

# FPGA Implementation of LMS Algorithm Used in Adaptive Equalizer

Kaliprasanna Swain<sup>1</sup>  
[kaleep.swain@gmail.com](mailto:kaleep.swain@gmail.com)

Manoj Kumar Sahoo<sup>2</sup>  
[mksahoo@hotmail.com](mailto:mksahoo@hotmail.com)

Gandhi Institute for  
Technological Advancement  
(GITA) /ECE, Bhubaneswar,  
Odisha, India

**Abstract-** Least mean squares (LMS) algorithms are used in adaptive filters to find the filter coefficients that relate to producing the least mean squares of the error signal which is the difference between the desired and the actual signal. It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. The gradient descent method finds a minimum, by taking steps in the direction negative of the gradient, by adjusting the filter coefficients to minimize the error. The aim of this paper is to implement LMS algorithm in FPGA in wireless communication system. The implementation results shows the error minimize technique. It is tested in hardware using FPGA kit.

**Index Terms**— LMS algorithm, FPGA, Communication, Xilinx ISE and DSP tool kit, System Generator, Co-HW Simulation.

## I. INTRODUCTION

Adaptive algorithms are a mainstay of Digital Signal Processing (DSP) [1]. They are used in a variety of applications including acoustic echo cancellation, radar guidance systems, and wireless channel estimation [1-4], among many others. An adaptive algorithm is used to estimate a time varying signal. There are many adaptive algorithms such as Recursive Least Square (RLS) [5] and Kalman filters [6], but the most commonly used is the Least Mean Square (LMS) algorithm. It is a simple but powerful algorithm that can be implemented to take advantage of Lattice FPGA architectures.

Wireless communication systems require signal processing techniques that improve the link performance in hostile mobile radio environments. The mobile radio

channel is particularly dynamic due to multipath propagation and Doppler spread, these effects have strong negative impact on the Bit Error Rate (BER) of any modulation technique. Mobile radio channel impairments cause signal at the receiver to distort or fade significantly. Equalization [6] compensates for inter-symbol interference (ISI) created by multipath within time dispersive channels. An equalizer within a receiver compensates for the average range of expected channel amplitude and delay characteristics. Equalizers must be adaptive since the channel is generally unknown and time varying. For the time varying channel, adaptive LMS algorithm is used, so that the instantaneous values are taken to minimize the error, according to variation of time. For this, an adaptive system is developed as discussed in section II. Training the generic adaptive equalizer is developed in section III. Derivation of LMS algorithm and its realization are discussed in section IV and V respectively.

## II. ADAPTIVE SYSTEM

Before pursuing the study of adaptive systems, it is important to reflect briefly on the implications of adaptive systems in engineering design. Engineering system design usually begins with specifications. First the problem domain

Technical Article  
First Online on – 30 Dec 2014, Revised on – 30 March 2020

© 2020 RAME Publishers  
This is an open access article under the CC BY 4.0 International License  
<https://creativecommons.org/licenses/by/4.0/>

Cite this article – Kaliprasanna Swain and Manoj Kumar Sahoo, “FPGA Implementation of LMS Algorithm Used in Adaptive Equalizer”, *International Journal of Computational and Electronics Aspects in Engineering*, RAME Publishers, vol. 1, issue 1, pp. 45-51, 2014, Revised in 2020.  
<https://doi.org/10.26706/ijceae.1.1.20141209>

is studied, specifications are established, and then a system is built to meet the specifications. The key point is that the system is built to meet the current specifications and will always use the designed set of parameters, even if the external conditions change.

An adaptive system is more complex because it not only has to accomplish the desired task, but also has to be equipped with a subsystem that adapts its parameters. But notice that even if the data changes in the future, this design methodology will modify the system parameters such that the best possible performance is obtained. Additionally, the same system can be used for multiple problems. Adaptive equalizer algorithms are usually simulated and implemented at baseband [7].

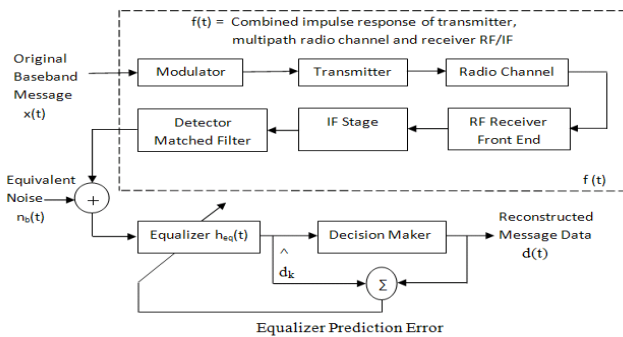


Figure.1 Block diagram of a simplified communication system using an adaptive equalizer at the receiver.

Figure 1 shows a block diagram of a communication system with an adaptive equalizer in the receiver. If  $x(t)$  is the original information signal, and  $f(t)$  is the combined complex baseband impulse response [8] of the transmitter, channel, and the RF/IF sections of the receiver. The signal received by the equalizer may be expressed as:

$$y(t) = x(t) \otimes f^*(t) + n_b(t) \tag{1}$$

where  $f^*(t)$  denotes the complex conjugate of  $f(t)$ ,  $n_b(t)$  is the baseband noise at the input of the equalizer, and  $\otimes$  denotes the convolution operation.

If the impulse response of the equalizer is  $h_{eq}(t)$ , then the output of the equalizer [8] is,

$$\begin{aligned} d(t) &= x(t) \otimes f^*(t) \otimes h_{eq}(t) + n_b(t) \otimes h_{eq}(t) \\ &= x(t) \otimes g(t) + n_b(t) \otimes h_{eq}(t) \end{aligned} \tag{2}$$

where  $g(t) = f^*(t) \otimes h_{eq}(t)$ , is the combined impulse response of the transmitter, channel, RF/IF section of the receiver, and the equalizer at the receiver.

### III. TRAINING A GENERIC ADAPTIVE EQUALIZER

An adaptive equalizer [9] is a time-varying filter which must constantly be returned. The basic structure of an adaptive equalizer is shown in figure 2, where the subscript  $k$  is used to denote a discrete time index.

In figure 2, there is a single input  $y_k$  into the equalizer at any time instant. The value of  $y_k$  depends upon the instantaneous state of the radio channel and the specific value of the noise.

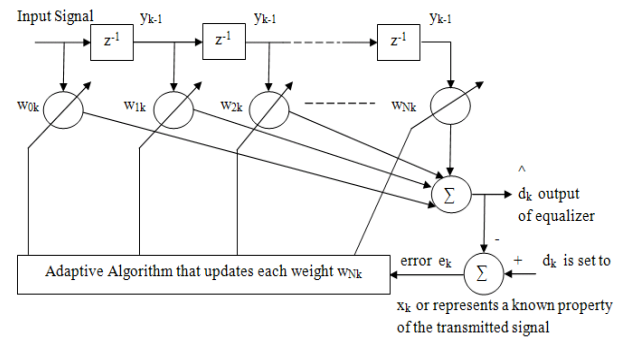


Figure. 2 A basic linear equalizer during training.

The adaptive equalizer structure shown above is called a transversal filter, and in this case has  $N$  delay element,  $N+1$  taps, and  $N+1$  tunable complex multipliers, called weights. These weights are updated continuously by the adaptive algorithm, either on a sample by sample basis (i.e. when  $k$  is incremented by one) or on a block by basis [10]. The adaptive algorithm is controlled by the error signal  $e_k$ . This error signal is derived by comparing the output of the equalizer,  $d_k$ , with some signal  $d_k$  which is either an exact scaled replica of the transmitted signal  $x_k$ . The adaptive algorithm uses  $e_k$  to minimize a cost function and updates the equalizer weights in a manner that iteratively reduces the cost function. For example least mean square (LMS) algorithm searches for the optimum or near-optimum filter weights by performing the following iterative operation:

$$\begin{aligned} \text{New weights} &= \text{PW} + (\text{PE}) \times (\text{CIV}) \\ \text{PW} &= \text{Previous weights} \end{aligned} \tag{3}$$

PE= Previous error

CIV= Current input vector

Where, Previous error = PDO - PAO (4)

PDO= Previous desired output

PAO= Previous actual output

Based on equalization theory [11], the most common function is the mean square error (MSE) [12] between the desired signal and the output of the equalizer. The MSE is denoted by  $E[e(k) e^*(k)]$ , and a known training sequence must be periodically transmitted when a replica of the transmitted signal is required at the output of the equalizer (i.e. when  $d_k$  is set equal to  $x_k$ ). By detecting the training sequence [13], the adaptive algorithm in the receiver is able to compute and minimize the cost function by driving the tap weights until the next training sequence is sent.

To study the adaptive equalizer of figure (2), it is helpful to use vector and matrix algebra. The input signal to the equalizer is a vector  $y_k$ , where

$$y_k = [y_k \ y_{k-1} \ y_{k-2} \ \dots \ y_{k-N}]^T \quad (5)$$

It should be clear that the output of the adaptive equalizer is a scalar given by

$$\hat{d}_k = \sum_{n=0}^N W_{nk} y_{k-n} \quad (6)$$

and following equation (5) is a weight vector and can be written as

$$w_k = [w_{0k} \ w_{1k} \ w_{2k} \ \dots \ w_{Nk}]^T \quad (7)$$

Using equation (5) and (6), equation (6) may be written in vector notation as

$$\hat{d}_k = y_k^T w_k = w_k^T y_k \quad (8)$$

It follows that when the desired equalizer output is known (i.e.  $d_k = x_k$ ), the error signal  $e_k$  is given by

$$e_k = d_k - \hat{d}_k = x_k - \hat{d}_k \quad (9)$$

and from equation (8)

$$e_k = x_k - y_k^T w_k = x_k - w_k^T y_k \quad (10)$$

To compute the mean square error  $|e_k|^2$  at time instant  $k$ , equation (10) is squared to obtain

$$|e_k|^2 = x_k^2 + w_k^T y_k y_k^T w_k - 2 x_k y_k^T w_k \quad (11)$$

Taking the expected value of  $|e_k|^2$  over  $k$  (which in practice amount to computing a time average) yields

$$E[|e_k|^2] = E[x_k^2] + w_k^T E[y_k y_k^T] w_k - 2 E[x_k y_k^T] w_k \quad (12)$$

The cross correlation vector  $p$  between the desired response

$\hat{d}_k = x_k$  and the input signal  $y_k$  is defined as

$$p = E[x_k y_k] = E[x_k y_k \ x_k y_{k-1} \ x_k y_{k-2} \ \dots \ x_k y_{k-N}]^T \quad (13)$$

and the input correlation matrix is defined as the  $(N+1) \times (N+1)$  square matrix  $R$ .

$$R = E[y_k y_k^*] = E \begin{bmatrix} y_k^2 & y_k y_{k-1} & y_k y_{k-2} & \dots & y_k y_{k-N} \\ y_{k-1} y_k & y_{k-1}^2 & y_{k-1} y_{k-2} & \dots & y_{k-1} y_{k-N} \\ \dots & \dots & \dots & \dots & \dots \\ y_{k-N} y_k & y_{k-N} y_{k-1} & y_{k-N} y_{k-2} & \dots & y_{k-N}^2 \end{bmatrix} \quad (14)$$

The matrix  $R$  called as the input covariance matrix. The major diagonal of  $R$  contains the mean square values of each input sample, and the cross terms specify the autocorrelation terms resulting from delayed samples of the input signal.

Using equation (13) and (14), equation (12) may be rewritten as,

$$\text{Mean Square Error} \equiv \xi = E[x_k^2] + w^T R w - 2 p^T w \quad (15)$$

By minimizing equation (15) in terms of the weight vector  $w_k$ , it becomes possible to adaptively tune the equalizer to minimize ISI (Inter Symbol Interference) in the received signal. This is due to the fact that, when the input signal  $y_k$  and the desired output response  $\hat{d}_k = x_k$  are stationary, the mean square error (MSE) is quadratic on  $w_k$ , and minimizing the MSE leads to optimal solution for  $w_k$ .

Using LMS algorithm the optimal solution for  $w_k$  can be obtained and details are discussed in next section.

#### IV. DERIVATION OF LMS ALGORITHM

The LMS equalizer is a more robust equalizer where the criterion used is the minimization of the mean square error (MSE) between the desired equalizer output and the actual equalizer output.

The prediction error is given by:

$$\epsilon_k = d_k - \hat{d}_k = x_k - \hat{d}_k \quad (16)$$

It can be shown that

$$\epsilon_k = x_k - w_k^T y_k \quad (17)$$

To compute the mean square error at time instant  $k$ , we obtain the error as

$$\zeta = E [\varepsilon_k \varepsilon_k^*] \tag{18}$$

The LMS algorithm seeks to minimize the mean square error given in the previous equation.

For a specific channel condition, the prediction error  $\varepsilon_k$  is dependent on the tap gain vector  $w_N$ , so of an equalizer is the function of  $w_N$ . Let the cost function  $J(w_N)$  denote the mean squared error as a function of tap gain vector  $w_N$ .

In order to minimize MSE, set the derivative to zero.

$$\partial J(w_N) / \partial w_N = -2p_N + 2R_{NN} w_N = 0 \tag{19}$$

Simplifying above equation we get

$$R_{NN} \hat{w}_N = p_N \tag{20}$$

Above equation is called the normal equation since the error is minimized and is made orthogonal to the projection related to the desired signal. When the above equation is satisfied the MMSE of the equalizer is

$$J_{opt} = J(\hat{w}_N) = E [x_k x_k^*] - p_N^T \hat{w}_N \tag{21}$$

To obtain the optimal tap gain vector  $\hat{w}_N$ , the normal equation in (20) must be solved iteratively as the equalizer converges to an acceptably small value of  $J_{opt}$ . One technique is to calculate

$$\hat{w} = R_{NN}^{-1} p_N \tag{22}$$

The LMS equalizer maximizes the signal to distortion ratio at its output within the constraints of the equalizer filter length. If an input signal has a time dispersion characteristic that is greater than proportional delay through the equalizer, then the equalizer will be unable to reduce distortion. The convergence rate of LMS algorithm is slow due to the fact that there is only one parameter, the step size  $\alpha$ , that controls the adaptation rate. So by taking suitable value for  $\alpha$ , which is described in next section, we can reduce the error.

### V. DESIGN OF COMMUNICATION SYSTEM USING AN ADAPTIVE EQUALIZER AT RECEIVER END

In fig.3 we are using FDA Tool, in which we are using LPF (low pass filter) of order 32 to get the better response. Sinusoidal signal with noise is given to Direct Form II Transpose filter in which we are using the LPF, so that it will allow only the required signal, eliminating

the instantaneous. Filter taps are used to point out the taps at each manipulation of the signal. Frequency response is plotted by using FFT block, which is connected to the tap points by using system generating the co-simulation block which is used to implement the design in FPGA, and finally compared the simulink output with FPGA output.

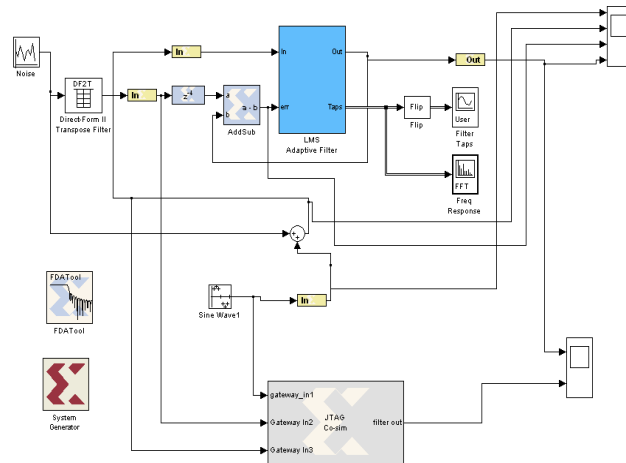


Figure. 3 Complete Block Diagram of LMS Algorithm

The blocks which are used inside the LMS adaptive filter subsystem are as shown in figure 4.

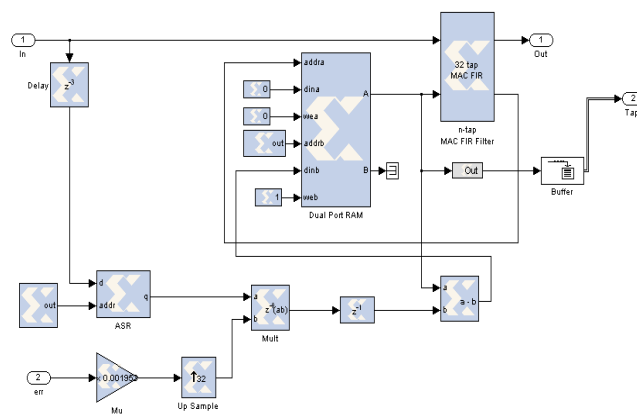


Figure. 4 Blocks inside The LMS Adaptive Filter

In LMS subsystem block (shown in figure 4), dual port block RAM is used to store the large amount of data. 32 tap MAC(Multiply and Accumulator) FIR Filter is used to manipulate the signal 32 times by multiplying the signal with corresponding window with its multiplier to minimize the error in signal. Finally the buffer is used to store the signal. Counter is used to count the signal till the loop is repeated for 32 times. Finally the signal is error-free and is given at the output port.

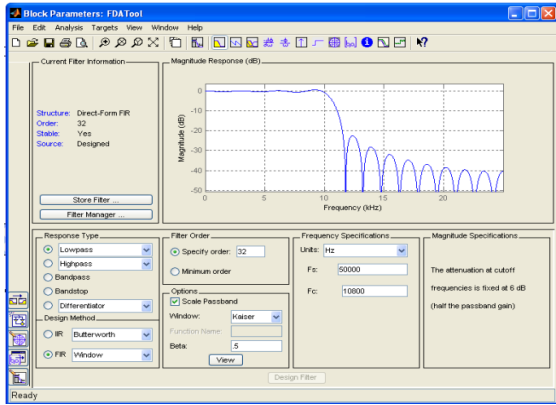


Figure. 5 Filter Design Analysis Tool

FDA tool is used to design required filter. This is shown in figure 5. This tool is used to specify the corresponding window so as to minimize the error and also to specify the order of the filter. This FDA tool is used in Direct form II Transpose Filter in Fig.3 for minimizing the noise.

### VI. CO-SIMULATION RESULTS

JTAG stand for Joint Test Action Group. JTAG boundary scan started as a method of testing ICs and their interconnections using a shift register built into the chip so that inputs could be shifted-in and the resulting outputs could be shifted-out using only four I/O pins (clock, input data, output data, and state machine mode control). This eliminated the need for complex, expensive, bed-of-nails cards for low-speed probing of IC I/O pins. When a model is implemented for JTAG hardware co-simulation, a new library is created that contains a custom JTAG co-simulation block with ports that match the gateway names from the original model. The co-simulation block interacts with the FPGA hardware platform during a Simulink simulation. Simulation data that is written to the input ports of the block are passed to the hardware by the block. Conversely, when data is read from the co-simulation block's output ports, the block reads the appropriate values from the hardware and drives them on the output ports so they can be interpreted in Simulink. In addition, the block automatically opens, configures, steps, and closes the platform.

### VII. SIMULATION AND RESULTS

A model of the circuit in figure 3 was simulated using Simulink and the results are shown in figure 6-12.

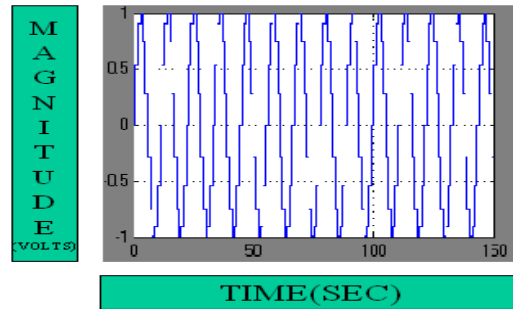


Figure. 6 Sinusoidal input of transmitted signal

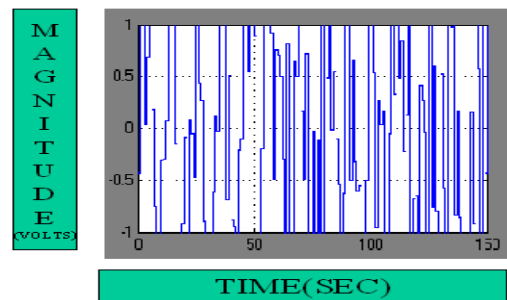


Figure. 7 Noise added to transmitted signal in the channel

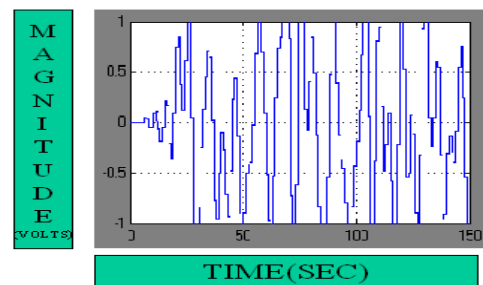


Figure. 8 Sinusoidal input with noise in the transmitted signal

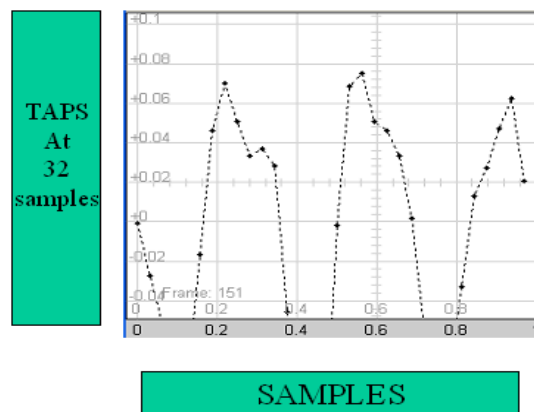


Figure. 9 Filter taps in the training sequence

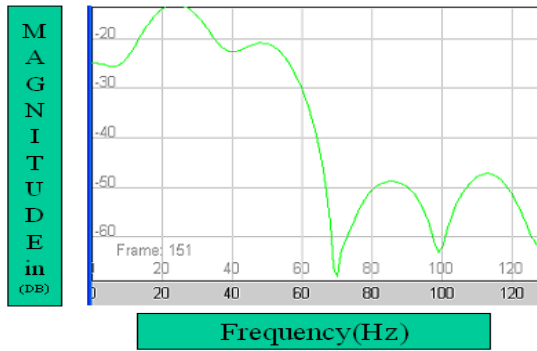


Figure. 10 Frequency response of signal at receiving end

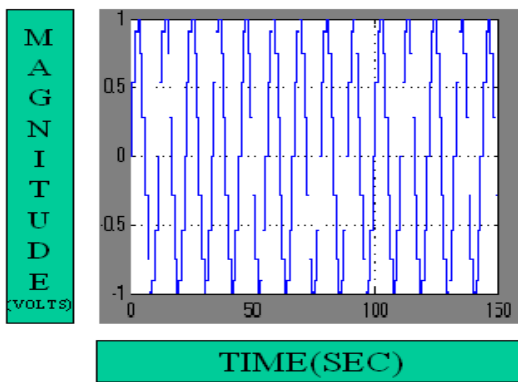


Figure. 11 Sinusoidal output at receiving end

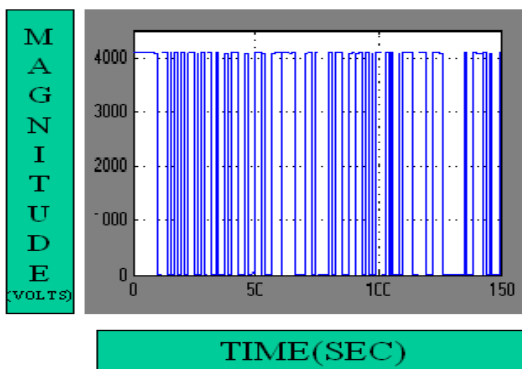


Figure. 12 FPGA output at receiving end

Figure 6 shows the sinusoidal input signal. Gaussian noise is generated using noise block shown in figure 3 and generated signal is shown in figure 7. Figure 8 shows the sinusoidal input with noise in transmitted signal, so that the signal is distorted. For rectifying the signal, adaptive LMS algorithm is used and is the purpose of the paper. This adaptive algorithm takes the instantaneous values of signal and at each and every time the signal is multiplied by the weights to minimize the error. In this algorithm, 32 filter taps are used to reduce

the error. Simulations of the 32 tap indicate that our analysis of the block LMS algorithm is indeed accurate. 32 Filter taps are shown in figure 9. In figure 10, the frequency response is shown for the low pass filter at the receiving end. Finally at the receiving end, the signal is reconstructed to its original form without any error by using our technique. This result is shown in figure 11. It indicates that our method eliminates noise which was added to the transmitted signal. Now this can be tested in prototype hardware to make industry ready using FPGA. For implementation in FPGA, the JTAG block is generated with system generator block in Xilinx blockset in MATLAB by specifying the parameters in System Generator. Then the output is verified in FPGA. In FPGA, the output is displayed as pulses by taking the synchronous edges of sinusoidal input as shown in figure 12. As the time is going on increasing, the signal to noise ratio is gradually increasing in magnitude. Comparing our theoretical predictions with simulation behaviour indicates that our analysis is accurate for predicting the mean square performance.

## VI. CONCLUSION

The main contribution of the paper was to propose, to develop, and to assess the performance of an adaptive step-size LMS algorithm for interference suppression in wireless communication systems and its hardware implementation using FPGA. In this paper, we presented MSE analysis of Adaptive LMS algorithm. Analysis has yielded a set of difference equations that describe the mean square behaviour of the algorithm, and a formula for the steady state excess MSE was derived. Steady state MSE analysis, supported by experimental results, indicated that the algorithm misadjustment essentially depends on step size ( $\alpha$ ) with a very little effect of the input signal and the adaptive filter length. The analysis was validated through simulations showing the input and output almost equal by reducing the error in received signal. Its implementation at hardware and the result obtained from the output of hardware shows that our technique can be used to minimize error and can be used directly in hardware using FPGA.

REFERENCES

- [1] Simon Haykin, *Adaptive Filter Theory*, Prentice-Hall, Inc., 1996.
- [2] J. Homer, "Quantifying the convergence speed of LMS adaptive fir filter with autoregressive inputs," *Electronics Letters*, vol. 36, no. 6, pp. 585–586, March 2000.
- [3] Huijuan Cui Yuantao Gu, Kun Tang and Wen Du, "Modifier formula on mean square convergence of lms algorithm," *Electronics Letters*, vol. 38, no. 19, pp. 1147 – 1148, Sep 2002.
- [4] M. Chakraborty and H. Sakai, "Convergence analysis of complex lms algorithm with tonal reference signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 2, pp. 286 – 292, March 2005.
- [5] A.H. Sayed and T. Kailath, "A state-space approach to adaptive rls filtering," *Signal Processing Magazine, IEEE*, vol. 11, no. 3, pp. 18 – 60, July 1994.
- [6] Brian D O Anderson, *Optimal Filtering*, Dover Publications, 2005.
- [7] Tranter, W. H., Shanmugan, K., Rappaport, T. S., and Kosbar, K., *Computer-Aided Design and Analysis of Communications Systems with Wireless Applications*, Prentice-Hall, Upper Saddle River, NJ, 2002.
- [8] Crozier, S. N., Falconer, D. D., and Mahmoud, S., "Short Block Equalization Techniques Employing Channel Estimation for Fading, Time Dispersive Channels", *IEEE Vehicular Technology Conference*, San Francisco, pp. 142-146, 1989.
- [9] Lo, N. K. W., Falconer, D. D., and Sheikh, A. U. H., "Adaptive Equalization and Diversity Combining for a Mobile Radio Channel", *IEEE Globecom*, San Diego, December 1990.
- [10] Molisch, A. F., Ed., *Wideband Wireless Digital Communications*, Prentice-Hall, Upper Saddle River, NJ, 2001.
- [11] Windrow, B., and Stearns, S. D., *Adaptive Signal Processing*, Prentice Hall, 1985.
- [12] Qureshi, S. U. H., "Adaptive Equalization", *Proceeding of IEEE*, Vol. 37, No. 9, pp. 1340-1387, September, 1985.
- [13] EIA/TIA Interim Standard, "Cellular System Dual Mode Mobile Station – Land Station Compatibility Specifications", IS-54, *Electronic Industries Association*, May 1990.