Comparative Performance Analysis of Adaptive Algorithms for Simulation & Hardware Implementation of an ECG Signal

D.C. Dhubkarya ¹, Aastha Katara ²
¹ Department of Electronics & Communication, BIET, Jhansi, India
² Department of Electronics & Instrumentation, Anand Engineering College, Agra, India
Email: dcd3580@yahoo.com, aastha.katara@gmail.com

Abstract — The cancellation of noise & distortion is the main area of concern in many applications of communications, signal processing and biomedical. The adaptive filters are attractive to work with unpredictable noisy inputs and are capable to work in real-time & non-realtime environments. This work emphasizes on the performance comparison of MATLAB Simulation and DSP Processor implementation of an adaptive filter for Least Mean Squared (LMS) and Normalized Least Mean Squared (NLMS) Algorithms. The input to the filter is a noisy ECG signal and the output of the filter is approximate to clean ECG signal. The designed filter is tested during simulation as well as hardware implementation for three different noisy samples of an ECG signal. And the performance is analyzed on the basis of Signal to Noise Ratio (SNR) improvement.

Keywords: Adaptive filter, DSP Starter Kit (DSK), ECG Signal, Least Mean Squared (LMS), Normalized Least Mean Squared (NLMS), TMS320C6713.

I. INTRODUCTION

In the process of transmission of information from the source to receiver, noise from the surroundings automatically gets added to the signal. In numerous application areas, including biomedical engineering, radar & sonar engineering, digital communications etc. the goal is to extract the useful signal from the noise corrupted signal. Therefore the effective removal of noise in the field of signal processing is an active area of research. The use of adaptive filter [1] is one of the most popular proposed solutions to reduce the signal corruption caused by unpredictable noise. An adaptive filter has the property of self-modifying its frequency response to change its behavior with time. It allows the filter to adjust the response as the input signal characteristics change.

Adaptive filters work on the principle of minimizing the mean squared error between the filter output and a target (or desired) signal. The general adaptive filter configuration is illustrated in Fig.1. The adaptive filter has two inputs: the primary input d(n), which represents the desired signal corrupted with undesired noise, and the reference signal x(n), which is the undesired noise to be filtered out of the system.

The basic idea for the adaptive filter is to predict the amount of noise in the primary signal, and then subtract that noise from it. The prediction is based on filtering the reference signal x(n), which contains a solid reference of

![Figure 1. General Adaptive filter configuration](image)
Comparative Performance Analysis of Adaptive Algorithms for Simulation & Hardware Implementation of an ECG Signal

the noise present in the primary signal. The noise in the reference signal is filtered to compensate for the amplitude, phase and time delay, and then subtracted from the primary signal. This filtered noise is the system’s prediction of the noise portion of the primary signal, y(n). The resulting signal is called error signal e(n), and it presents the output of the system. Ideally, the resulting error signal would be only the desired portion of the primary signal.

In this paper we have implemented the adaptive filter on software & hardware to compare their relative performance. The MATLAB software is used for simulation purpose and TMS320C6713 DSP Processor is used for hardware implementation. The resultant output of simulation and real-time hardware implementation are compared in terms of Average SNR Improvement for an ECG signal as an input.

II. ADAPTIVE ALGORITHMS

The algorithms used to perform the adaptation, and the configuration of the filter depends directly on the use of the filter. The two classes of adaptive filtering algorithms namely Least Mean Squared (LMS) and Recursive Least Squares (RLS) are capable of performing the adaptation of the filter coefficients. The LMS based algorithms are simple to understand and easy to implement whereas RLS based algorithm are complex and requires so much memory for implementation. So in this work we have focuses on LMS based algorithms.

A. Least Mean Square Algorithm

The LMS algorithm [2], is a type of adaptive filter known as stochastic gradient-based algorithms as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution. With each iteration of the LMS algorithm, the filter tap weights of the adaptive filter are updated according to the following formula:

\[ w(n+1) = w(n) + 2\mu e(n)x(n) \]  

(1)

Here x(n) is the input vector of time delayed input values,

\[ x(n) = [x(n), x(n-1), x(n-2), \ldots, x(n-N+1)]^T \]  

(2)

The vector \( w(n) = [w_0(n), w_1(n), w_2(n), \ldots, w_{N-1}(n)]^T \) represents the coefficients of the adaptive FIR filter tap weight vector at time n.

The parameter \( \mu \) is known as the step size parameter and is a small positive constant. This step size parameter controls the influence of the updating factor. Selection of a suitable value for \( \mu \) is imperative to the performance of the LMS algorithm, if the value is too small, the time taken by the adaptive filter to converge on the optimal solution will be too long; if \( \mu \) is too large, the adaptive filter becomes unstable and its output diverges.

B. Normalized LMS Algorithm

In the standard LMS algorithm, when the convergence factor \( \mu \) is large, the algorithm experiences a gradient noise amplification problem. In order to solve this difficulty, we can use the NLMS (Normalized Least Mean Square) algorithm. The correction applied to the weight vector \( w(n) \) at iteration \( n+1 \) is “normalized” with respect to the squared Euclidian norm of the input vector x(n) at iteration n.

We may view the NLMS algorithm as a time-varying step-size algorithm, calculating the convergence factor \( \mu \) as in Eq. (3)[1].

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

(3)

Where: \( \alpha \) is the NLMS adaption constant, which optimize the convergence rate of the algorithm and should satisfy the condition \( 0 < \alpha < 2 \), and \( c \) is the constant term for normalization and is always less than 1.

The Filter weights are updated by the Eq. (4).
III. EXPERIMENTAL SETUP

A. MATLAB Simulation

For simulation of adaptive algorithms, a MATLAB program is written which implements the mathematical equation of LMS and NLMS algorithms as given in eq.1 & eq.4 respectively. The reference input signal $x(n)$ is a white Gaussian noise of power 1-dB generated using random function in MATLAB, and source signal $s(n)$ is a clean amplified ECG signal recorded with 12-lead configuration [6], the desired signal $d(n)$ obtained by adding a delayed version of $x(n)$ into clean signal $s(n)$, i.e. $d(n) = s(n) + x_d(n)$.

B. Real time DSP hardware Implementation

DSP processors have been very successful because of the development of low-cost software and hardware support. DSP processors are concerned primarily with real-time signal processing. To implement Adaptive filter on DSP Starter Kit (DSK), two methods can be adopted. In first method a C program can be written in code composer studio to perform the desired task, but writing C code for DSK environment is not comfortable for most of the researchers. The second method is quite easy in which we can design a Simulink model and it can be converted into a code composer studio project with the help of Real Time Workshop (RTW) facility available in the MATLAB.

Figure 2 illustrate the block diagram for DSP hardware implementation. A noisy ECG signal of 360mv generated from 12-lead configuration is applied at line-in port of TMS320C6713. Another additional noise reference input is also applied at line-in port. The input signals are processed by DSK that is controlled by simulink model running in the computer system. And the filtered output of DSK is displayed on Digital Storage Oscilloscope (DSO) as shown in Figure 3.

Figure 2. Block diagram for DSP processor Implementation
Comparative Performance Analysis of Adaptive Algorithms for Simulation & Hardware Implementation of an ECG Signal

The Adaptive Noise Cancellation (ANC) Simulink model is designed using LMS and NLMS algorithms for generating C code and to download this code on DSP target processor. The ANC model is designed with help of inbuilt library of Simulink and the blocks are reconfigured as per the requirements of TMS320C6713 DSP processor [7, 10] as shown in figure 4.

IV. RESULTS & DISCUSSION

A. Simulation

The simulation of the LMS and NLMS algorithms is carried out with the following specifications:
Filter order N=19, step size µ= 0.009, iterations n= 1000, and c= 0.001
The algorithms are simulated at different noise variances (0.02, 0.05, and 0.1) and the output SNR is measured for each case. The pre SNR (noisy signal SNR) is subtracted from post SNR (filtered signal SNR) to find out the SNR improvement. The SNR improvement at different noise variances for LMS and NLMS algorithm is tabulated in Table-I. The detailed simulation results are presented in our previous work [7].
Table-1
Comparison of SNR Improvement

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Noise Variance</th>
<th>SNR Improved(LMS)</th>
<th>SNR Improved (NLMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0.02</td>
<td>9.47</td>
<td>11.02</td>
</tr>
<tr>
<td>2.</td>
<td>0.05</td>
<td>7.88</td>
<td>9.23</td>
</tr>
<tr>
<td>3.</td>
<td>0.1</td>
<td>6.15</td>
<td>8.37</td>
</tr>
</tbody>
</table>

From Table-1 it is clear that the performance of NLMS algorithm is better than LMS algorithms with an average 2.5dB additional SNR Improvement. It is also noticeable that the performance of algorithm is better in low noise environment.

B. Hardware Implementation

The real-time hardware implementation is also done to de-noise an ECG signal which is corrupted by various types of interferences & distortions like 50Hz power line interference. Figure 5 shows a clean ECG signal of 360mv, 35Hz which is used as a reference signal to calculate pre SNR.

In Figure 6(a) first signal shows a noise corrupted signal with the noise variance 0.02 and the second signal shows the filtered output for NLMS algorithm.
Comparative Performance Analysis of Adaptive Algorithms for Simulation & Hardware Implementation of an ECG Signal

Similarly Figure 6(b) and Figure 6(c) shows the filtered output for the noise corrupted single with the noise variance 0.05 and 0.1 respectively.
The SNR Improvement for filtered outputs at different noise variances for LMS and NLMS algorithms are presented in Table-2.

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Noise Variance</th>
<th>Sampling Rate (kHz)</th>
<th>SNR Improvement (dB) LMS</th>
<th>SNR Improvement (dB) NLMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0.02</td>
<td>1.5</td>
<td>6.26</td>
<td>7.46</td>
</tr>
<tr>
<td>2.</td>
<td>0.05</td>
<td>1.5</td>
<td>5.23</td>
<td>6.02</td>
</tr>
<tr>
<td>3.</td>
<td>0.1</td>
<td>1.5</td>
<td>4.73</td>
<td>5.22</td>
</tr>
</tbody>
</table>

From Table-2 it is observed that the performance of NLMS algorithm is still superior in hardware implementation also. But if we compare the results of Simulation (refer Table-1) & hardware Implementation (refer Table-2), we can say that the performance during simulation is very good but it does not have any practical significance, when we implement the same algorithm on hardware we got 67.7% efficiency in filtering as compared to simulation results. The efficiency may be improved by employing different algorithm and by varying the filter parameters this may be a good extension of this work. Figure 7 shows the performance comparison of NLMS & LMS algorithm when implemented on MATLAB simulator and TMS320C6713 hardware.
Comparative Performance Analysis of Adaptive Algorithms for Simulation & Hardware Implementation of an ECG Signal

V. CONCLUSIONS

The implementation of adaptive algorithms (LMS & NLMS) on MATLAB simulator and DSK TMS320C6713 hardware for an ECG signal has been done successfully and the results are compared in the terms of SNR Improvement. During the simulation for NLMS algorithm, we got the average SNR Improvement of 11.02dB for low noise signals, upto 8.37dB for highly noisy signals and during hardware implementation we got an average SNR Improvement of 7.46dB for low noise signals and 5.22 dB for highly noisy signals. Therefore we can conclude that the MATLAB simulation outcomes are implemented and verified successfully on DSP processor (TMS320C6713) with an average efficiency of 67.7%.

VI. REFERENCES