Blind Restoration of Radiological Images Using Hybrid Swarm Optimized Model Implemented on FPGA

Slami Saadi[1], Abderrezak Guessoum[2], Maamar Bettayeb[3], and Kamel Abdelhafidi[4]
[1]Department of Technology, University of Djelfa, Algeria
[2]Department of Electronics, University of Blida, Algeria
[3]Department of Electrical and Computer Engineering, University of Sharjah, UAE
[4]LMetallic and Semiconducting Materials, University of Biskra, Algeria

Abstract: *Image restoration step is important in many image processing applications. In this work, we attempt to restore radiological images degraded during acquisition and processing. Details of the work, carried out to optimize a Neural Network (NN) for identifying an AutoRegressive Moving Average (ARMA) model used for nonlinearly degraded image restoration, are presented in this paper. The degraded image is expressed as an ARMA process. To improve the learning performance, the NN is fast trained using a hybrid swarm intelligence optimization approach based on the synergy of Particle Swarm (PSO) and Bacterial Foraging (BFO) Algorithms, which is compared with other training techniques such as: The back propagation, Quasi-Newton and Levenberg-Marquardt Algorithms. Both original image and blur function are identified through this model. The optimized ARMA-NN model is implemented on a Xilinx reconfigurable Field-Programmable Gate Array (FPGA) using hardware description language: VHDL. This VHDL code is tested on the rapid prototyping platform named ML505 based on a Virtex5-LXT FPGA chip of Xilinx. Simulation results using some test and real images are presented to support the applicability of this approach compared to the standard blind deconvolution method that maximizes the likelihood using an iterative process. The comparison is based on performance evaluation using some recent image quality metrics.*

Keywords: *Radiological image restoration, ARMA, NN, PSO, BFO, FPGA.*

Received January 25, 2012; accepted March 28, 2013; published online February 26, 2014

1. Introduction

Radiology is the study of images of the human body. Although Radiology began with the use of X-Rays and large flat sheets of photographic films, the modern Radiologist now has a variety of tools for taking images of living patients. Many of these newer tools create an image with a computer Computed Tomography (CT) and some do not use any X-Rays, nor radiation of any kind, such as Magnetic Resonance Imaging (MRI) and UltraSound (US). Nuclear medicine studies involve making the patient temporarily radioactive, with a very small amount of an isotope. The images are obtained by looking for the small amount of radioactivity given off by the patient. Angiography is the name of a procedure that produces an image of blood vessels inside the body. Image in any form is never an exact representation of the object under observation because it is always corrupted by the imaging system itself. In radiology, there are several components tend to degrade the image, then limiting the resolution of such images. The image degraded sources in radiology are geometric unsharpness associated with lack of collimation in the beam, statistical fluctuation associated with low intensities or high background, motion unsharpness due to object motion during the exposure, and limitations in the imaging and processing systems. To overcome this problem, digital image restoration is a solution to reduce the blurring and noise effects on the image. Restoration techniques are different from image enhancement techniques. These techniques try to perform an inverse transformation of the observed degraded image to estimate the original scene. Because of this approach, image restoration techniques are oriented toward modeling the degradations, in order to apply an "inverse" technique. In practice, exact restoration of the original scene from the observed image data may be impossible, even with knowledge of the degrading system characteristics. This is due to the ill-posed nature of the image restoration problem, and the presence of observation noise. Authors of Saadi *at el.* [19, 20] did a comparative study between some state of the art approaches to select an image deconvolution method, and they got good results by introducing two swarm intelligence algorithms in the optimization step. The process of simultaneously estimating the degradation function and restoring an unknown image using partial or no information about the imaging system is known as blind image restoration [10]. Blind deconvolution

technique was another restoration method in image processing field. The objective of the blind image restoration is to reconstruct the original image from a degraded observation without the knowledge of either the true image or the degradation process. Authors of [24] used a novel rule base Fuzzy 2D Kalman Filter (RBFK) to restore images, which are heavy corrupted with the mix of Gaussian and impulse noises with preserving the details on the angiograph images, which is especially important in medical imaging techniques, they used blood vessel as a test image. Thavavel *et al.* [23] proposed a technique that incorporates genetic algorithm within wavelet denoising framework for threshold optimization. Applied of this technique on ultrasound and MR images of brain has shown a superior performance over the state-of-art wavelet-based denoising methods in terms of visual quality as well as quantitative metrics such as PSNR. Guo *et al.* [8] used iterative Constrained Least Squares algorithm for both blur identification and image restoration in blind restoration of images degraded by space-variant blurs. They extended the Expectation-Maximization (EM) algorithm and combine it with the region adaptive technique to handle the problem of identifying spatially variant blurs. They got good estimate with reduced artifacts in restored images in addition to blur identification. Liu [13] present a novel Decision-based Fuzzy Averaging (DFA) filter consisting of a Dempster-Shafer (D-S) noise detector and a two-pass noise filtering mechanism. Experimental results of this filter confirm its effectiveness both in suppressing impulsive noise as well as a mix Gaussian and impulsive noise. Blind deconvolution using ARMA parameter estimation methods involves modeling the true image as a two-dimensional (AR) process and the blurring function as a two-dimensional (MA) process. Based on these models, the resulting blurred image is represented as an (ARMA) process. Identifying the ARMA parameters allows us to identify both the true image and the degradation function [11]. Based on a novel 2D-ARMA parameter estimation, this paper propose a neural network model for fast blind image restoration. The proposed approach can overcome local minimization problem by using Particle Swarm Optimization (PSO) algorithm [7]. Comparison with iterative blind deconvolution based on results, shows that the proposed approach can obtain a better image estimate with a faster speed than the Blind Image Deconvolution (BID) algorithm. Maximum Likelihood (ML) [12, 16], Generalized Cross Validation (GCV) [18] and neural networks [6] are three popular methods that are employed to determine these parameters. The disadvantage of both ML and GCV lies in the higher computational cost which requires small AR and MA support sizes at the cost of diminished modeling effectiveness. However, in order to improve the modeling effectiveness, the support sizes can be increased in neural network based on ARMA models. Bustos *et al.* [5] summarize the main properties of the spatial ARMA models and describe some of the well-known methods used in image filtering based on estimation of spatial autoregressive models. They also, suggested a new filtering algorithm based on robust AR estimation. Shkvarko [21] addressed a new approach to the problem of improving the quality of remote-sensing images, by exploiting the idea of Neural Network (NN) based imaging system fusion. Because NN is easy to be trapped in local minima and converge too slow. The traditional training methods based on gradient searching technique are not effective and fast in determining accurate weights/biases of the neural network. In literature, various algorithms are proposed for training NN's, but most of them are derivative based and have some weaknesses such as converging to a local minima and time-consuming. Many global optimization methods have been used for this application such as Genetic Algorithm (GA) [3], Particle Swarm (PSO) [13], Bacterial Foraging (BFO) [26], and the Differential Evolution (DE) algorithm [25]. Qureshi [17], a GA is used to minimize the error performance surface in a NN based ARMA model with random Gaussian process is presented.

Traditionally image processing applications are implemented either in general-purpose Programmable Digital Signal Processors (DSPs) or built using Application Specific Integrated Circuit (ASIC) technology. Typically, Programmable DSP processors contain built-in Multiply/Accumulate units (MACs) and use a high level instruction set to implement image processing algorithms. Field-Programmable Gate Array (FPGA) has become a very cost-effective means of off-load computationally intensive digital signal processing algorithms to improve overall system performance. They provide flexibility of software while keeping intact the hardware performances [14].

Many works used hybrid implementations for image enhancement has been published [1]. In this paper, a hybrid implementation of two swarm algorithms is employed for training the neural network to attain the global error minimum and find optimized synaptic weights of a multilayer NN that will model the ARMA presenting both a better restored image and close blurring function. Simulation results and performance evaluation are provided. A statistical comparison is accomplished with BID method based on various recent image quality metrics. Blind deconvolution takes more time to restore an image than a direct algorithm due to the heavy computational load required to process frames, therefore it is not applicable to real-time images. In this work, to accelerate processing, the ARMA-NN model is implemented on a Xilinx reconfigurable FPGA chip using hardware description language: VHDL. This VHDL code is synthesized using ISE12.4 Xilinx

software, simulated on the ModelSim 6.5 software for time and space constraints and then, tested on the rapid prototyping platform named ML505 based on a Virtex5-LXT FPGA chip of Xilinx. Hardware schemes and simulation results are presented in the last section of this paper using ISE and ModelSim Softwares.

2. Image Degradation with ARMA Model

The true image is modeled as a two an image can be considered to be a sample function of random variables array. This characterization of an ensemble of images is useful in developing image processing techniques that are valid for an entire class and not just for an individual image. Two dimensional linear stochastic systems led to the modeling of blurred image as an ARMA process, where AR part determines the image model coefficients and MA part determines the blur function of the system [22]. Therefore, blind image deconvolution is transformed into an ARMA parameter estimation problem. Identifying the ARMA parameters allows us to identify the true image and the degradation function. This task can be achieved using neural networks [6] trained with classical optimization Algorithms that have the drawbacks of ill-convergence to local minima and sensitivity to initial conditions.
dimensional AR process represented by:

$$f(x,y) = \sum_{\substack{(l,m)\in R_a \\ (l,m)\in(0,0)}} a(l,m) f(x-l, y-m) + v(x,y) \quad (1)$$

f(x, y) is the true image, and *v(x, y)* is the modelling error which is a zero-mean homogeneous noise process which is statistically independent of *f(x, y)*. Using matrix-vector notation, equation 1 can be written as:

$$f = Af + v \quad (2)$$

For smooth and homogeneous true images as in photography, only three AR coefficients *{a(0, l), a(1,0), a(1,l)}* are sufficient to reasonably model the image. In most practical situations, the blurring function is of finite extent and its effect on the true image can be modeled as that of a two-dimensional FIR filter. The linear degradation model can be written:

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y) \quad (3)$$

Where the degradation function is *h(x, y)*, and *η(x, y)* is the additive noise of the imaging system assumed to be zero-mean Gaussian. The degraded image *g(x, y)* can be expressed as:

$$g(x,y) = \sum_{(l,m)\in R_h} h(l,m) f(x-l, y-m) + \eta(x,y) \quad (4)$$

Using matrix-vector notation, equation 4 becomes:

$$g = Hf + \eta \quad (5)$$

Rearranging Equation 2, substituting into Equation 5 and rearranging yields:

$$g = H(I - A)^{-1} v + \eta \quad (6)$$

Where *I* is the identity matrix A complete model for the blurred image using Equation 6 is given in Figure 1, where capital letters denote the Z-transforms of their lowercase counterparts. Therefore, the problem of blind deconvolution consists of estimating the AR parameters: *a (l, m)* for *(l, m)* $\in R_a$, and the MA parameters: *h (l, m)* for *(l, m)* $\in R_h$. Once the blurring function *h (1, m)* is determined, one of the classical linear image restoration methods can be used to estimate the true image:

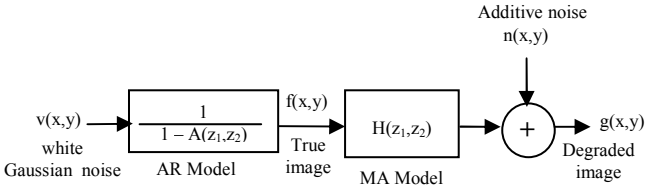


Figure 1. ARMA Model of the degraded image.

The practical difficulties with estimating {*a(l, m), h(l, m)*} using Equation 6 include high computational complexity with large support, instability of the estimation algorithms, and non-unique solutions. To overcome these problems, the following additional assumptions are commonly made on the blurring function by existing second-order statistics methods.

1. The blurring function is positive, and the mean value of the true image is preserved in the degradation process. That is: $\sum_{(l,m)\hat{I}R_h} h(l,m) = 1$. The use of these assumptions limits the number of possible ambiguous solutions to the problem.
2. The blurring function is symmetric and zero-phase. These assumptions are made for the stability and the uniqueness of solution of the estimation algorithms.
3. The blurring function has a known parametric form consisting of only a few parameters. Use of such models significantly lowers the computational complexity.

3. Swarm Intelligence

3.1. Particle Swarm Optimization

The PSO algorithm was first described in 1995 by James and Russell [9]. PSO is a stochastic, population-based evolutionary computer algorithm for problem solving. In a PSO system, a swarm of individuals (called particles) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself and the position of the best particle in its neighborhood. The performance or quality of each particle (i.e., how close the particle is from the global optimum) is measured

using a fitness function that varies depending on the optimization problem [24].

3.2. Bacterial Foraging Optimization

Foraging means finding, handling, and ingesting food. Animals that have successful foraging strategies are privileged since they obtain enough food to enable them to reproduce [15]. This has led scientists to model the activity of foraging as an optimization process. In [15], the author explains the biology and physics underlying the chemotactic (foraging) behavior of *E.coli* bacteria and gives a computer program that emulates the distributed optimization process represented by the activity of social bacterial foraging and applies that in adaptive controllers. The foraging strategy of *E.coli* bacteria present in human intestine can be explained by four processes namely: Chemotaxis, swarming, reproduction and Elimination/ Dispersal.

We want to find the minimum of $f(x)$, $x \in \Re^p$, where we do not have measurements or an analytical description of the gradient $\nabla f(x)$. Here we use ideas from bacterial foraging to solve this non gradient optimization problem. First, suppose that x is the position of a bacterium and $f(x)$ represents the combined effects of attractants and repellents from the environment, with, for example, $f(x)<0$, $f(x)=0$, and $f(x)>0$ representing that the bacterium at location x is in nutrient-rich, neutral, and noxious environments, respectively. Bacteria try to climb up the nutrient concentration (find lower and lower values of $f(x)$, avoid noxious substances, and search for ways out of neutral media). Chemotactic is a tumble followed by a tumble or a tumble followed by a run.

Algorithm 1: BFO Algorithm:

1.Initialization: We choose p, S, N_c , N_{re} , N_{ed} , P_{ed} and the C(i), i=1,2,...,S. for swarming, we choose also, parameters of the cell-to-cell attractant functions. Initial values for θ^i ,i=1,2,...S are also, chosen.

2.Elimination-dispersal loop: l=l+1

3. Reproduction loop: k=k+1

4. Chemotaxis loop: j=j+1

a) For i=1 to S take a chemotaxis step for bacterium I as follows.

b) Compute f(i,j,k,l) and let:

$$f(i,j,k,l) = f(i,j,k,l) + f_{cc}(\theta^i(j,k,l), P(j,k,l))$$

, we add on the cell-to-cell attractant effect to the nutrient concentration.

c) Let f_{last} = f(i,j,k,l) to save this value since we may find a better cost via a run.

d) Tumble: generate a random vector $\Delta(i) \in \Re^p$ with each element $\Delta_m(i)$, m=1,2,...P, a random number on [-1,1].

e) Move: let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$

this results in a step of size C(i) in the direction of the tumble for bacterium i.

f) Compute f(i,j+1,k,l), and then let:

$$f(i,j+1,k,l) = f(i,j+1,k,l) + f_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$$

g) Swim: let m=0 and While m<N_s put m=m+1,if f(i,j+1,k,l) < f_{last}

let f_{last} =f(i,j+1,k,l) and let:

$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ *and use this position to calculate the new cost value.*

Else, let m=N_s end while.

h) Go to the next bacterium.

5. if j<N_c then go to step 4.

6. Reproduction: For i=1,2,...,S. $f^i_{health} = \sum_{j=1}^{N_c+1} f(i,j,k,l)$ *(health of bacterium)*

i) . Sort bacteria and chemotactic parameters C(i) in order of ascending cost f_{health} (higher cost means lower health). The S/2 bacteria of the highest cost will die and the healthiest are placed at the same location as their parent.

7. if k<N_{re} go to step 4.

8. Elimination-dispersal: for i=1,2,...,S, with probability P_{ed} , eliminate and disperse each bacterium.

9. if l<Ned then go to step 1, otherwise end algorithm.

3.3. Hybrid Implementation: PSO-BFO

In the proposed hybrid approach, after undergoing a chemo-tactic step to perform a local search, each bacterium gets mutated by a PSO operator to accomplish a global search over the entire space. At this phase, the bacterium is stochastically attracted towards the globally best position found so, far in the entire population at current time and also, towards its previous heading direction. The PSO operator uses only the 'social' component and eliminates the 'cognitive' component as the local search in different regions of the search space [4]. BFO is changed by directing positions of bacteria and updating their velocities from the first chemotactic step using the power of PSO reaching the global solution. This hybridization improved the convergence speed and accuracy of solutions obtained by the classical BFO, however, what is requested is to attain a best approach to the original by finding the best solution, which is accomplished by this hybridization. In the BFO Algorithm, inside the Chemotaxis loop (step 4, point g), we introduce the PSO operator to update the global position of each bacterium, then determining fitness and subsequently we update both the global position and velocity of each bacterium before letting the bacteria swimming with the new speed on the way of the new updated direction:

g) We introduce PSO operator (for each chemotactic stepS):

- *Update the θ_{g_best} and f_{best}(i, j, k, l)*
- *Update position and velocity of the d-th coordinate of the i-th bacterium to the following rule:*

$$V_{id}^{new} = \omega . V_{id}^{new} + C_1 . f_1 . (\theta_{g\_best_d} - \theta_d^{old}(i, j+1, k))$$

$$\theta_d^{new}(i, j+1, k) = \theta_d^{old}(i, j+1, k) + V_{id}^{new}$$

4. Simulation Results

4.1. ARMA Neural Network Modeling

A NN is a parallel and distributed network of simple nonlinear processing units interconnected in a layered arrangement. Parallelism, modularity and dynamic adaptation are three computational characteristics typically associated with NN's. The Multi-Layer Perceptron (MLP) consists of various layers: An input and output ones between which lay one or several hidden ones whose outputs are not observable, Figure 2:

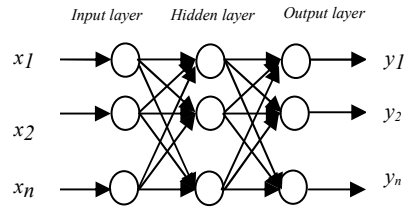


Figure 2. Schematic diagram of a MLP.

These layers are based upon some processing units (neurons) interconnected by means of feed-forward pondered links Figure 3:

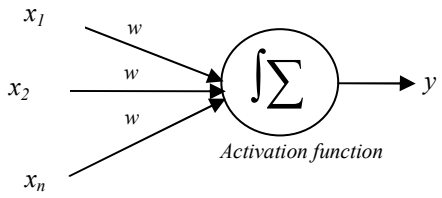


Figure 3. Processing unit in a MLP.

All these processing units carry out the same operation: The sum of their weighted inputs, Equation 7. Then they apply the result to a non-linear function named activation function and generally based upon the sigmoid function, Equation 8:

$$y_j = \left[\left(\sum_i w_{ij} . x_{ij}\right) - b_j\right] \quad (7)$$

$$f(x) = \frac{1}{\left(1 + e^{-x}\right)} \quad (8)$$

Where y_i is the output of the processing unit, w_{ij} are thesynaptic weight coefficients and b_j is the bias Back - Propagation (BP) has been widely adopted as a successful learning rule to find the appropriate values of NN weights. Using the hybrid implementation (PSO - BFO), each bacterium (or particle) position vector is defined by all connecting weights matrix w_{ij}. The fitness value of each bacterium is the value of the error function evaluated at this position. To achieve the same error goal both with back - propagation BP and Hybrid swarm algorithms, we find that the Hybrid PSO - BFO implementation requires less number of computations; which is an observed performance.

For our application, a multi-layer NN structure trained using a hybrid swarm implementation, to minimize the mean squares of errors function in the NN, is used to represent the ARMA model for identifying the blur function and restore the degraded image, simultaneously. The main difficulty in our approach is to learn correctly the perceptron because the learning sets are very large (about 100 examples), Figure 4. We developed a MLP with 3×3 input/output dimensions and two hidden layers, Figure 5. Also, the swarm optimization algorithm parameters must be chosen carefully. The input of the NN model is a white Gaussian noise and the output of the last layer is the observed degraded image. The output of the second layer is the estimated original image and the weights between the third and the output layers represent the blurring function.

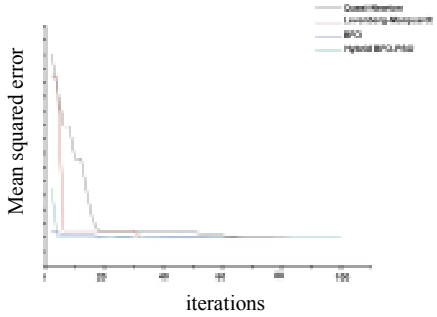


Figure 4. Comparison between different learning algorithms using MSE evolution.

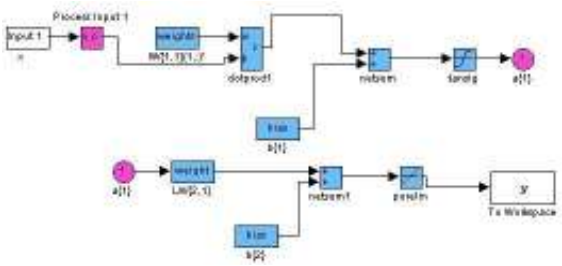


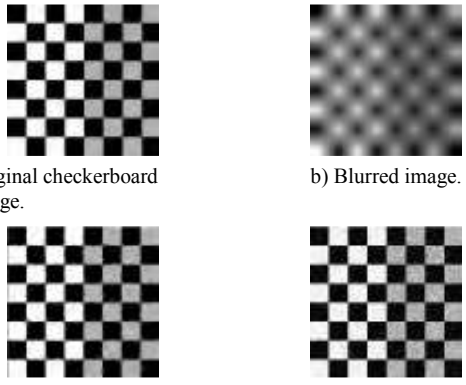
Figure 5. The resultant simulated neural network on matlab/ simulink.

In our simulation we use gray scale images with size of 256x256 pixels and 256 gray-levels. Some practical simulation results are given below. Before application to radiological images, we launch a step by step assessment procedure of our model using some test images restoration Figures 6 and 7 shows a comparison between the classical BID that maximizes the likelihood using an iterative process and restoration via the optimized ARMA - NN model. The reference images, used also, for comparison, are two text images, Figure 6 and one generated by the MATLAB function 'checkerboard', Figure 7. The last reference image contains all gray levels (from 0 to 255). The original image in Figure 7 - a is blurred by 5×5 Gaussian blur and Gaussian noise with 10dB was added to the blurred image of Figure 7 - b. Restoration results presented in Figures 7 - c and 7 - d, reveal that the proposed approach gives a better image estimate than the BID method.

a) Original text image. b) Blurred image.

c) Restored image with iterative BID algorithm. d) Restored with the ARMA-NN model.

Figure 6. Comparison between the classical BID and our ARMA-NN model using a text image.



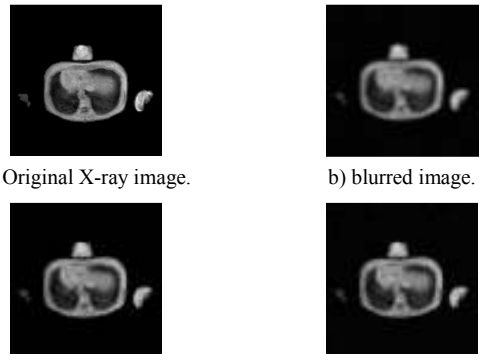
a) Original checkerboard image. b) Blurred image.

c) Restored image withiterative blind deconvolution algorithm. d) Restored with the ARMA-NN model.

Figure 7. Comparison between the classical BID and our ARMA-NN model using checkerboard image.

4.2. Application to Radiological Images

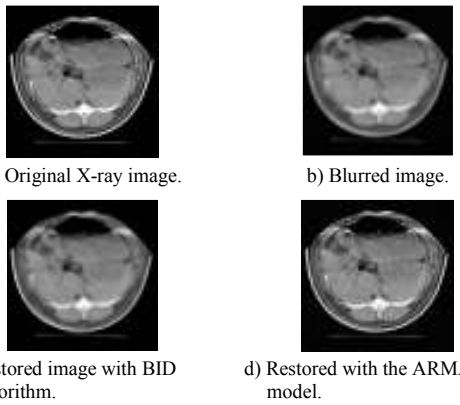
Medical radiology offers many good techniques helping doctors in their diseases diagnosis work and also, it is widely used in medical research. Radiological images used here are gathered from radiological databases. Five images of different types are selected for experimentation to validate the proposed model, Figures 8 - 12.



a) Original X-ray image. b) blurred image.

c) Restored image with BID Algorithm. d) Restored with the ARMA-NN model.

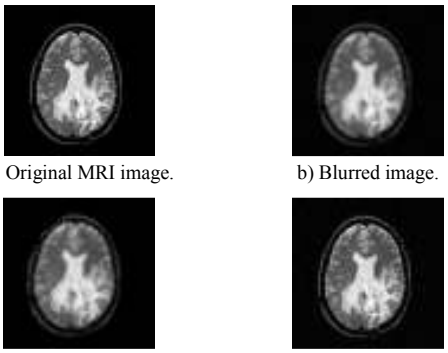
Figure 8. Comparison between the classical BID and our ARMA-NN model using an *x-ray* image.



a) Original X-ray image. b) Blurred image.

c) Restored image with BID algorithm. d) Restored with the ARMA-NN model.

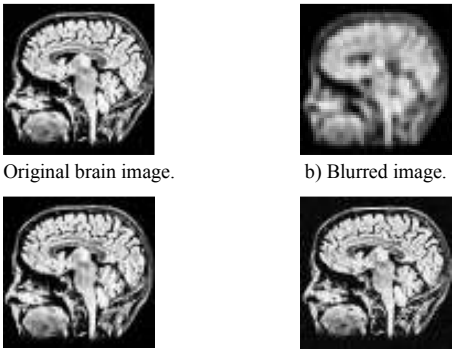
Figure 9. Comparison between the classical BID and our ARMA-NN model using an x-ray image.



a) Original MRI image. b) Blurred image.

c) Rrestored image with BID Algorithm. d) Restored with the ARMA-NN model.

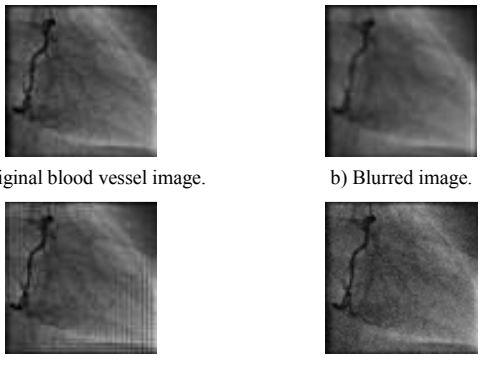
Figure 10. Comparison between the classical BID and our ARMA-NN model using a MRI image.



a) Original brain image. b) Blurred image.

c) Restored image with BID algorithm. d) Restored with the ARMA-NN model.

Figure 11. Comparison between the classical BID and our ARMA-NN model using a Brain image.



a) Original blood vessel image. b) Blurred image.

c) Restored image with BID algorithm. d) Restored with the ARMA-NN model.

Figure 12. Comparison between the classical BID and our ARMA-NN model using a blood vessel image.

Some image quality measures are calculated for all restored images with reference to their original images. To extend tests, a set of seven graphs are constructed below, Figures 13-18. The implemented image quality measures are defined using the following expressions:

1. *Mean Square Error (MSE)*:

$$MSE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right)^2$$

2. *Peak Signal to Noise Ratio (PSNR in dB)*:

$$PSNR = 10log \frac{\left(2^n - 1\right)^2}{MSE} = 10log \frac{255^2}{MSE}$$

3. *Normalized Cross-Correlation (NCC)*:

$$NK = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k} \cdot x'_{j,k}}{\sum_{j=1}^{M} \sum_{k=1}^{N} x^2_{j,k}}$$

4. *Average Difference (AD)*:

$$AD = \sum_{j=1}^{M} \sum_{k=1}^{N} \frac{\left(x_{j,k} - x'_{j,k} \right)}{MN}$$

5. *Structural Content (SC)*:

$$SC = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} x^2_{j,k}}{\sum_{j=1}^{M} \sum_{k=1}^{N} x'^2_{j,k}}$$

6. *Maximum Difference (MD)*:

$$MD = Max \left(\left| x_{j,k} - x'_{j,k} \right| \right)$$

7. *Normalized Absolute Error (NAE)*:

$$NAE = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} - x'_{j,k} \right|}{\sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} \right|}$$

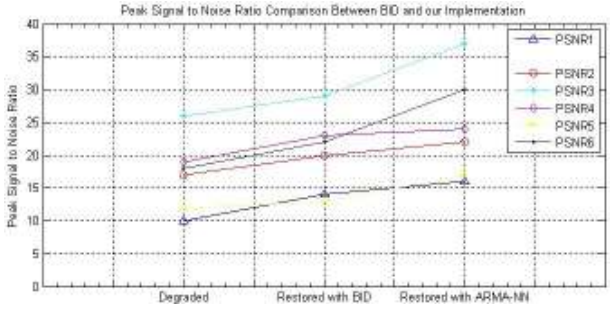


Figure 13. PSNR evolution: comparison between classical BID and ARMA-NN model.

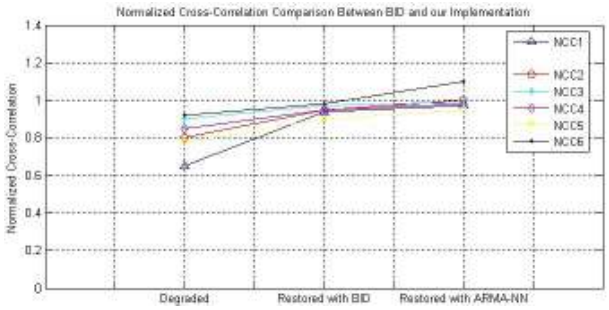


Figure 14. The normalized cross correlation: comparison between classical BID and ARMA-NN model.

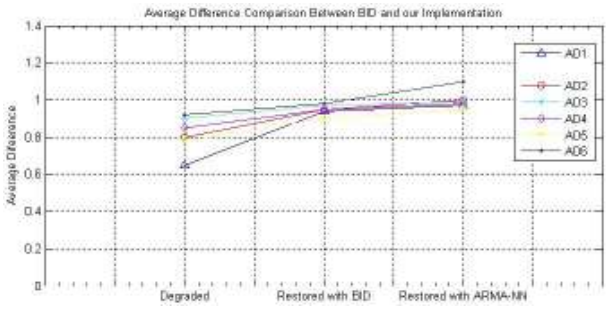


Figure 15. The average difference: comparison between classical BID and ARMA-NN model.

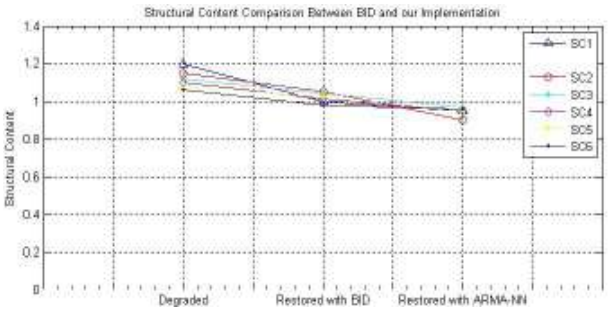


Figure 16. The structural content: comparison between classical BID and ARMA-NN model.

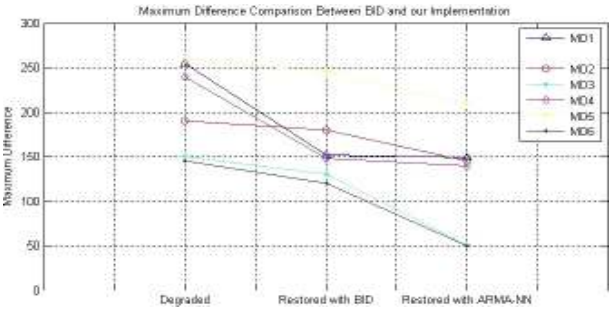


Figure 17. The maximum difference: comparison between classical BID and ARMA-NN model.

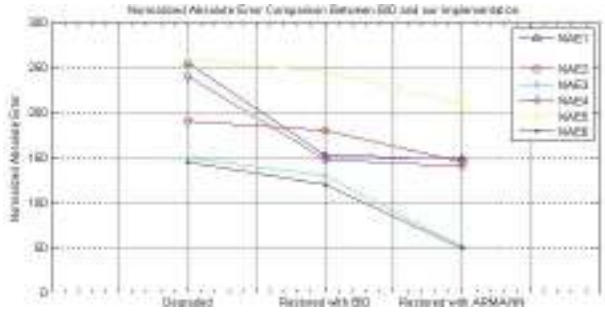


Figure 18. The normalized absolute error: comparison between classical BID and ARMA-NN model.

From images visual inspection with numerical evaluation from Figures 13-18, we can judge that the PSNR is a little enhanced using the proposed model compared to standard BID. The performance of this approach for restoring radiological images degraded by Gaussian and motion blur and an additive noise is revealed from experiments performed on some database images. Although, the performance is demonstrated for such images, the proposed approach can be used to restore others images degraded with the same blurring function. In the future, we will try to

implement models for other known blurring functions encountered in radiological images to further improve the performance.

5. FPGA-Based Implementation of the ARMA-NN Model

The advances in high density and high performance FPGAs offer an attractive alternative for realizing complex image processing applications. This technology offers an attractive combination of low cost, high performance combined with an apparent flexibility, while maintaining the advantages of custom functionality. Furthermore, FPGAs can be reconfigured repeatedly and their configurations upgraded to match any changes in the design, the classical electronics lacks this property [2]. The description of a neuron in VHDL language allows, during compilation, specifying some characteristics such as the number of inputs and the data length in bits in a generic way, and possibly changes the type of arithmetic operators (adder and multiplier). The neuron performs primarily the product of input data X, stored in the RAM with the corresponding synaptic weights W which are in the ROM and then adds them to a bias term b. Results of these transactions are subject to a unit that approximates an activation function tansig, Figure 19.

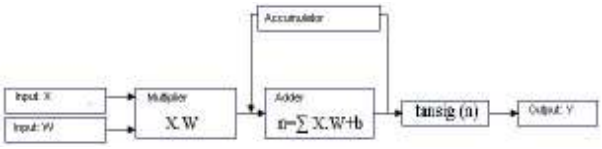


Figure 19. Neuron structure.

The following Equations 9 and 10 summarize the operations accomplished in each neuron:

$$n = \sum X.W + b \tag{9}$$

$$Y = tansig(n) \tag{10}$$

Where X is an input vector, W is the weights matrix and b is the bias vector.

The computation in neural networks (multiplication and addition) requires the use of signed real numbers. To represent fractional numbers it is necessary to define the position of the decimal point, for this, two methods are possible:

- The fixed-point representation.
- The floating point representation.

The second is more dynamic than the first; it allows the encoding of a larger number of real values for the same number of bits. In addition, it does not require overflow management which is essential in the fixed-point representation. On the other hand, the fixed-point representation has undeniable assets, making it the preferred choice in various applications. In our approach we have chosen a fixed-point representation of 18 bits (5 bits for integer part and 12 bits for fractional part), figure 20, this facilitates the Multiplier-Accumulator (MAC) realization which is the heart of the NN.

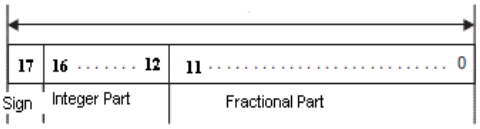


Figure 20. Fixed point representation of 18 bits.

It is enough with this word length to represent weights and biases with high precision. Several types of adders can be used to perform the weighted sum of states: combinatorial serial, dynamic, carry look ahead, manchester, carry select, Wallace tree...etc. Similarly, there are several ways to perform the multiplication, the most traditional are: In serial, serial/parallel and fully parallel. The MAC in Figure 21 is realized by a multiplier associated to an adder looped on itself in order to obtain an accumulation.

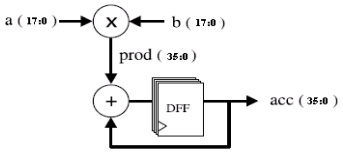


Figure 21. MAC Structure.

In our implementation, we need 45 (9×5) MAC's in the hidden layer and 5(5×1) in the output layer if we want a completely parallel NN. However, we significantly reduced the space by adopting this number of MAC's for all neurons (for the entire network) and multiplication is done in turn through a multiplexer. This approach provides us with a partially parallel NN less dense but slower.

5.1. Activation Function (Tansig) Approximation

Digital simulation of the activation function, obtained from the basic Equation 11 requires a relatively large computation time compared to the time required for the MAC:

$$F(x) = (1 - e^{-2x}) / (1 + e^{-2x}) \tag{11}$$

Several possibilities exist to materialize this function (hyperbolic tangent). The lookup table places in memory (ROM) the digitalized values of the activation function, for each value of x correspond a value for y. The x values are used to address this memory. The advantage of this method is not only ability to simulate several types of activation functions, but also, it is simple to implement. The accuracy depends on the size of the memory. The problem comes from both the area occupied in the memory and its access time. Thus, the direct implementation for non - linear sigmoid transfer

functions is very expensive. To simplify realization, we used the method of linear piecewise approximation that divides the activation function into intervals and models each interval by a line. It consumes less memory space and has better precision. It is useful to test the model by introducing arbitrary values to the input and compare the output with the exact value. The simulation result is shown in Figure 22. In Figure 23, we present a comparison between MATLAB plotted tansig and its estimation using VHDL language

Figure 22. Simulation results.

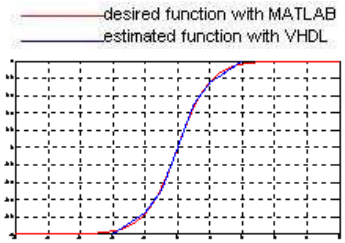


Figure 23. Comparison between the desired function tansig (red) and its approximation (blue) with VHDL.

5.2. Elementary Processor Implementation

The entire network consists of elementary processors (the basic component of the NN), each of which contains a MAC circuit connected to the piecewise circuit that approximates the activation function. Neuron inputs and output have the same length (18 bits). The MAC output is 36 bits in length and will be reduced at the piecewise circuit. Thus, the outputs of the first layer will be compatible with the second layer inputs. The neuron module is made by combing the MAC with the linear piecewise function (approximating the tansig function).

5.3. Implementation of the Neural Network

There are two different kinds of neurons: one is used in the first level and the other is used at second level. Those of the first - level have 9 inputs; the others have a number of inputs equals the number of neurons in the first level. A step construction of these circuits allows the pipeline calculations. The hidden layer operates at clock rising edge, while the output layer works at the clock descending front. The design placement on a Virtex5 LXT FPGA chip of Xilinx, is shown in Figure 24. This chip was largely sufficient to implement the whole ARMA - NN model on the space of this chip as revealed in Figure 25 - a. The used platform for this implementation is given in Figure 25 - b.

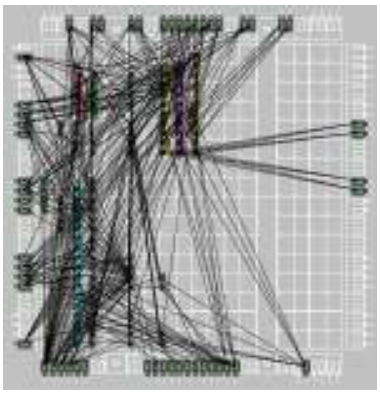
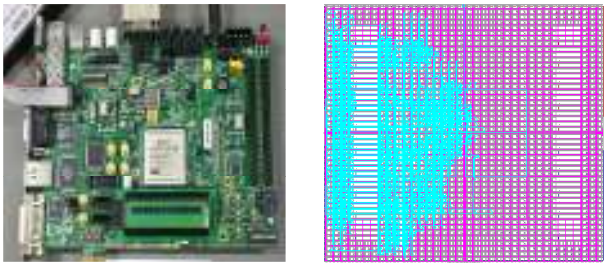


Figure 24. Design placement: The developed architecture implemented in FPGA.



a) Floor planning of the FPGA implementation (the used space).

b) The FPGA Hardware used.

Figure 25. The whole ARMA-NN model implemented on the chip space.

6. Conclusions

For better diagnostics, restoring radiological images degraded during acquisition and processing becomes very essential. Getting a reliable, efficient and authentic system for this purpose is a challenge in image processing. In this work, we could reach approximately this goal, compared to blind image deconvolution BID, by combining: the parallel computing scheme of NN and real time reconfigurable FPGA hardware to identify ARMA model parameters. In addition, we reduced the NN configuration by enhancing its learning using the powerful optimization swarm intelligence Algorithms: synergy of PSO and BFO. The result is obtaining well restored estimate of the desired image, simultaneously with identification of the blurring function. Application to radiological images reveals the usefulness of this implementation in real life world.

References

[1] Abu - Sadah Y., Al-Najdawi N., and Tedmori S., "Exploiting Hybrid Methods for Enhancing Digital X-Ray Images," *the International Arab Journal of Information Technology*, vol. 10, no. 1, pp. 28 - 35, 2013.

[2] Alotaibi K., "A High Level Hardware Description Environment for FPGA - Based Image Processing Applications," *PhD Thesis*, The Queen's University of Belfast, 1999.

[3] Barreto A., Barbosa H., and Ebecken N., "Growing Compact RBF Networks Using a Genetic Algorithm," *in Proceedings of 7th Brazilian Symposium on Neural Networks*, Recife, Brazil, pp. 61 - 66, 2002.

[4] Biswas A., *Synergy of PSO and Bacterial Foraging Optimization: A Comparative Study on Numerical Benchmarks*, *Innovations in Hybrid Intelligent Systems*, Springer - Verlag, Berlin Heidelberg, 2007.

[5] Bustos O., Ojeda S., and Vallejos R., "Spatial ARMA Models and its Applications to Image Filtering," *Brazilian Journal of Probability and Statistics*, vol. 23, no. 2, pp. 141 - 165. 2009.

[6] Cho M. and Don H., "Blur Identification and Image Restoration Using a Multilayer Neural Network," *in Proceedings of IEEE International Joint Conference on Neural Networks*, Vancouver, British Columbia, vol. 3, pp. 2558 - 2563, 1991.

[7] Farzi S., "Training of Fuzzy Neural Networks via Quantum-Behaved Particle Swarm Optimization and Rival Penalized Competitive Learning," *the International Arab Journal of Information Technology*, vol. 9, no. 4, pp. 306 - 313, 2012.

[8] Guo Y., Lee H., and Teo C., "Blind Restoration of Images Degraded by Space-Variant Blurs Using Iterative Algorithms for Both Blur Identification and Image Restoration," *Image and Vision Computing Journal*, vol. 15, no. 5, pp. 399 - 410, 1997.

[9] Kennedy J. and Eberhart R, "Particle Swarm Optimization," *in Proceedings of IEEE International Conference on Neural Networks*, Australia, Perth, Western Australia, vol. 4, pp. 1942 - 1948, 1995.

[10] Kundur D. and Hatzinakos D., "Blind Deconvolution Scheme for Image Restoration Using Recursive Filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 2, pp. 375 - 390, 1998.

[11] Lagendijk R., Biemond J., and Boekee D., "Identification and Restoration of Noisy Blurred Images Using the Expectation-Maximization Algorithm," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 38, no. 4, pp. 1180 - 1191, 1998.

[12] Lin T., "Decision-Based Fuzzy Image Restoration for Noise Reduction Based on Evidence Theory," *Expert Systems with Applications Journal*, vol. 38, no. 7, pp. 8303 - 8310, 2011.

[13] Liu Y., "Training Radial Basis Function Networks with Particle Swarms," *in Proceedings of International Symposium on Neural Networks*, Dalian, China vol. 3173, pp. 317 - 322, 2004.

[14] Meyer-Baese U., *Digital Signal Processing with Field Programmable Gate Arrays*, Springer Berlin Heidelberg, Berlin, Germany, 2007.

[15] Passino K., "Biomimicry of Bacterial Foraging, for Distributed Optimization and Control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52 - 67, 2002.

[16] Qian Y., Miao H., Shao J., and Jin W., "Real-Time Restoration of an Image Blurred by Arbitrary Motion Using an Optoelectronic Hybrid Joint Transform Correlator," *Journal of Optics*, vol. 19, no. 11, pp. 10762 - 10768, 2011.

[17] Qureshi I., "Genetic Algorithms Based Artificial Neural Networks for Blur Identification and Restoration of Degraded Images," *Pakistan Journal of Information and Technology*, vol. 2, no. 1, pp. 21 - 24, 2003.

[18] Reeves S. and Mersereau R., "Blur Identification by the Method of Generalized Cross-Validation" *in Proceedings of IEEE Transaction on Image Processing*, vol. 1, no. 3, pp. 301 - 311, 1992.

[19] Saadi S., Kouzou A., Guessoum A., and Bettayeb M., "A Comparative Study to Select an Image Deconvolution Method," *in Proceedings of the 7th IEEE International Multi-Conference on Systems*, *Signals and Devices*, Amman, Jordan, pp. 1 - 6, 2010.

[20] Saadi S., Kouzou A., Guessoum A., and Bettayeb M., "Bacterial Foraging Algorithm for Neutron Radiography Image Quality Improvement," *in Proceedings of the 7th IEEE International Multi-Conference on Systems*, Amman, Jordan, pp. 1 - 6, 2010.

[21] Shkvarko Y., "System Fusion in Passive Sensing Using a Modified Hopfield Network," *Journal of the Franklin Institute*, vol. 338, no. 4, pp. 405 - 427, 2001.

[22] Tekalp A. and Sezan M., "Identification of Image and Blur Parameters for the Restoration of Non-Causal Blurs," *in Proceedings of IEEE Transaction Acoustics, Speech, Signal Processing*, vol. 34, no. 4, pp. 963 - 972, 1986.

[23] Thavavel V., Basha J., and Murugesan R., "A Novel Intelligent Wavelet Domain Noise Filtration Technique: Application to Medical Images," *Expert Systems with Applications*, 2010.

[24] Toprak A. and Guler I., "Angiograph image Restoration with the Use of Rule Base Fuzzy 2D Kalman Filter," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1752 - 1761, 2007.

[25] Yu B. and He X., "Training Radial Basis Function Networks with Differential Evolution," *in Proceedings of IEEE International Conference on Granular Computing*, Atlanta, USA, pp. 369 - 372, 2006.

[26] Zhang Y., Wu L., and Wang S., "Bacterial Foraging Optimization Based Neural Network for Short-term Load Forecasting," *Journal of Computational Information Systems*, vol. 6, no. 7, pp. 2099 - 2105, 2010.

Slami Saadi received his BSc degree in Electrical and Electronics Engineering from INELEC (1993), University of Boumerdes (Algeria) and MSc degree in Signal Processing from the University of Blida (2005), and PhD degree both at the University of Blida (Ageria) and University of Sharjah (UAE) (2012). His interests include artificial intelligence applied to signal and image processing, hardware and software reconfigurable implementation of digital signal processing tools, embedded electronics…etc. He is, currently in the Faculty of Sciences and Technology, university of Djelfa, Algeria, as an assistant Professor in Signal Processing and Computer Science. Mr. S. Saadi has published many journal and conference articles and technical reports.

Abderrezak Guessoum (S'78-M'84) graduated from the Ecole Nationale Polytechnique of Algiers in 1976. He received the MSc and PhD degrees from the Georgia Institute of Technology, Atlanta, in 1979 and 1984, respectively. He also, received the MSc degree in applied mathematics from Georgia Tech in 1983. During the academic year 1984-1985 he was an Assistant Professor of Computer Science at Jackson State University, Jackson,USA. Prof. A. Guessoum is currently teaching signal processing at the University of Blida (Algeria) and leading the Laboratory of Signal and Image processing (LATSI) at Blida University. His interests include signal/Speech processing, intelligent Computing, embedded electronics.

Maamar Bettayeb received his BSc (1976), MSc (1978) and PhD (1981) degrees in Electrical Engineering from University of Southern California, Los Angeles, California, USA. He is currently a Professor at Sharjah University, UAE and Coordinator, of Intelligent Systems Research Group. He is a senior Member, IEEE. Prof. M. Bettayeb interests are Large Scale Systems Analysis and Design, H∞ Control and Filtering, Identification, Estimation & Filtering, Higher Order Statistics and Applications, Wavelets and Applications.

Kamel Abdelhafidi received his BSc degree in Electrical Engineering from Laghouat University (Algeria) in 1993, and MSc degree in Signal Processing from the ENSET (Oran, Algeria) in 2009. He is currently working towards a PhD Degree. His interests include VLSI circuits development. He is, currently, with the laboratory Metallic and Semiconducting Materials and assistant Professor in electrical engineering in Biskra University.