Research Journal of Recent Sciences ISSN 2277-2502
Vol. 3(**ISC-2013**), 4-8 (**2014**) *Res. J. Recent. Sci.*

Noise Cancellation Using Adaptive Filters in FPGA

Anjana S.[1] and Anitha Mary X.[2]
[1]Embedded Systems, Karunya University, Coimbatore, INDIA
[2]Dept of EIE, Karunya University, Coimbatore, INDIA

Available online at: www.isca.in, www.isca.me
Received 10th December 2013, revised 31st December 2013, accepted 4th February 2014

Abstract

Adaptive filters have gained popularity over the years due to their ability to adapt themselves to different environmental situations without substantial intervention by the user. The implementation of an adaptive noise cancellation filter process is done here. The filter is designed using the Recursive least square (RLS) algorithm due to its computational simplicity, robust behavior when implemented in finite-precision hardware and well understood convergence behavior. The correctness and response of the adaptive noise cancellation filter can be checked by the RLS algorithm using the Matlab/ Simulink tool. To implement this algorithm the Simulink model is used as a reference using the Xilinx Tool Box. To implement the adaptive filter on Xilinx, the System Generator ("SysGen") tool in the Xilinx block set is used to generate the bit file which can be downloaded onto the FPGA through hardware co-simulation. This project presents the adaptive noise cancellation filter using RLS algorithm suitable for noise cancellation and the results are verified by plotting the output using MATLAB.

Keywords: Cancelling noise, adaptive filtering process, recursive least square (rls) algorithm.

Introduction

A filter is a one which selects or controls the characteristics of the signal it produces, by conditioning the incoming signal. In some cases, however, the required filter characteristic is unknown; we know only the output required from a test-input signal. When a signal is distorted by transmission through the atmosphere, or an imperfect communication system, it needs a filter that can convolve or equalize the distorted signal to produce the original, undistorted signal[1]. Such a filter which adapts itself to the distorted signal present in the system us is known as an adaptive filter. An example is an echo canceller: the desired output cancels the echo signal (an output result of zero when there is no other input signal). In this case, the coefficients cannot be determined initially since they depend on changing line or transmission conditions[1]. For applications such as echo canceller, it is necessary to rely on adaptive filtering techniques.

Adaptive filter varies in time by adapting its coefficients based on its references. The adaptive filters are accomplished to minimize the error in the estimation according to a certain criteria, which leads to the adaptive filtering process. For example, when a telephone call is initiated, the transfer function of the telephone channel is unknown. This produces an undesirable echoes which involves signal reflection. The above problem is corrected with a linear filter after the transfer function is measured[2]. Only after the process starts running the measurements can be made. Therefore, the correction filter must be designed by the system itself, and in a reasonably short time. This can be done using adaptive filters. Recursive Least Square (RLS) algorithm is used here to design the adaptive filter. The RLS adaptive algorithm is mostly used in applications where noise cancellation, line enhancing, etc., is requierd to design a filter[3]. The designing does not require any frequency information or specification. The self learning process is defined by a adaptive filtering algorithm, which reduces the error between the output and the desired signal. Initially zero values are set for the weight. When the RLS performance criteria for error signal have achieved its minimum value through the iterations of the adaptive algorithm, the filter completes its weight updation process and its coefficients have converged to a solution[4]. The output thus obtained from the adaptive filter matches closely the desired signal. When the input data characteristics changes, the filter environment by generating a new set of coefficients for the new data, it is noticed that when error of the signal goes to zero a perfect adaptation can be achieved, a tedious process to occur in the real world.

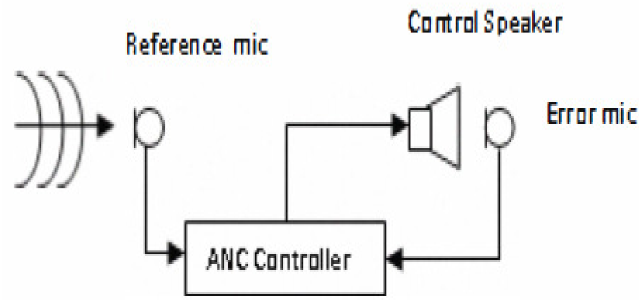


Figure-1
Concept of ANC

ANC systems differs its architecture based on its goal. When the ambient sounds is reduced in a process, a simple architecture with a feedback system can be used. As in our case, to compensate a single source of unwanted sound we require a feed forward system[1]. A feed forward system is characterized by two inputs per channel: one reference signal, and an error.

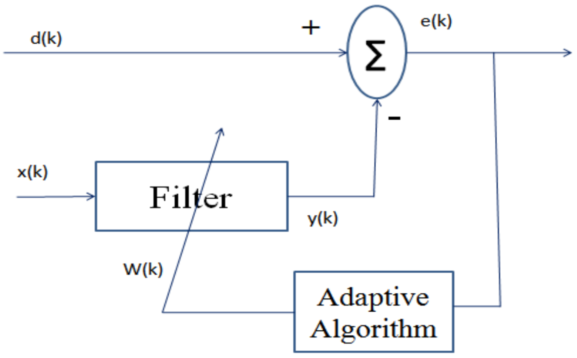


Figure-2
Adaptive filtering process

The first inputs is primary signal which is wanted signal corrupted by noise. The reference signal is taken as the second input signal, this can be an interfacing noise supposed to be uncorre1ated with the wanted signal but correlated with noise affecting original signal in an unknown way[5]. The filter output signal is an estimate of the noise signal with inverted sign. The filter output signal and the desired signal are superposed, the noise signal is cancelled and the result is an error signal which constitutes the system output. The adaptive filtering operation achieved the best results when system output is noise free. This goal is achieved by minimizing the mean square of the error signal. This method implements the RLS algorithm, due to its reduced complexity and less computations. This is advantageous in the implementation cancellation filter on a Field Programmable Gate Array (FPGA) with optimal resource utilization[6].

Algorithm Description: Two modules are used to compose the process of an adaptive filters:
adaptive system: a digital filter that is defined in a time instant n by a coefficient vector and the inputis proposed. The coefficient and the input vectors are as given below
$W_k = w_n(0), w_n(1), \ldots, w_n(\mathrm{k}-1)$
$X_k = x(n), x(n-1), \ldots, x(n-k+1)$
the output signal y(n) is a convolution between the input and the coefficient vectors.

adaptive algorithm: the coefficients are adjusted in such a way the adaptive system output $y(k)$ and the desired signal $d(k)$ becomes equal in value. The error signal is the similarity between both the signals[7].
$e(k) = d(k) - y(k)$
The error signal $e(k)$ is used to adjust the coefficients of the adaptive system for the next time instant. For the coefficient adjustment there are many algorithms used. Here in this section two different algorithms are discussed: i. LMS Algorithm, ii. RLS Algorithm

LMS Algorithm: LMS filters is good, the filter length is short, the simple structure and the smaller operation makes the process easy when realized in a hardware. The LMS algorithms is a gradient-type search for time track-varying signal characteristics[8].

Filtering: $y(k) = w^{T}(k)x(k)$
Error estimation: $e(k)=d(k)-y(k)$
Tap-weight vector adaptation: $w(k+1) = w(k) + 2\mu e(k)x(k)$

The LMS algorithm of adaptive filter had a few disadvantages. It has a low rate of convergence and only a single parameter is used to tune for convergence.

RLS Algorithm: The Recursive least squares (RLS) is an algorithm for adaptive filters that can recursively find the filter coefficients. This minimizes a weighted linear least squares cost function relating to the input signals. The mean square error can be reduced using this algorithm[8]. The input signals are deterministic for RLS algorithm, whereas stochastic for the LMS. The cost of high computational complexity is the benefit that which exhibits extremely a fast convergence when compared with the other algorithms.

The minimization of a certain cost function is a feature of adaptive filters. The quadratic error is most widely used. The cost function is made optimum when the error follows the Gaussian distribution. The cost function commonly used provides a good result,
$$\mathrm{J(k)} = \log\,[\cosh\,(\beta \cdot \mathrm{e(k)})]/\beta \qquad (1)$$
Where $\beta \in \mathrm{R} +$ is a positive integer.

The RLS algorithms are known for its excellent performance when working in time varying environments but at the cost of an increased computational complexity and some stability problems[9]. The Delta rule is used for the update of the weight coefficients, we obtain the following equation where $\mu \in \_+$ is the learning rate:
$$\mathrm{Wk{+}1} = \mathrm{wk} + \mu \cdot \tanh\,[\beta \cdot \mathrm{e(k)}] \cdot \mathrm{xk} \qquad (2)$$

The intermediate gain vector used to compute tap weights are as given below:
$$m(k) = \underline{U(k)}/(\lambda + \mathrm{x}^{\mathrm{T}}(\mathrm{k})\mathrm{u}(\mathrm{k})) \qquad (3)$$
$$u(k) = \overline{w}_{\lambda}^{-1}(k-1)x(k) \qquad (4)$$
λ - a small positive constant <1.

The filter output and the error signal are calculated as shown below:
$$\overline{y_{k-1}}(k) = \overline{w}^{T}(k-1)x(k) \qquad (5)$$
$$\bar{e}_{k-1}(k) = d(k) - \bar{y}_{k-1}(k) \qquad (6)$$

The estimate of previous samples of output signal, error signal and filter weight in RLS algorithm is required. This leads to a large memory requirement.

Simulation Results

In this section the performance of the RLS algorithms and LMS is shown. The algorithm is implemented according to the steps mentioned above. The RLS algorithm is first tested for its efficiency with a sinusoidal signal. Here the MATLAB programming language is used .Initially the sine wave is generated using the MATLAB comments. The input signal (desired signal)is a sinusoidal signal. Now a noise signal with known frequency is generated. The noise signal is mixed with the input signal (desired signal) and is given as an input to the adaptive filter. Now the filter process is carried over and the output is obtained. The obtained filter output is more or less equal to that of the desired input signal. This process has many iterations that has been done and the weight is been updated based on the error signal obtained from the difference between the desired and the filter output signal. The weight is initially set to zero and it is updated based on the error signal. In the simulation the reference input signal x(k) is a white Gaussian noise generated using randn function a power of two dB signal, the desired signal d(k) ,is obtained by delaying the version of x(k) into clean signal s(k), d(k) = s(k) + x1(k) as shown in Figure-3. And for LMS its shown using figure-4.

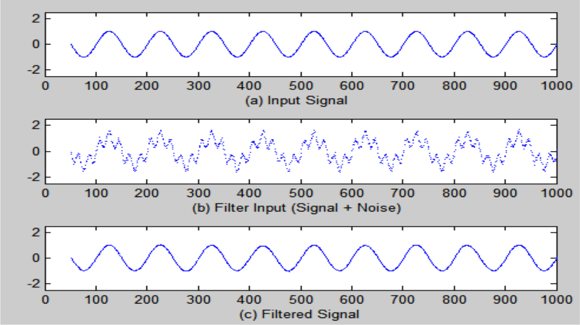


Figure-3
Sinusoidal Signal-RLS Algorithm
(a)input signal,(b)filter input,(c)filtered output

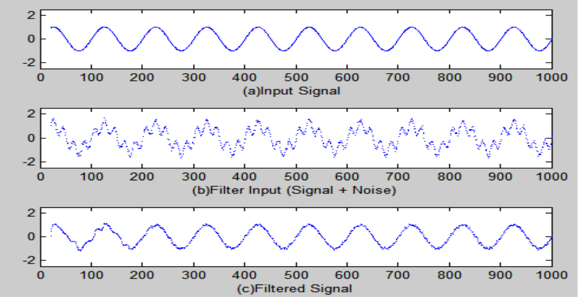


Figure-4
Sinusoidal signal-LMS Algorithm
(a)input signal,(b)filter input,(c)filtered output

Update the weight W_k by,

$$W_k= W_{k-1} + 2\mu x_k e_k \quad (7)$$

The weight vector is a column as:

$$W=\begin{bmatrix} w0 \\ w1 \\ . \\ . \\ . \\ wn- \end{bmatrix} \quad (8)$$

X_k is the column vector of regressors (or past values of input) given by

$$X_k=\begin{bmatrix} u_k \\ u_{k-1} \\ . \\ . \\ . \\ u_{k-n} \end{bmatrix} \quad (9)$$

If n is the system order then there are n + 1 weights. Thus by updation of weight and calculation of error signals, the output obtained is more or less equal to the desired input by cancelling the noise added to it. The same is repeated for an audio signal with 250kB as an input and the output is as shown in figure-5 for RLS algorithm and figure-6 for LMS algorithm.

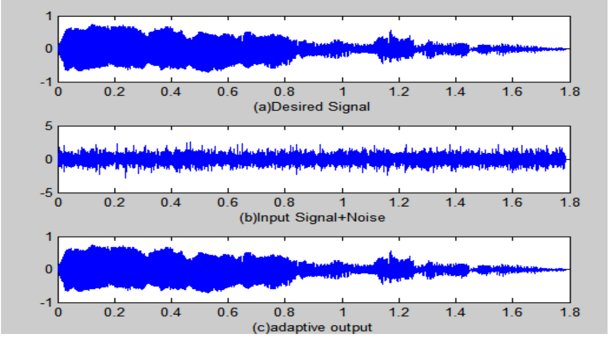


Figure-5
An Audio signal-RLS Algorithm
(a)desired signal, (b)input+noise, (c)adaptive output

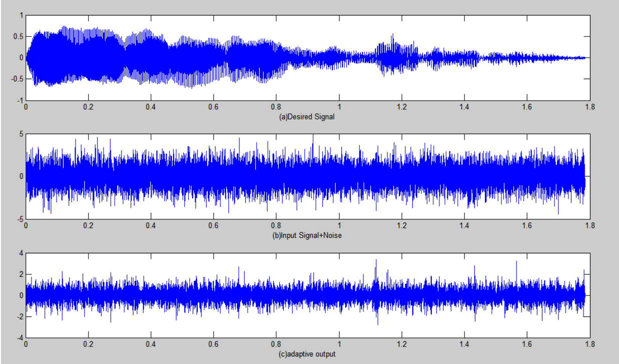
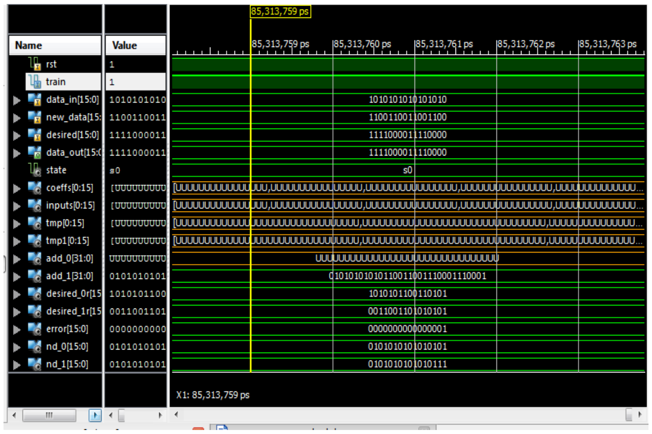


Figure-6
An Audio signal-LMS Algorithm
(a)desired signal, (b)input+noise, (c)adaptive output

From the simulation results shown it is understood that, RLS for adaptive filters gives a higher quality in the disturbance and noise cancellation. The LMS algorithms though they are simpler in evaluation process, they attain lower quality in the cancellation of disturbing signals. Cancelling the noise has been the objective, so RLS algorithms achieves a higher quality in the disturbing signal and noise cancellation. The highest convergence speed with a lesser mean square error and a highest noise reduction percentage is provided by a non stationary signal with the RLS algorithm. This is comparatively low in LMS algorithm, but at the cost of large computational complexity and memory requirement.

The continuation of the same process is done using the Xilinx software. Here the input is given in terms of bits. By providing the input bit and the new bit (i.e., the noise bit) and following the process the output is not mere but more or less equal to the input given.The MATLAB code was converted using the hardware co-simulator tool in the Simulink Xilinx block set. Thus the output for RLS algorithm is as shown in figure-7.



```
Name               Value        85,313,759 ps
                                85,313,759 ps  85,313,760 ps  85,313,761 ps  85,313,762 ps  85,313,763 ps
rst                1
train              1
data_in[15:0]      1010101010   1010101010101010
new_data[15:       1100110011   1100110011001100
desired[15:0]      1111000011   1111000011110000
data_out[15:0      1111000011   1111000011110000
state              s0           s0
coeffs[0:15]       [UUUUUUUUUU  [UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUU...
inputs[0:15]       [UUUUUUUUUU  [UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUU,UUUUUUUUUUUUU...
tmp[0:15]          [UUUUUUUUUU  [UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU,UUUUUUUUUUUUU...
tmp1[0:15]         [UUUUUUUUUU  [UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU,UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU,UUUUUUUUUUUUU...
add_0[31:0]        UUUUUUUUUU   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
add_1[31:0]        0101010101   01010101010110011001110001110001
desired_0r[15      1010101100   1010101100110101
desired_1r[15      0011001101   0011001101010101
error[15:0]        0000000000   0000000000000001
nd_0[15:0]         0101010101   0101010101010101
nd_1[15:0]         0101010101   0101010101010111

X1: 85,313,759 ps
```

Figure-7
RLS in VHDL

Conclusion

The development of algorithm and design, for a sinusoidal signal is presented in MATLAB and Xilinx software. Adaptive filter varies in time thus by adapting its coefficients according to the reference inputs using RLS algorithm. The estimation of an unknown random signal is often faced in the presence of noise. The minimization of the error in the estimation according to a certain criteria is done using adaptive filters. The RLS for adaptive filters gives a higher quality in the disturbance and noise cancellation. The LMS algorithms are simpler in process evaluation. In the cancellation of disturbing signals they attain lower performance. When compared the RLS algorithms in the disturbing signal cancellation achieves a higher performance. The RLS algorithm when used with RLS filter coefficient have large numerical claims.

Future Work: The implementation process can be done using FPGA for checking the above mentioned process used in real time applications, the MATLAB code can be loaded to an FPGA using hardware co-simulator tool in the Simulink Xilinx tool set.

References

1. M.A. Gandhi, C. Ledoux and L. Mili, Robust estimation method for impulsive noise suppression in speech, IEEE *Int. Symp. SignalProcess. Inf. Technol,* **3(7),** 755–760 **(2004)**

2. J. Armstrong, H. A. Suraweera, C. Chai and M. Feramez, Impulse noise mitigation techniques for OFDM receivers and their application in digital video broadcasting, Mediterr, *J. Electron. Commun,* **1(1)**, 1–10 **(2005)**

3. E.S. Nejevenko and A.A. Sotnikov, Adaptive modeling for hydroacoustic signal processing, Pattern Recognit, *Image Analysis,* **16(1)**, 5–8 **(2006)**

4. Tuning Fuzzy Control Rules via Genetic Algorithms: An Experimental Evaluation Pitalúa Díaz N., Lagunas Jiménez R. and González Angelesa, *Res. J. Recent Sci.,* **2(10),** 81-87 **(2013)**

5. J. Benesty, T. Gänsler, D.R. Morgan, M.M. Sondhi and S.L. Gay, Advances in Network and Acoustic Echo Cancellation, Berlin, Germany: Springer-Verlag, **(2001)**

6. Comparative Survey on Time Interleaved Analog to Digital Converter Mismatches Compensation Techniques Hafiz Allah Nawaz, Abida Sharif and Muhammad Sharif, *Research Journal of Recent Sciences,* **2(9)**, 95-100 **(2013)**

7. I. Aizenberg, T. Bregin and D. Paliy, Method for the impulsive noise detection and its application for the improvement of the impulsive noise filtering algorithms, in Image Process. Conf., San Jose, CA, **4667**, 204–214 **(2002)**

8. U.M. Baese, Digital Signal Processing With Field Programmable Gate Arrays, 3rd ed. Berlin, Germany:Springer-Verlag, **(2007)**

9. A.Ordaz-Moreno, R. de Jesus Romero-Troncoso, J. A. Vite-Frias, J. R. Rivera-Gillen, and A. Garcia-Perez, Automatic online diagnosis algorithm for broken-bar detection on induction motors based on discrete wavelet transform for FPGA implementation, *IEEE Trans. Ind. Electron.,* **55(5)**, 2193–2202 **(2008)**

10. Enhanced SLAM for a Mobile Robot using Unscented Kalman Filter and Radial Basis Function Neural Network Panah Amir, *Res. J. Recent Sci.,* **2(2)**, 69-75 **(2013)**

11. D. Zhang and H. Li, A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms, *IEEE Trans. Ind.Electron.,* **55(2),** 551–561 **(2008)**

12. Z. Shu, Y. Guo and J. Lian, Steady-state and dynamic study of active power filter with efficient FPGA-based control algorithm, *IEEE Trans.Ind. Electron.,* **55(4),** 1527–1536 **(2008)**

13. T.P. Pander, A suppression of an impulsive noise in ECG signal processing, in *Proc. Conf. IEEE Eng. Med. Biol. Soc.,* **1(3),** 596–599 **(2005)**

14. E. Soria, J.D. Martín, G. Camps, A.J. Serrano, J. Calpe and L. Gómez, A low-complexity fuzzy activation function for artificial neural networks, *IEEE Trans. Neural Netw.,* **14(6)**, 1576– 1579 **(2003)**