FPGA Implementation of NLMS Algorithm for Receiver in wireless communication system

K. R. Rekha *, Dr B. S. Nagabushan ** And Dr K.R..Nataraj ***

*E&C Dept, M. G. R. University, Chennai, India
** Sanlab Technologies, Bangalore, 560038, India
*** SJB Institute of Technology, Bangalore, 560060, India

Abstract
This paper proposes a verilog implementation of a normalised Least Mean Square (NLMS) adaptive algorithm. The envisaged application in the wireless communication identification system. The good convergence of NLMS algorithm has made us to choose it. It also has good stability. Adaptive filtering constitutes one of the core technologies in digital signal processing and finds numerous application areas in science as well as in industry. In this paper NLMS algorithm is used to reduce the error at the output of the receiver in wire less communication system. A verilog implementation is developed for a 5th order NLMS adaptive filter. As compared conventional LMS it has been proven that NLMS Algorithm has good behaviour. Model Sim simulations results altogether with plots obtained in Mat lab prove the same.

Keywords:
Adaptive filters, Digital signal processing, FPGA NLMS, LMS.

1. Introduction
Even though many interesting adaptive algorithms are present around us. The applications with limited precision and processing power, the Normalized Least-Mean-Square (NLMS) algorithm [3] and some versions of it (e.g., frequency-domain or sub-band versions [1]) are usually used. The step-size parameter will control the algorithm, in terms of convergence rate, maladjustment, and stability. Within the stability conditions, the choice of this parameter reflects a trade-off between fast convergence rate and good tracking ability on the one hand and low maladjustment on the other hand. To meet these conflicting requirements, the step size needs to be controlled. This algorithm gives good performance.

Reducing an error in receiver system has been a central issue in wireless communication networks and teleconferencing etc. reducing an error in the procedure of specifying the receiver model in terms of the available experimental evidence, that is, a set of measurements of the input output desired response signals and an appropriately error that is optimized with respect to Receiver parameters. Adaptive identification refers to a particular procedure where we learn more about the model as each new pair of measurements is received and we update the knowledge to incorporate the newly received information.

2. Need for NLMS
In this paper, we will be describing the more stabled adaptive algorithm Normalized Least Mean Square (NLMS) algorithm. The paramount reasons for this decision are as follows:

1. The LMS adaptive FIR filter is the most popular adaptive estimation technique and is likely to remain so in the foreseeable future [9]. Despite speculations that the LMS algorithm is losing its established status as the workhorse for the design of linear adaptive systems, there are still numerous ongoing researches and state-of-the-art advances in this algorithm [7]. A forthcoming new book titled —Least-Mean-Square Adaptive Filters, edited by Bernard Widrow (originator of LMS) and Simon Haykin, is a good representative of the devoted interests in this adaptive algorithm shown by researchers around the world.

2. The LMS algorithm can be easily modified to a normalized step-size version known as the Normalized LMS (NLMS) algorithm. NLMS, not only provides a potentially faster adaptive algorithm, but also guarantees a more stable convergence in response to variations of input signal power.

2.1 Following this approach, the main objectives of this paper are:

1. Study of LMS /NLMS adaptive algorithms.
2. Implement an NLMS-based adaptive algorithm for receiver system in wireless communication to show its good convergence rate and reduction in error.

3. Design of modified LMS algorithm

In this section the derivation of modified LMS has been given and it is named as normalised least mean square algorithm (NLMS).

In many adaptive filter algorithms Normalized least mean square algorithm (NLMS) is also derived from conventional LMS algorithm. The objective of the alternative LMS-based algorithms is either to reduce computational complexity or convergence time. The normalized LMS (NLMS), algorithm utilizes a variable convergence factor that minimizes the instantaneous error. Such a convergence factor usually reduces the convergence time but increases the misadjustment. In order to improve the convergence rate the updating equation of the conventional LMS algorithm can be employed variable convergence factor. It is derived as below.

\[ \mu_2 \sigma^2_x \] directly affects the convergence rate and stability of the LMS adaptive filter. As the name may imply, the NLMS algorithm is an effective approach to overcome this dependence, particularly when the variation of input signal power is large, by normalizing the update step-size with an estimate of the input signal variance, \( \sigma^2_x(n) \) [10]. In practice, the correction term applied to the estimated tap-weight vector \( w(n) \) at the n-th iteration is ‘normalized’ with respect to the squared Euclidean norm of the tap input \( x(n) \) at the (n-1)-th iteration [8],

\[
w(n+1) = w(n) + \frac{\mu}{\|x(n)\|^2} e(n)x(n)
\]

3.1

Apparently, the convergence rate of the NLMS algorithm is directly proportional to the NLMS adaptation constant \( \mu \), i.e. the NLMS algorithm is independent of the input signal power. Theoretically, by choosing \( \mu \) so as to optimize the convergence rates of the algorithms, the NLMS algorithm converges more quickly than the LMS algorithm [10]. Indeed as reported in [11], by taking into account the variation of signal level at the filter input and selecting a normalized correction term, we get a stable as well as a potentially faster converging adaptation algorithm for both uncorrelated and correlated input signal. It has also been stated that the NLMS is convergent in the mean square if the adaptation constant \( \mu \) (note that it is no longer called the step size) satisfies the following condition [12]:

\[ 0 < \mu < 2 \quad \ldots \ldots \ldots \ldots \ldots 3.2 \]

Despite this particular edge that NLMS exhibits, it does have a slight problem of its own. Consider the case when the input vector \( x(n) \) is small. Instability may occur since we are trying to perform numerical division by a small value of the Euclidean Norm \( ||x(n)||^2 \).

However, this can be easily overcome [8] by appending a positive constant to the denominator in

\[ W(n+1) = w(n) + \frac{\mu}{c+\|x(n)\|^2} e(n)x(n) \]

3.3

where \( c + \|x(n)\|^2 \) is the normalization factor. With this, we obtain a more robust and reliable implementation of the NLMS algorithm. In summary, we can write the LMS algorithm for every search iteration, in the form of three operations:

1. Initial Condition: \( \mu < 2 \)
2. Filter Output: \( y(n) = w(n) x'(n) \)
3. Error Estimation: \( e(n) = d(n) - y(n) \)
4. Tap-weight adaptation:

\[
w(n + 1) = w(n) + \frac{\mu}{c+\|x(n)\|^2} e(n)x(n)
\]

4. DETAILED DESIGN AND IMPLEMENTATION

The below block diagram (figure 4.1) shows the inputs and outputs of the NLMS algorithm. It has four inputs and two outputs. Inputs are \( x_\text{in} \) (input data to adaptive filter), \( d_\text{in} \) (desired input), \( \text{clk}(\text{clock}) \) and \( \text{adpt}\_\text{enable} \) (input bit used for ………). Outputs are \( \text{error}\_\text{out} \) (difference between output of filter(y-out) and desired input (d_in)) and final output (…).

The inputs \( x_\text{in} \) and \( d_\text{in} \) and outputs \( \text{error}\_\text{out} \) and final output are 8bit data. \( \text{clk} \) and \( \text{adpt}\_\text{enable} \) are single bit data.

The NLMS block consists of two shift registers, calculator, adder and a multiplexer. The inner structure of the NLMS is as shown in figure 4.2 below
The above block shows the inner structure of arrangements of the internal blocks of NLMS. Here the input to the adaptive filter $x_{in}$ is given to the 20-bit shift register and desired input $d_{in}$ is given to the 21-bit shift register. The two outputs of the 21-bit shift register are given to the calculator block which generates the output $y_{out}$. The output of the 20-bit shift register is subtracted with the output of calculator $y_{out}$. The subtracted output is given to the multiplexer where the enable bit is used as a select line. The other input of the multiplexer is “00000000”. The calculator block is used for calculation of $y_{out}$ at several iterations. Here we used for five iterations. We can call the calculator block as the core filter block because it generates the filter output $y_{out}$. When we considering the calculator block as main block which is used for several iterations, it consist of...
five inputs and four outputs. In which only clock is the single bit input, all others are 8-bit data. The calculator block is as shown in figure 4.3.

![Figure 4.3: Block Diagram of Calculator](image_url)

The different blocks inside the core filter are adder, multiplier, shift register, saturation, scaling, and truncation. The single calculator block uses some signals which are used for the calculation. The signals are shiftx(32), shiftx(32), shiftue(24), shifty(16), coeff16(16), coeff8(8), xnin_ue(16), xnin_ue_scaled(16), new_coeff(16), delayed_new_coeff(16), y_out16(16) and y_out8(8). The numbers in bracket refers to the bit size of that signal. These signals are used during the different arithmetic operation of the input data at the calculator block.

The first block of the calculator is multiplier which multiplies the 8-bit data x_N_in and ue_in, the output signal xnin_ue is 16-bit data. The second stage is scaling. In scaling stage the output may be upgrade or degrade of the input signal. Here we always degrade the output of the multiplier stage. The output is xnin_ue_scaled which is also 16-bit, which is added to the 16-bit co-efficient signal coeff16 at next stage. The signal new_coeff is the output of the adder stage which is delayed by using shift register in next step. The delayed new coefficient which is the output of shift block is 16-bit data which is used in calculation of the next coefficient for core filter. The output checked for its limit in saturation stage. In saturation stage care is taken about the data not to exceed the limit of the coefficient. The saturated data is truncated to 8-bit in next stage.

The next stage is 8-bit multiplier which multiplies the 8-bit truncated coefficient with 8MSB bits of shiftx signal which gives 16-bit product. These 16 bit output i.e y_out16 is truncated to 8-bit in next stage. The 8-bit output is the final output of the core filter after 5th iterations.

![Figure 4.4: Inner Structure of the NLMS Core Filter](image_url)
Figure 4.5 Inner Structure of the Calculator

Figure 4.6: Internal Blocks of the Core Filter and its data flow
5. SYNTHESIS AND SIMULATION RESULTS

The implemented VHDL program for NLMS adaptive algorithm is now synthesised using Xilinx 10.1. Modelsim is used to study the waveforms of each stage. The VHDL programs are synthesised separately for each block. The synthesis result observed for core filter of the NLMS block and the total NLMS block separately are shown below.

Figure 5.1 Top Level Structure for NLMS Algorithm.

Figure 5.2 1st levels Internal Structure
Figure 5.3 Internal Structure of Core filter

Figure 5.4 Internal Structure for Unit Calculators
Figure 5.5: Outed DesignFPGA R for NLMS

Figure 5.6: Power Analysis of NLMS using Xilinx X Power Analyzer
Figure 5.7 Top level Simulation Results for NLMS

Figure 5.8 Tap 1 Simulation Result
Figure 5.9 Simulation output for 2K Hz

Figure 5.10 Error Curve
Figure 5.11 System Output

Figure 5.12 Comparison of the Actual Weights and the Estimated Weights
Figure 5.13 Simulation results for 20K

Figure 5.14 System Output
Figure 5.15 Comparison of the Actual Weights and the Estimated Weights

Figure 5.16 Design Summary for Normalised Least Mean Square Algorithm
REFERENCES

[22] Douglas, S. C., Zhu, Q. and Smith, K. F., A Pipelined LMS Adaptive FIR Filter Architecture without...


[40] S. Haykin, Adaptive Filter Theory, Prentice-Hall, third edition,

**K. R. Rekha** obtained her ME degree from Bangalore University, India in 2000. She is working as a Asst Professor in the Department of Electronics and Communication in SJB Institute of Technology, Bangalore. Presently, she is pursuing her Ph. D. degree in Dr MGR University, Chennai. Her research interests include Wireless communication, FPGA implementation, and Microcontroller and Embedded systems design. She is a member of MIE, MISTE and IETE

**Dr B. S. Nagabushan** obtained his M. Tech. and Ph. D. degrees from Mysore University and Indian Institute of Science, Bangalore respectively. He has wide academic as well as software industrial experience for over 25 years. He has worked as Professor in various engineering colleges as well as a consultant to software industries like OMED (Software), Japan, BFL, CG-Smith Software Pvt. Ltd, KPT Cum Info systems (Bangalore), Pvt. Limited, San Lab Technologies etc. His research interests include Wireless Communication, Neural Network, and Fuzzy Logic and Embedded systems. He is the recipient of NRDC Independence Day award for the year 1992, Best Project Execution award for the year 2000 from M/s BMC

**Dr K. R. Nataraj** obtained his ME degree from Bangalore University, India in 2000. He worked as Professor and Head of the Department during 2000-2008 and currently he is the Post Graduate Coordinator in the Department of Electronics and Communication in SJB Institute of Technology, Bangalore. Presently, his research interests include Wireless communication, FPGA implementation, and Microcontroller and Embedded systems design. He is a member of MIE, MISTE, IETE and IEEE.