameter values is obtained with error less than 1.75% for the diffusion length and less than 3% for the normalized surface recombination velocity in 95% of the cases, and an error less than 2.5% for the diffusion length and an error less than 5% for the normalized surface recombination velocity in 100% of the cases. The maximum error obtained for the extraction of the parameters in all cases is of 5%, this result shows that the strategy proposed in this work is very fruitful.

Chapter references

1. L. Cazzanty, M. Khan, F. Cerrina, "Parameter extraction with neural networks", Proceeding SPIE, 3332, 654. 1998

2. Puhmann, N. Oelgart, G. "Semiconductor Characterization by means of EBIC, Cathodoluminescence, and photoluminescence". *Physica Status Solidi*. (a) 122,705. 199031.
3. Oka. Kurniawan, "device parameters characterization with the use of EBIC", a thesis submitted for the degree of Doctor of Philosophy, Nanyang Technological University, 2008, supervised by Professor V. K. S. Ong.
4. Kurniawan. O, Ong. K. S, "Choice of generation volume models for electron beam induced current computation", *IEEE Transactions on Electron Devices* 56(5), 1094-1099, 2009.

Chapter 7

Semiconductor Parameter Extraction Using Genetic Algorithm

Contents

5.1 Introduction	
5.2 Parametre extraction using genetic algorithms	
5.3 Application to cathodoluminescence	
5.4 Application to EBIC	
5.5 Conclusion.....	
Chapter references.....	

7.1 Introduction

To the best of our knowledge, no work has been reported regarding the use of genetic algorithms (GA) in semiconductor parameter extraction from EBIC/CL signals. We present and discuss in this chapter a new parameter extraction algorithm based on genetic algorithms for the simultaneous extraction of related semiconductor parameters.

The proposed parameter extraction algorithm is applied to the simultaneous extraction of five related semiconductor parameters, that is: the diffusion length L , the absorption coefficient α , the dead layer thickness Z_t , the normalized surface recombination velocity S , and the relative quantum efficiency Q , from any steady state CL signal of a free defect semi-infinite semiconductor.

Similarly, the proposed parameter extraction algorithm is also applied to the simultaneous extraction of two related semiconductor parameters, that is, the diffusion

length L and the normalized surface recombination velocity S from any EBIC line scan in a normal collector configuration.

7.2 Parameter extraction using genetic algorithms

Assume that the nonlinear functional relationship is given by: $\mathbf{Y} = F(X_1, X_2, \dots, X_n)$; where X_i $i=1, \dots, n$ represents the input vector and \mathbf{Y} is the output vector (signal). The parameter extraction algorithm based on genetic algorithms can be summarized into the following steps:

7.2.1 Initialize the parameters

The range over which the parameters (X_1, X_2, \dots, X_n) vary is set as: $\mathbf{X}_1 = [X_1^{\text{lower}}, X_1^{\text{upper}}]$, $\mathbf{X}_2 = [X_2^{\text{lower}}, X_2^{\text{upper}}]$... $\mathbf{X}_n = [X_n^{\text{lower}}, X_n^{\text{upper}}]$, where lower and upper states for the lower and upper bounds of each parameter.

7.2.2 Define the objective function

The objective function \mathcal{G} is defined as the difference in terms of the Euclidian distance between a certain measured signal Y^{measured} for which the parameters $(X_1^*, X_2^*, \dots, X_n^*)$ are to be determined and the theoretical Y . This objective function can be written as follows:

$$\mathcal{G} = \left\| Y - Y^{\text{measured}} \right\|_2 \quad (7-1)$$

7.2.3 Apply the genetic algorithm

The genetic algorithm is applied through the following steps (see chapter 4):

- An initial population is defined by an $N_{\text{pop}} \times N_{\text{var}}$ matrix, where N_{pop} is the number of chromosomes and N_{var} is the number of variables.

- The objective function defined previously is evaluated for all chromosomes in the population.
- The GA operators (selection, crossover, and mutation) are performed.
- The algorithm stops when one of the stopping criteria is met (i.e. an acceptable solution is reached or a certain number of iterations is exceeded).

7.2.4 Extract the solution

The solution exhibiting the best fitness value is considered as the final extracted set of parameters ($X_1^*, X_2^* \dots X_n^*$).

The different steps of the proposed parameter extraction algorithm can be summarized in the flowchart depicted in figure7.1.

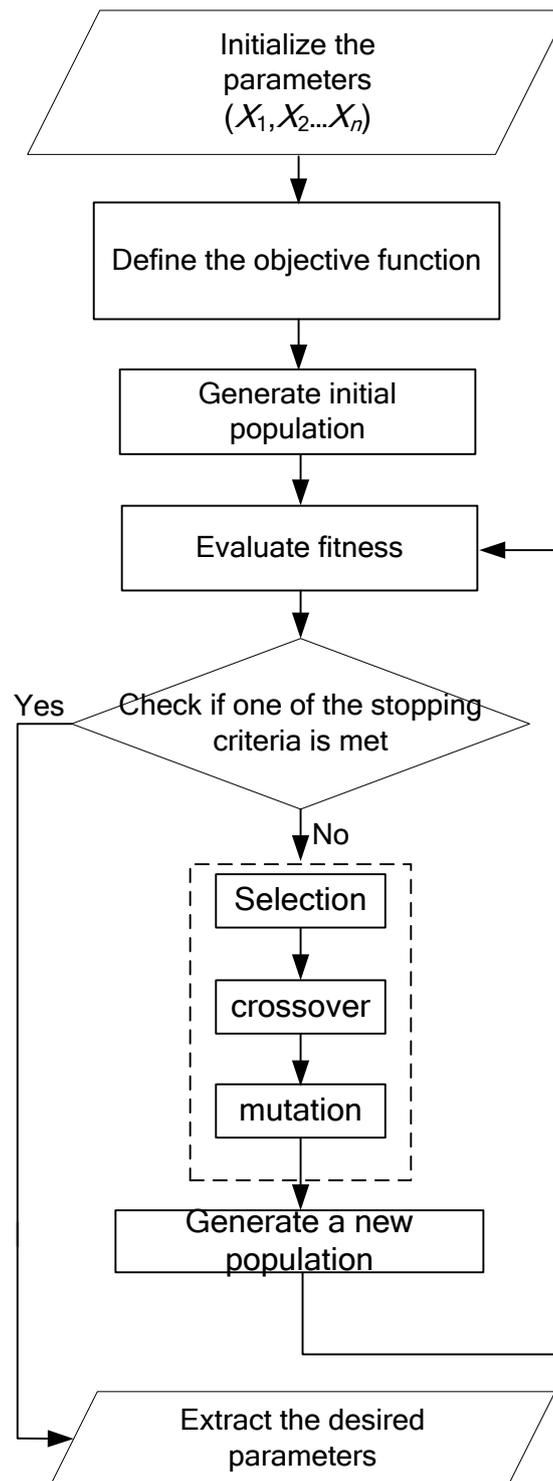


Figure7.1. Flowchart of the proposed parameter extraction algorithm based on GA.

7.3 Application to cathodoluminescence

We present in this section the application of the parameter extraction algorithm to the simultaneous extraction of five parameters: the absorption coefficient α , the diffusion length L , the normalized surface recombination velocity S , the dead layer thickness Z_t , and the relative quantum efficiency Q , directly from any steady state CL signal of a free defect semi-infinite semiconductor (Hergert's et al model; see chapter2).

The first step in the proposed algorithm is to initialize the semiconductor parameters (α , L , Z_t , Q , S) where upper and lower bounds are selected from intervals provided in [1] and are detailed in Table 7.1.

Table7.1. Lower and upper bounds of semiconductor parameters (CL)

Parameter	Lower bound	Upper bound
Energy beam (KeV)	10	50
Diffusion length $L(\mu\text{m})$	0.73	1.78
Absorption coefficient $\alpha (\mu\text{m}^{-1})$	0.785	0.825
Normalized surface recombination velocity S	3	6
Dead layer thickness $Z_t(\mu\text{m})$	0.03	0.18
Relative quantum efficiency Q (arbitrary)	5.15	5.45

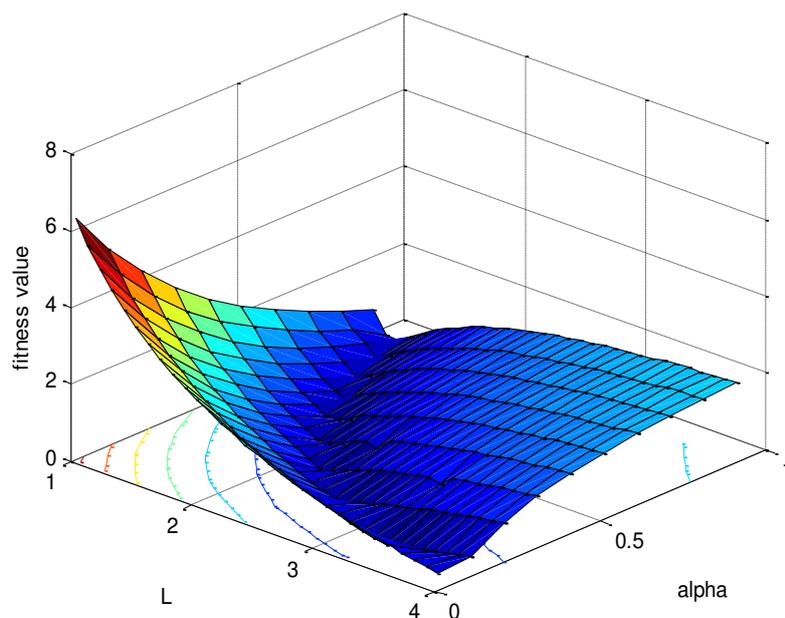
We consider an n-type GaAs semiconductor, with a critical angle of 16° and an atomic number of 32. The electron beam is perpendicular to the surface of the semiconductor sample.

The GA parameters used in the simulation are detailed in table7.2.

Table7.2. GA parameters (CL)

	Parameter	Value
Initial population options and other GA options	Chromosome Length	5 (number of semiconductor)
	Initial population size	1000
	Creation function	Uniform
	Fitness scaling	Rank
Reproduction options	Selection function	Roulette wheel
	Elite count	2
	Crossover function	Scattered
	Crossover fraction	0.8
	Mutation function	Gaussian (mean=0, scale=1, shrink=1)
Stopping criteria options	Number of generations	100
	Stall generation	50
	Function tolerance	1e-6

In order to have an idea about the objective function to minimize, a typical example is plotted below in figure7.2 for the two parameters only, that is, L and α :

**Figure7.2 Cost surface of the objective function of the two parameters L and α .**

It is clear that the objective function for L and α is very similar to the Rosenbrock (banana) function [2], which is one of the most difficult test functions in the optimization field. This provides us with an indication about the difficulty of the objective function when considering other parameters such as, S , Q and Z_t .

Figures 7.3.a and 7.3.b show the best and mean fitness values and the average distance between individuals.

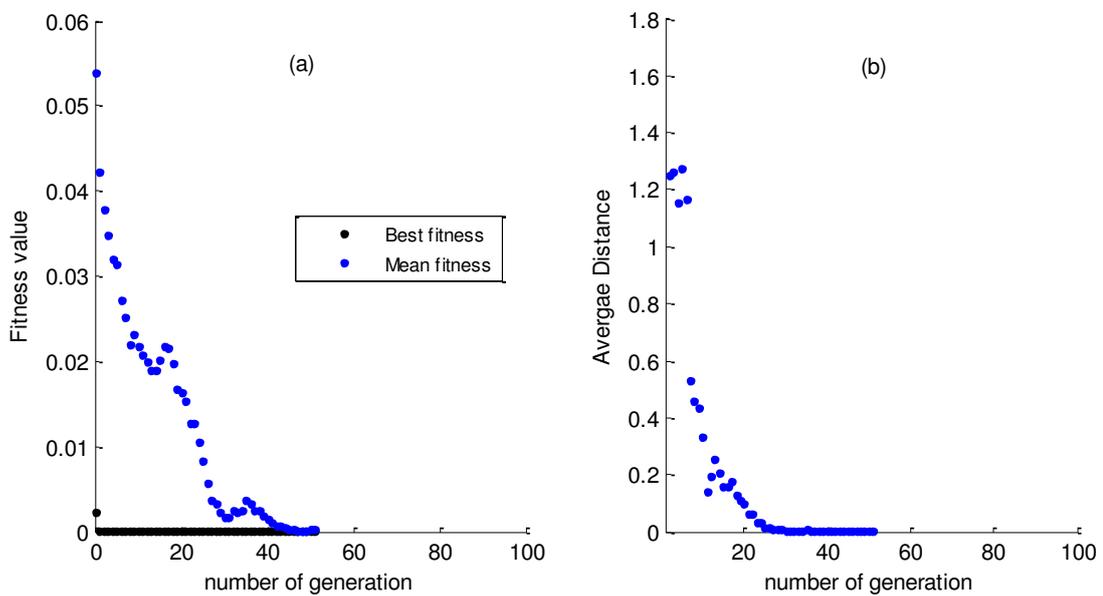


Figure 7.3. Convergence of the GA: (a) best and mean fitness, (b) average distance between individuals.

Figure 7.3.a illustrates the best and mean fitness values for each generation. The mean fitness value and the best fitness value become equal when the GA converges.

Figure 7.3.b illustrates the average distance between individuals; it shows how the individuals explore the search space and converge to a single point that can be local or a global minimum.

The parameter extraction algorithm is run 100 times (the reason behind is that the performance of the GA is random in the sense that it does not give the same results at

each run) and the Probability Density Function (PDF) / Cumulative Density Function (CDF) of the percentage error of each extracted parameter L, α , Q, S and Zt is evaluated.

The initial population of the GA algorithm changes from one run to another (it is generated randomly using a uniform distribution with bounds specified in table7.1). This is because the genetic algorithm doesn't have any prior information about the objective function and therefore the best strategy in this case is to generate the population using a uniform distribution. However, if some apriori information about the objective function is available then other types of distributions can be used to generate the initial population.

In the following, the Probability Density Function (PDF) / Cumulative Density Function (CDF) of the percentage error of each extracted parameter L, α , Q, S and Zt is plotted in figure7.4.

The percentage error of each extracted parameter is given by:

$$\%error = \frac{|Parameter_{No\ minal} - Parameter_{Calculated}|}{Parameter_{No\ minal}} \times 100 \quad (7-2)$$

where Parameter in (7-2) states for one of the parameters to be extracted.

Parameter.

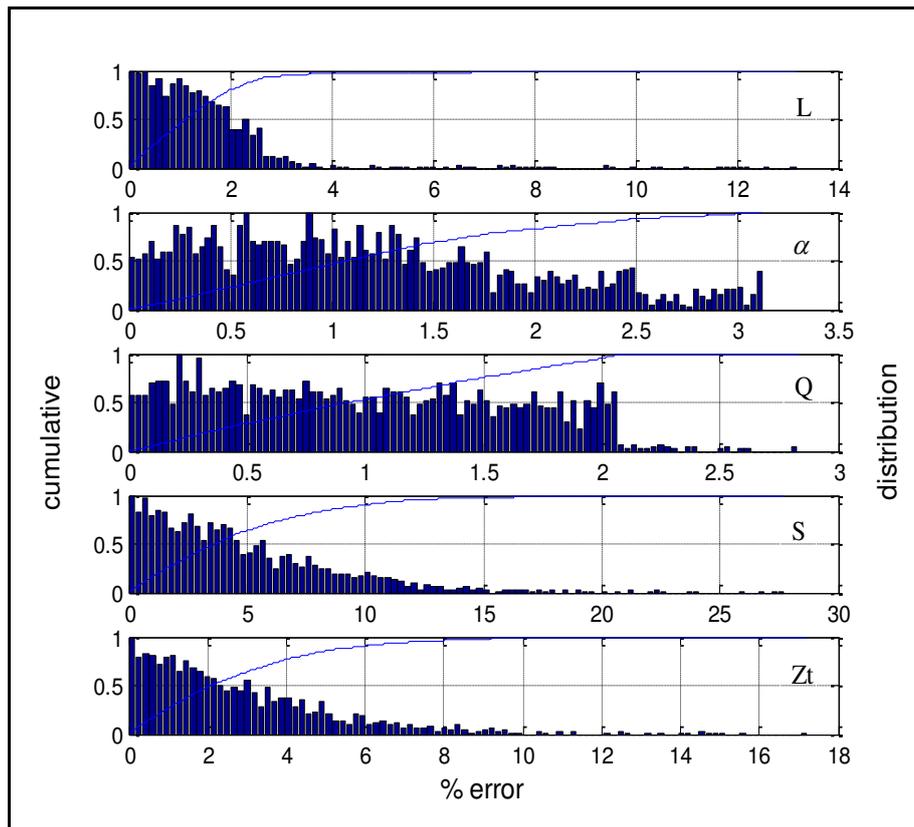


Figure7.4. PDF (bars) and CDF (solid line) of the percentage error for each of the five parameters L , α , S , Q , and Z_t .

Using Figure7.4, the percentage error in 95% and 100% of the runs for each parameter is detailed in Table7.3.

Table7.3. Maximum percentage error of the five parameters L , α , S , Q , and Z_t for 95% and 100% of the runs, respectively.

Semiconductor parameter	For 95% of the runs, error is below	For 100% of the runs, error is below
L (μm)	3.1 %	14 %
α (μm^{-1})	2.6 %	3.5 %
Q (arbitrary units)	2 %	3 %
S	12 %	30 %
Z_t (μm)	7.3 %	18 %

The maximal percentage error in 95% of the cases is of 12%.

Effect of the initial population size

Table 7.4 illustrates the percentage error in 95% and 100% of the runs for each parameter for three different initial population sizes: 1000 (presented in table 7-6), 100, and 10.

It is clear that in general the larger the population size the less the percentage error. The size of the initial population obviously affects the performance of the extraction algorithm, and it is expected that the performance of the extraction algorithm improves (in a statistical sense) as the size of the population increases, that is, as the initial population increases the performance of the extraction algorithm improves as the number of runs tends to infinity. This is justified by the fact that a larger population size is able to search a larger space of the objective function and therefore has more chance to find a good solution.

Table 7.4. Maximum percentage error of the five parameters L , α , S , Q , and Z_t for 95% and 100% of the runs, respectively, with different initial population size.

Initial size population	Semiconductor parameters	For 95% of the runs, error is below	For 100% of the runs, error is below
10	L (μm)	22	30
	α (μm^{-1})	2.45	2.5
	Q (arbitrary units)	2.8	3
	S	31.5	35
	Z_t (μm)	39	60
100	L (μm)	20	25
	α (μm^{-1})	2.4	2.5
	Q (arbitrary units)	2.75	3
	S	31	35
	Z_t (μm)	14.5	18
1000	L (μm)	3.1 %	14 %
	α (μm^{-1})	2.6 %	3.5 %
	Q (arbitrary units)	2 %	3 %
	S	12 %	30 %
	Z_t (μm)	7.3 %	18 %

7.4 Application to EBIC

We present now the application of the parameter extraction algorithm described above to the simultaneous extraction of the diffusion length L and the normalized surface recombination velocity S from an EBIC line scan in a normal collector configuration (Donolato's model; see chapter 2).

The first step in the proposed algorithm is to initialize the semiconductor parameters (L , S) where upper and lower bounds are selected from intervals provided

in [3,4] and are detailed in Table7.5. The sample material is chosen to be silicon (a density of 2.33 g/cm^3). The electron beam is perpendicular to the surface of the semiconductor sample, the line scans the region outside the junction, and the energy beam is equals to 14KeV which gives for the energy range a value of $1.7\mu\text{m}$.

Table7.5. Lower and upper bounds of semiconductor parameters (EBIC)

Parameter	Lower bound	Upper bound
Beam position (μm)	6.2	6.5
Diffusion length $L(\mu\text{m})$	3	3.1
Normalized surface recombination velocity	2	2.1

The GA parameters used in the simulation are detailed in table7.6.

Table7.6. GA parameters (EBIC)

	Parameter	Value
Initial population options and other GA options	Chromosome Length	2 (number of semiconductor parameters)
	Initial population size	20
	Creation function	Uniform
	Fitness scaling	Rank
Reproduction options	Selection function	Roulette wheel
	Elite count	2
	Crossover function	Scattered
	Crossover fraction	0.8
	Mutation function	Gaussian (mean=0, scale=1, shrink=1)
Stopping criteria options	Number of generations	10
	Stall generation	50
	Function tolerance	1e-6

The parameter extraction algorithm is executed 10 times and the PDF and CDF of the percentage error of each extracted parameter L, and S is plotted in figure7.5. The percentage error of each extracted parameter is evaluated using (7.2).

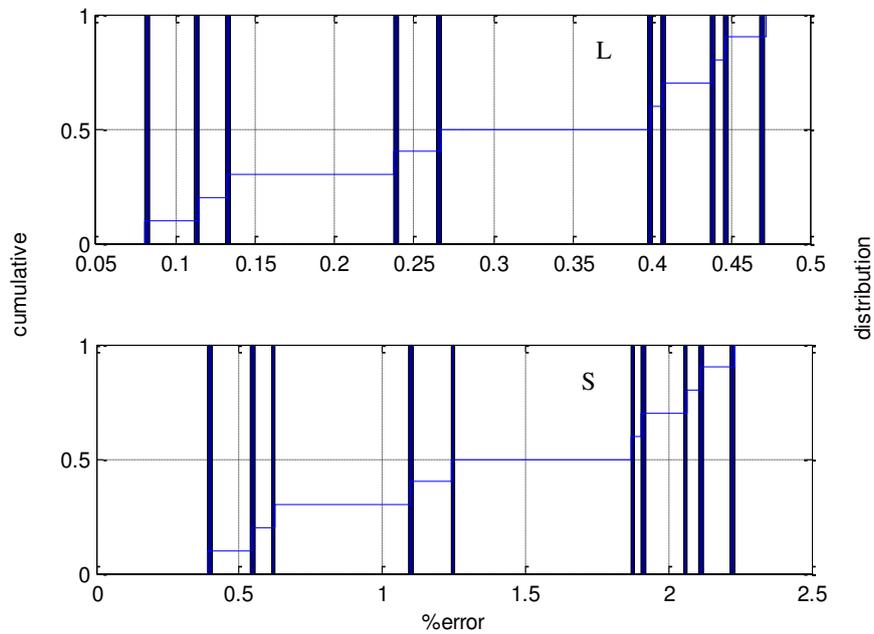


Figure7.5. PDF and CDF of the percentage error for each of the two parameters L, and S.

Using Figure7.5, the percentage error in 95% and 100% of the runs for each parameter are detailed in Table7.7.

Table7.7. Maximum percentage error of the two parameters L and S for 95% and 100% of the runs, respectively.

Semiconductor parameter	For 95% of the runs, error is below	For 100% of the runs, error is below
Diffusion length L (μm)	0.48 %	0.5 %
Normalized surface recombination velocity S	2.25 %	2.5 %

The results show that for 95% of the runs the maximum error for L is 0.48% and for S is 2.25%, and for 100% of the runs the maximum error for L is 0.5%, and for S is 2.5%.

7.5 Conclusion

In this chapter we have presented a new parameter extraction algorithm based on genetic algorithms for jointly extracting semiconductor related parameters, the model is applied to the simultaneous extraction of five semiconductor parameters from CL/EBIC signals. We can summarize the results into the following points:

- For the simultaneous extraction of the five related parameters (L, α , Zt, Q) from a CL signal, a set of parameter values is obtained with errors less than: 3.1% for L, 2.6% for α , 2% for Q, 12% for S and 7.3% for Zt, in 95% of the cases, and an error values less than: 14% for L, 3.5% for α , 3% for Q, 30% for S and 18% for Zt, in 100% of the cases. The results show that for 95% of the runs the largest error is that of S, this is reasonable as these parameters are jointly extracted. One way to reduce the percentage error is to reduce the dimension of the search space by removing some parameters (i.e. L, S) from the extraction process using CL and determining them using other techniques such as EBIC. This solution is actually considered for future work
- The effect of the population size was discussed here as well; it is found that the percentage error becomes smaller for larger population sizes. This result is clearly noticeable in the 95% of the runs. This is expected to be true also for 100% of the runs if the number of runs tends to infinity.

- For the simultaneous extraction of the two parameters (L,S) from EBIC a unique set of parameters is obtained with error less than 0.48% for L, and 2.25% for S in 95% of the cases, and an error less than: 0.5% for L, and 2.5% for S in 100% of the cases. Since the parameters are jointly extracted the results obtained show that the proposed strategy is very successful.

Chapter references

1. D. S. H. Chan, K. L. Pey, and J. C. H. Phang, "Semiconductor parameter extraction using cathodoluminescence in the scanning electron microscope", IEEE transactions on electron devices, Vol. 40, No. 8, Aug 1993.
2. H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function." Computer J. **3**, 175-184, 1960.
3. L. Cazzanty, M. Khan, F. Cerrina, "Parameter extraction with neural networks", Proceeding SPIE, 3332, 654. 1998
4. Oka. Kurniawan, "device parameters characterization with the use of EBIC", a thesis submitted for the degree of Doctor of Philosophy, Nanyang Technological University, 2008, supervised by Professor V. K. S. Ong.

Chapter 8

Conclusion and future work

Contents

8.1 Conclusion	
8.2 Future work	

8.1. Conclusion

In this work, the semiconductor parameter extraction problem is formulated as an optimization problem. By doing so, powerful optimization tools such as neural networks and genetic algorithms can be used to find high-quality solutions with a reasonable cost.

In this work, the Cathodoluminescence and EBIC signals in the Scanning Electron Microscopy are used along with artificial neural networks (ANN) and genetic algorithms (GA) to extract the semiconductor parameters (absorption coefficient, diffusion length, dead layer thickness, relative quantum efficiency and the normalized surface recombination velocity) with a relatively small percentage error.

The new techniques developed in this dissertation exhibit many advantages such as the simultaneous obtainment of near-optimum values for the extracted related semiconductor parameters and moderate computational complexity.

This work has stepped forward into the use of powerful optimization tools like artificial neural networks and genetic algorithms in the field of scanning electron microscopy which is a great tool in the characterization of the semiconductor materials and devices.

Simulation results of the parameter extraction algorithms for CL/EBIC signals using ANNs and genetic algorithms indicated that these techniques are successful and of reasonable cost. Moreover, the parameter extraction algorithms based on neural networks do not require a theoretical model and can be used directly with experimental data. This is very useful because in practice theoretical models have limitations due to the approximations and assumptions used during the development of these models. Moreover theoretical models do not take into consideration measurement errors and hence parameter extraction techniques based on these models do not give accurate results with experimental data. Therefore, our approach represents a good alternative to these parameter extraction techniques.

7.2. Future work

This work left in the mind of the writer a number of questions, these could be answered by further improvements of both parameter extraction algorithms and optimization techniques.

The investigation of electronic and optical properties in semiconductor characterization is of fundamental importance, thus a joint use of CL/EBIC signals along with the techniques presented in this work can offer possibilities for determining reliable quantitative information on electronic, optical properties and electrical activity of localized defects.

Our techniques can be extended to nano-contacts and nano-scale characterization, which are promising fields of research since most of the microscopes used in nanoscience and nanotechnology are electronic microscopes (SEM, STEM...).

Authors publications

1. S.Soualmia, A.Bouldjedri, A.Benhaya, "Semiconductor parameter extraction using cathodoluminescence and genetic algorithms", *Materials Science in Semiconductor Processing*, vol.14 (March 2011) pp 62-68.
2. S.Soualmia, A.Bouldjedri, A.Benhaya, "a new approach in Semiconductor parameter extraction using cathodoluminescence and artificial neural networks", *Scanning*, under press.2011.
3. Four other papers are under preparation.

Appendix A: Monte Carlo simulation of electron trajectories

1. Theoretical background

The physics underlying the interaction of non-relativistic electrons with specimen is detailed here.

The Coulomb force is the fundamental force which describes the interaction between the incident electrons and the bombarded sample particles, and ionization is the principal energy loss mechanism and elastic collisions with the nuclei produce the majority of the relatively large angular deflection.

The "slowing down approximation" is the assumption that the energy lost per unit path length dE/dS is a given function of the energy, the atomic number and the angular deflection.

The energy lost per unit distance dS by the electron is given by the Bethe formula [1]:

$$\frac{dE}{dS} = -7.83 \left(\frac{\rho Z}{AE} \right) \ln \left(\frac{174E}{Z} \right) \text{ (KeV / } \mu\text{m)} \quad (\text{A-1})$$

Where : Where, $\rho(\text{g/cm}^3)$ is the sample density, and $A(\text{g})$ is the atomic weight of the sample.

A shielded Rutherford cross section is given by [1]:

$$\frac{d\sigma}{d\Omega} = \frac{Z(Z+1)e^4}{p^2 v^2} \frac{1}{(1 - \cos\theta + 2\beta)^2} \quad (\text{A-2})$$

Where $\beta = 0.25(1.12\hbar/p)^2$, $\lambda_0 = Z^{1/3}/0.885a_0$, $p = mv$ (the electron momentum), and a_0 is the Bohr radius. θ is the scattering angle.

The integration of (A-1) over all solid angles gives the total screened cross section [1]:

$$\sigma_T = \int_0^\pi \int_0^{2\pi} d\Omega \left(\frac{d\sigma}{d\Omega} \right) = \frac{\pi}{\beta(1+\beta)} \frac{Z(Z+1)e^4}{p^2 v^2} \quad (\text{A-3})$$

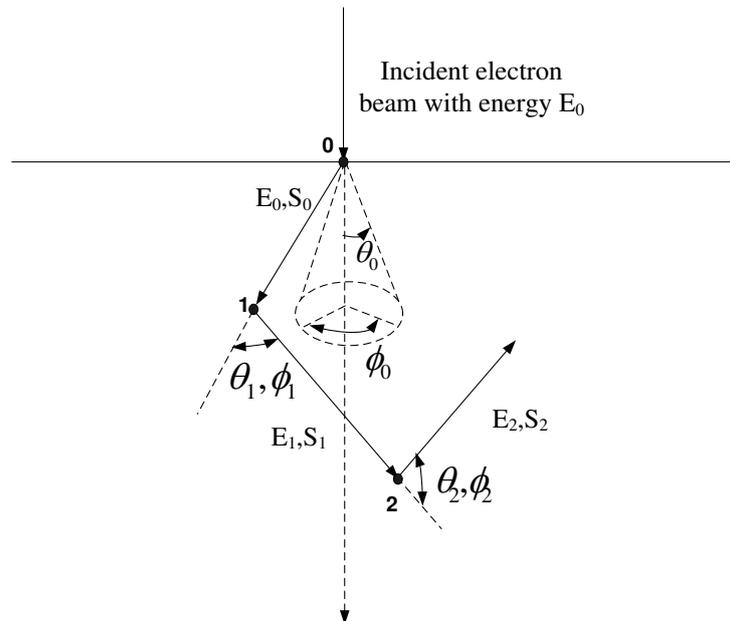
If the electron beam is exponentially attenuated the mean free path λ is given by [1]:

$$\lambda = \left(\int_0^\infty s \exp[-\rho N_0 \sigma_{T/A} s] ds \right) \times \left(\int_0^\infty \exp[-\rho N_0 \sigma_{T/A} s] ds \right)^{-1} \quad (\text{A-4-a})$$

$$\lambda = 1.02 \beta(1+\beta) A T^2 / Z(Z+1) \rho \quad (\text{A-4-b})$$

Where N_0 is Avogadro's number.

The path of an electron as it moves inside the matter is shown by figureA-1:



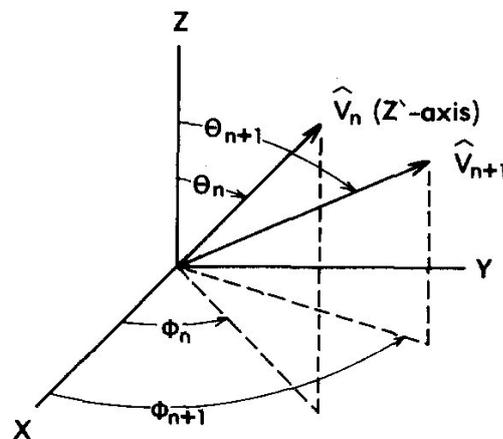
FigureA-1. Schematic geometry of the initial steps of electron scatterings.

The azimuthal angle is assumed to be a uniformly distributed random mangle between 0 and 2π rad at each scattering event.

The two frames of reference to consider are:

- The laboratory frame which is the fundamental frame is attached to the sample.
- The scattering frame which co-moves with the electron.

The geometrical connection between the two frames is shown in figure A-2.



FigureA-2 laboratory and scattering frames.

At a given step the direction of the electron is defined by a unit vector \mathbf{V}_n . the angles θ_n, ϕ_n specify \mathbf{V}_n relative to axes fixed in the laboratory frame.

The scattering angles θ and ϕ are defined in the scattering frame which uses \mathbf{V}_n to define the \mathbf{Z}' axis.

The electron travels in the direction \mathbf{V}_n until it undergoes a scattering defined by the scattering angle θ . The new direction of the electron in the laboratory frame is defined by the unit vector \mathbf{V}_{n+1} .

We want to find θ_{n+1} and ϕ_{n+1} in terms of θ_n, ϕ_n, θ and ϕ [1].

Using figure A-2 we have :

$$\mathbf{V}_n = \sin \theta_n \cos \phi_n \vec{i} + \sin \theta_n \sin \phi_n \vec{j} + \cos \theta_n \vec{k} \quad (\text{A-5-a})$$

$$\mathbf{V}_{n+1} = \sin \theta_{n+1} \cos \phi_{n+1} \vec{i} + \sin \theta_{n+1} \sin \phi_{n+1} \vec{j} + \cos \theta_{n+1} \vec{k} \quad (\text{A-5-b})$$

$$\mathbf{V}_{n+1} = \sin \theta \cos \phi \vec{i}' + \sin \theta \sin \phi \vec{j}' + \cos \theta \vec{k}' \quad (\text{A-5-c})$$

Where $\vec{i}', \vec{j}', \vec{k}'$ refer to the scattering frame.

The scattering frame is connected to the laboratory frame as follows:

$$\begin{aligned} \vec{k}' &= \mathbf{V}_n \\ \vec{j}' &= \mathbf{V}_n \times \vec{k} / |\mathbf{V}_n \times \vec{k}| \\ &= \mathbf{V}_n \times \vec{k} / \sin \theta_n \\ &= \sin \theta_n \vec{i} - \cos \phi_n \vec{j} \\ \vec{i}' &= \vec{j}' \times \vec{k}' \\ &= (\vec{k} - \cos \theta_n \mathbf{V}_n) / \sin \theta_n \end{aligned} \quad (\text{A-6})$$

Then:

$$\begin{aligned} \mathbf{V}_{n+1} \cdot \vec{k} &= \cos \theta_{n+1} \\ &= \cos \theta_n \cos \theta + \sin \theta_n \sin \theta \cos \phi \\ \mathbf{V}_{n+1} \cdot \mathbf{V}_n &= \cos \theta_{n+1} \\ &= \cos \theta_n \cos \theta_{n+1} + \sin \theta_n \sin \theta_{n+1} \cos(\phi_{n+1} - \phi_n) \end{aligned} \quad (\text{A-7})$$

Or:

$$\cos(\phi_{n+1} - \phi_n) = \frac{\cos \theta - \cos \theta_{n+1} \cos \theta_n}{\sin \theta_n \sin \theta_{n+1}} \quad (\text{A-8})$$

To determine $\sin(\phi_{n+1} - \phi_n)$ the vectors \mathbf{V}_n and \mathbf{V}_{n+1} are projected onto the x-y plan and the projections are named \mathbf{P}_n and \mathbf{P}_{n+1} respectively (figure A-3).

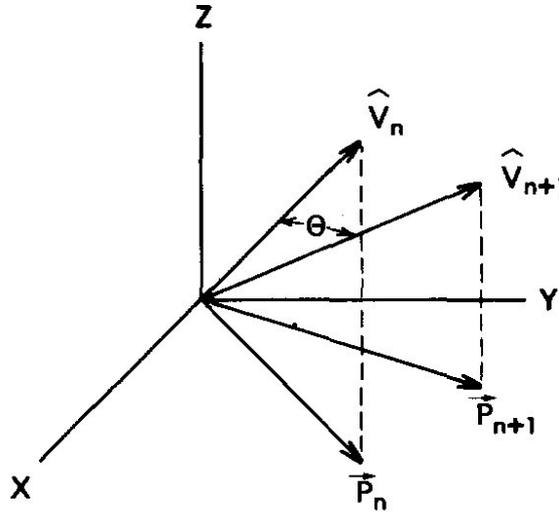


Figure A-3 projection onto the x-y plan.

We have:

$$\begin{aligned} \mathbf{P}_n \mathbf{P}_{n+1} &= |\mathbf{P}_n| |\mathbf{P}_{n+1}| \sin(\phi_{n+1} - \phi_n) \vec{k} \\ &= \sin \theta \sin(\phi_{n+1} - \phi_n) \vec{k} \end{aligned} \quad (\text{A-9})$$

Direct evaluation of $\mathbf{P}_n \times \mathbf{P}_{n+1}$ in terms of the angles θ_n , ϕ_n , θ and ϕ gives :

$$\mathbf{P}_n \mathbf{P}_{n+1} = -\sin \theta_n \sin \theta \sin \phi \vec{k} \quad (\text{A-10})$$

We finally have:

$$\begin{aligned}
 \cos \phi &= \cos \theta_n \cos \theta - \sin \theta_n \sin \theta \cos \phi \\
 \cos(\phi_{n+1} - \phi_n) &= \frac{\cos \theta - \cos \theta_{n+1} \cos \theta_n}{\sin \theta_n \sin \theta_{n+1}} \\
 \sin(\phi_{n+1} - \phi_n) &= \sin \theta \sin \phi / \sin \theta_{n+1}
 \end{aligned}
 \tag{A-11}$$

The angles θ_{n+1}, ϕ_{n+1} are now determined in terms of θ_n, ϕ_n and the scattering angles θ and ϕ .

2. Monte Carlo procedure

We follow one electron through one scattering event. Initially the electron direction of motion is given by $\mathbf{V}_n(\theta_n, \phi_n)$ and it has kinetic energy E_n . The angles θ, ϕ defines an elastic scattering event. The angles $\theta, \phi, \theta_n, \phi_n$ serve to define θ_{n+1}, ϕ_{n+1} and thus \mathbf{V}_{n+1} .

The normalized probability that the electron scatters through an angle θ is given by the use of (A-2) and (A-3):

$$\begin{aligned}
 P(\theta) &= \frac{1}{\sigma_T} \int_0^\theta \int_0^{2\pi} \left(\frac{d\sigma}{d\Omega} \right) d\Omega \\
 &= \frac{(1-\beta)(1-\cos\theta)}{(1+2\beta-\cos\theta)}
 \end{aligned}
 \tag{A-12}$$

Thus:

$$\cos \theta = \frac{1-2\beta P(\theta)}{[1+\beta-P(\theta)]}
 \tag{A-13}$$

Now θ, ϕ can be determined by:

$$\begin{aligned}\cos \theta &= \frac{1 - 2\beta R_\theta}{1 + \beta - R_\theta} \\ \phi &= 2\pi R_\phi\end{aligned}\tag{A-14}$$

Where R_θ, R_ϕ are random numbers between 0 and 1.

If the number of electrons which persist after a distance S then:

$$N(S) = N(0)e^{-S/\lambda}\tag{A-15}$$

λ is the mean free path.

The probability that an electron interacts after traveling a distance S is then:

$$\begin{aligned}P(S) &= \frac{[N(0) - N(S)]}{N(0)} \\ &= 1 - e^{(-S/\lambda)}\end{aligned}\tag{A-16}$$

And so:

$$S = -\lambda \ln[1 - P(S)]\tag{A-17}$$

By choosing for $[1 - P(S)]$ random numbers R_S the path length S traveled by an electron can be found as:

$$S = -\lambda \ln R_S\tag{A-18}$$

If \mathbf{R}_n is the position of the electron in the laboratory frame before the n th scattering, and E_n is the kinetic energy of the electron for the n th scattering than the procedure can be summarized as follows:

1. Given θ_n, ϕ_n .
2. Generate three random numbers: R_θ, R_ϕ, R_S .
3. (A-14).
4. $\cos(\theta_{n+1}), \tan(\phi_{n+1} - \phi_n)$ from (A-11). And S_n from (A-18) and (A-4).
5. $\mathbf{R}_{n+1} = \mathbf{R}_n + S_n (\sin \theta_{n+1} \cos \phi_{n+1} \vec{i} + \sin \theta_{n+1} \sin \phi_{n+1} \vec{j} + \cos \theta_{n+1} \vec{k})$.
6. $E_{n+1} = E_n - \left| \frac{dE}{dS} \right| S_n$.
7. $\theta_{n+1} \rightarrow \theta_n, \phi_{n+1} \rightarrow \phi_n$.

3. MATLAB codes

```

%=====
%= this code gives electron trajectories for any angle of incidence =
%=                               and any beam energy                               =
%=====
clear
clc
%close all
format long

%=== definition of constants =====
Z=32;           % atomic number of the target(Ge);
A=72.59;       % atomic weight of the target (gm/mole);
rho=5.3;       % density of the target (gm/cm3);
thickness=2;   % target thickness(microm);
E0=15;%[5 10 20 50]; % electron beam energy (KeV);
length_E0=length(E0);
number_e=200;  %number of electrons;

%=====
%===== the program starts here =====
%=====
%===== the iterations starts here =====

```

```

for j=1:length_E0
    for k=1:number_e
        x(1)=0; % initial position of the electron (point of impact);
        y(1)=0;
        z(1)=0;
        teta_n(1)=90; % initial scattering angle at the scattering
frame (normal incidence)=
        phi_n(1)=0; %initial azimuthal angle at the scattering
frame;
        E(1)=E0(j); %initial kinetic energy of the electron (KeV);
        i=1;
        while ((E(i)>0.5)&&(z(i)>=0)&&(z(i)<thickness))

            %==== generation of random numbers =====
            random_teta(i)=rand(1);
            random_phi(i)=rand(1);
            random_s(i)=rand(1);
            %= calculation of beta the screening parameter=
            beta(i)=3.4*10^(-3)*Z^(2/3)*E(i)^(-1);
            %== calculation of the scattering and the azimuthal
            % angles at the scattering frame.
            teta(i)=acos(1-((2*beta(i)*random_teta(i))/(1+beta(i)-...
                random_teta(i)))));
            phi(i)=2*pi*random_phi(i);
            %== calculation of the mean free path in microm =
            lamda(i)=1.02*beta(i)*(1+beta(i))*E(i)^2*A/(Z*(Z+1)*rho);
            %=== calculation of the path traveled by the electron =
            s(i)=-lamda(i)*log(random_s(i));
            %== calculation of the scattering and azimuthal angles
            % of the step i+1 at the laboratory frame;
            teta_n(i+1)=acos(cos(teta_n(i))*cos(teta(i))-...
                (sin(teta_n(i))*sin(teta(i))*cos(phi(i))));
            phi_n(i+1)=phi_n(i)+asin((sin(teta(i))*sin(phi(i)))/...
                sin(teta_n(i+1)));
            %== calculation of the new position at the step i+1 ===
            x(i+1)=x(i)+s(i)*(sin(teta_n(i+1))*cos(phi_n(i+1)));
            y(i+1)=y(i)+s(i)*(sin(teta_n(i+1))*sin(phi_n(i+1)));
            z(i+1)=z(i)+s(i)*cos(teta_n(i+1));
            %== calculation of electron's energy at the step i+1 =
            delta_E(i)=s(i)*abs(-
7.83*rho*Z*log(174*E(i)/Z)/(A*E(i)));
            E(i+1)=E(i)-delta_E(i);
            i=i+1;
        end
        k=k
        plot(x,z);
        hold on;
        xlabel('x');
        ylabel('z');
        clear random_teta random_phi random_s beta teta phi lamda ...
            s teta_n phi_n x y z delta_E E
    end
    figure;
    j=j
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%=====
%==== this code calculates the energy dissipation of the ==
%== incident electrons for different energy beam values ==
%=====
clear
clc
close all
format long
%===== definition of constants =====
Z=14; % atomic number of the target (silicon);
A=28.0855; % atomic weight of the target (gm/mole);
rho=2.33; % density of the target (gm/cm3);
thickness=3;% target thickness(microm);
E0=[10 15 20];% electron beam energy (KeV);
length_E0=length(E0);
number_e=100; %number of electrons;

%=====
%===== the program starts here =====
%=====

%=== the iterations starts here =====
z_s=0.01;%division step of thickness
z_axis=[0:z_s:thickness];%division vector of thickness;
buffered_z=buffer(z_axis,2,1,'nodelay');%buffered of the z_axis;
size_buffered_z=size(buffered_z);%size of the bufferd vector;
for l=1:length_E0
%definition of the sum energy dissipation vector into each
layer (KeV);
    sum_delta_E=zeros(1,size_buffered_z(2));
%=====
    for k=1:number_e
        x(1)=0;% initial position of the electron;
        y(1)=0;
        z(1)=0;
        E(1)=E0(l);%initial kinetic energy(KeV);
%initial scattering angle at the scattering frame (normal incidence);
        teta_n(1)=0;
%initial azimuthal angle at the scattering frame;
        phi_n(1)=0;
        i=1;
        % disp('while starts here');
        while ((E(i)>0.5)&&(z(i)>=0)&&(z(i)<thickness))
%= generation of random numbers =====
            random_teta(i)=rand(1);
            random_phi(i)=rand(1);
            random_s(i)=rand(1);
%= calculation of beta the screening parameter =====;
            beta(i)=3.4*10^(-3)*Z^(2/3)*E(i)^(-1);
%= calculation of the scattering and the azimuthal angles at the
%scattering frame;
            teta(i)=acos(1-((2*beta(i)*random_teta(i))/...
                (1+beta(i)-random_teta(i))));
            phi(i)=2*pi*random_phi(i);
%= calculation of the mean free path in microm =
            lamda(i)=1.02*beta(i)*(1+beta(i))*E(i)^2*A/(Z*(Z+1)*rho);
%= calculation of the path length traveled by the electron (micro m);
            s(i)=-lamda(i)*log(random_s(i));
%= calculation of the scattering and azimuthal angles of the step i+1
%at the laboratory frame;

```

```

        teta_n(i+1)=acos(cos(teta_n(i))*cos(teta(i))-...
            (sin(teta_n(i))*sin(teta(i))*cos(phi(i))));
        phi_n(i+1)=phi_n(i)+asin((sin(teta(i))*sin(phi(i)))/...
            sin(teta_n(i+1)));
%= calculation of the new position of the electron at the step i+1 =
        x(i+1)=x(i)+s(i)*(sin(teta_n(i+1))*cos(phi_n(i+1)));
        y(i+1)=y(i)+s(i)*(sin(teta_n(i+1))*sin(phi_n(i+1)));
        z(i+1)=z(i)+s(i)*cos(teta_n(i+1));
%= calculation of the energy of electron at the step i+1; =====
        delta_E(i)=s(i)*abs(-
7.83*rho*Z*log(174*E(i)/Z)/(A*E(i)));
        E(i+1)=E(i)-delta_E(i);
%= testing z(i+1)value at each iteration to calculate the energy
% dissipation at each layer
        for j=1:size_buffered_z(2)
            if
(z(i+1)<=buffered_z(2,j))&&(z(i+1)>buffered_z(1,j))
                sum_delta_E(j)=sum_delta_E(j)+delta_E(i);
            end
        end
        i=i+1;
    end
    k=k
    clear random_teta random_phi random_s beta teta phi lamda...
        s teta_n phi_n x y z delta_E E
    end
    sum_delta_E_norm=sum_delta_E/max(sum_delta_E);
%= making the curve smooth=====
    smooth_curve=smooth(sum_delta_E,500,'lowess');
    hold on
    %figure
    plot([1:size_buffered_z(2)],smooth_curve)
    xlabel('depth_s')
    ylabel('NDE_s')
    % hold on
    % figure
    % plot([1:size_buffered_z(2)],sum_delta_E)
    % xlabel('depth')
    % ylabel('NDE')
    % hold on
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Appendix reference

1. W. Williamson. Jr, G. C. Duncan, "Monte Carlo simulation of nonrelativistic electron scattering", Am. J. Phys, **54** (3), March 1986.