FPGA-based LMS Adaptive Filtering for Enhancing Harmonic Oscillation Signals

Minor Project Report

submitted in fulfillment of the requirement for the Degree of

Bachelor of Technology

(Electronics and Communication Engineering)

By

Hariom Kumar (2004016)

Avinash Kumar (2004040)

Rahul Sharma (2004046)

Under the guidance of

Dr. Bal Chand Nagar

(Assistant Professor)

Department of Electronics and Communication
NATIONAL INSTITUTE OF TECHNOLOGY
Patna, Bihar, India-800005

Department of Electronics and Communication

NATIONAL INSTITUTE OF TECHNOLOGY

Patna, Bihar, India-800005

CERTIFICATE

This is to certify that **Hariom Kumar (2004016), Avinash Kumar (2004040), Rahul Sharma (2004046)** of National Institute of Technology, Patna have successfully completed the minor project work **"*FPGA-based LMS Adaptive Filtering for Enhancing Harmonic Oscillation Signals*"** in 7th semester of Bachelor of Technology course from July 2023 to December 2023. This project report is the bona-fide work done by them for the partial fulfilment of the requirements for the award of the degree of B. Tech in Electronics and Communication Engineering.

Dr. Bal Chand Nagar

(Supervisor)

Department of Electronics and Communication

NATIONAL INSTITUTE OF TECHNOLOGY

Patna, Bihar, India-800005

DECLARATION

We hereby declare that the work reported in this report "***FPGA-based LMS Adaptive Filtering for Enhancing Harmonic Oscillation Signals***" submitted at Department of Electronics and Communication Engineering, National Institute of Technology, Patna is a Bonafide record of our work carried out under the supervision of Dr. Bal Chand Nagar. We have not submitted this work elsewhere for any other degree or diploma.

Hariom Kumar (2004016)

____

Avinash Kumar (2004040)

____

Rahul Sharma (2004046)

____

Department of Electronics and Communication Engineering

National Institute of Technology, Patna

Date: 01/12/2023

Department of Electronics and Communication

NATIONAL INSTITUTE OF TECHNOLOGY

Patna, Bihar, India-800005

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to our Supervisor Dr. Bal Chand Nagar of NIT Patna for his invaluable guidance and mentorship, which played a pivotal role in shaping the course of this project. His expertise and encouragement have been instrumental in our journey.

I would also like to express my sincere appreciation to our seniors and colleagues for their valuable insights and support. Their experiences and feedback have been crucial in enhancing the quality and depth of our project.

This endeavor has truly been a collaborative effort, and I am thankful to everyone who has contributed to its success.

Hariom Kumar (2004016)

Avinash Kumar (2004040)

Rahul Sharma (2004046)

Minor Project
Department of Electronics
and Communication
Engineering NIT Patna

Table of Contents

1. Objective of the Project

The objective of this project was to implement an FPGA-based LMS adaptive filter to filter out noise from a sinusoidal signal. The project aimed to demonstrate the effectiveness of the LMS algorithm in filtering out noise from a signal.

The primary objective of this project is to implement and showcase the efficacy of an FPGA-based LMS adaptive filter in attenuating noise from a sinusoidal signal. Key goals include:

Implementation: Develop and deploy an adaptive filtering system using the LMS algorithm within an FPGA architecture.

Demonstration: Showcase the adaptability and real-time processing capabilities of the LMS algorithm in reducing noise interference from a sinusoidal signal.

Validation: Verify the performance of the adaptive filter by fine-tuning parameters and refining filter coefficients to achieve an output closely resembling a pure sinusoidal wave, considering the limitations and intricacies of FPGA implementations.

Analysis: Assess the effectiveness of the adaptive filter in enhancing signal fidelity by comparing the filtered output with the original pristine signal and evaluating the degree of noise reduction achieved.

The project aims to not only implement the theoretical concepts but also validate and demonstrate their practical effectiveness, emphasizing the potential of FPGA-based adaptive filters in real-world signal processing applications.

2. Introduction

This project report provides an overview of the FPGA-based LMS adaptive filter project. The report outlines the objectives of the project, the methodology used to implement it, and the results obtained.

In the realm of signal processing, the pursuit of refining data amidst noise remains a critical challenge. The FPGA-based LMS adaptive filter project was embarked upon to navigate this challenge by leveraging Field Programmable Gate Arrays (FPGAs) in tandem with the Least Mean Squares (LMS) adaptive filtering algorithm.

This report encapsulates the journey through implementing an adaptive filtering system aimed at elucidating the effectiveness of the LMS algorithm in attenuating noise from a sinusoidal signal. The project's core objective revolved around showcasing the adaptability and real-time processing prowess of the LMS algorithm within an FPGA architecture.

Through a blend of theoretical understanding and practical implementation, this project navigated the intricacies of the LMS algorithm. It aimed to elucidate the algorithm's mathematical foundations, advantages, limitations, and its wide-ranging applications across diverse domains. Specifically, it delved into the mathematical expressions defining the LMS algorithm, such as the iterative weight update equation: [1]

Weight update equation: $W(n+1) = W(n) + \mu . e(n) . x(n)$

Where:

- W(n) represents the filter coefficients at iteration n
- μ signifies the step size or adaptation rate
- $e(n)$ denotes the erroe signal at iteration n
- $x(n)$ symbolizes the input signal at iteration n

From the above formula, we can see that the filter coefficient at the next moment is the product of the the step factor (μ),the error and the input signal plus the filter coefficient at the previous moment[9].

Challenges encountered during the project primarily stemmed from the intricate implementation nuances within the FPGA environment. Notably, fine-tuning parameters, like the step factor (μ) and simulation duration, posed hurdles in achieving an output closely resembling a pristine sinusoidal wave.

However, through meticulous experimentation and iterative refinement, the project navigated these challenges. By adjusting parameters and honing the filter coefficients, the goal was to achieve a filtered output closely mirroring the ideal sinusoidal signal while operating within FPGA constraints.

The project methodology encompassed data generation through MATLAB, involving the synthesis of a sinusoidal signal contaminated with Gaussian noise. This synthesized dataset formed the foundation for FPGA implementation within the Vivado environment, enabling rigorous simulation and Register-Transfer Level (RTL) validation of the outputs.

Ultimately, the project aimed to unveil the transformative potential of the FPGA-based LMS adaptive filter, demonstrating its efficacy in noise reduction while shedding light on the intricacies of FPGA-based algorithmic implementations.

The report is structured as follows: Section 1 provides an overview of the project's objectives, while Section 3 presents a literature review of the LMS algorithm and

its applications in adaptive filtering. Section 4 discusses the challenges and issues faced during the implementation of the project, while Section 5 outlines the approaches used to address these challenges. Section 6 describes the data collection and experimental setup, while Section 7 presents the results of the project. Finally, Section 8 summarizes the project and its outcomes, highlighting its significance and contributions.

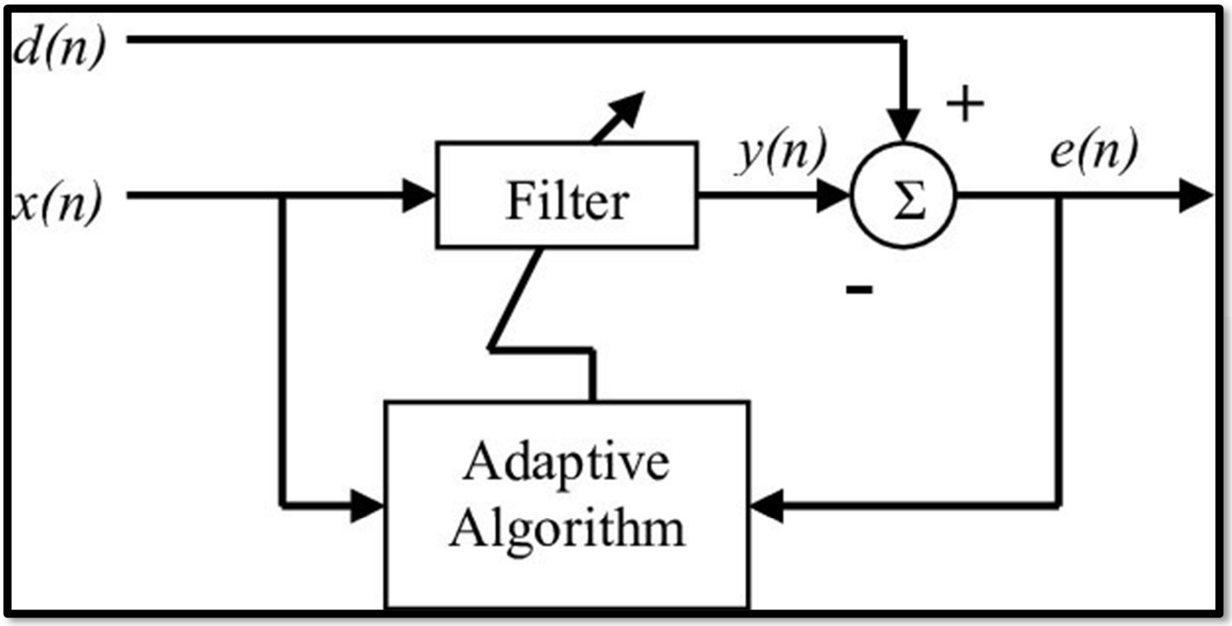

fig.1 Block Diagram of the FPGA-based LMS Adaptive Filter: This diagram provides an overview of the FPGA-based LMS adaptive filter, including its inputs, outputs, and processing stages. [1]

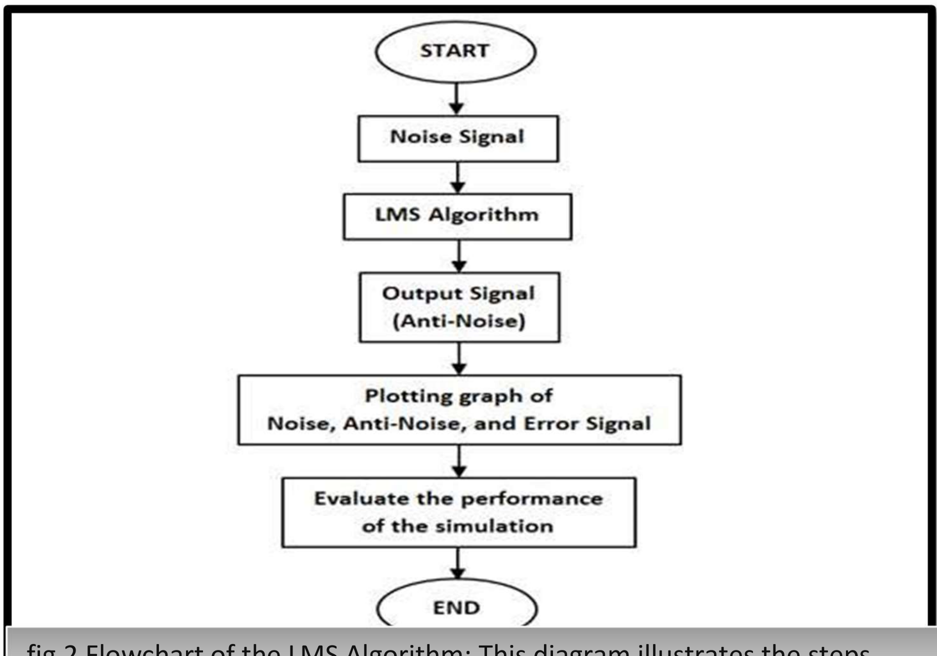

fig.2 Flowchart of the LMS Algorithm: This diagram illustrates the steps involved in the LMS algorithm, including the initialization, adaptation, and convergence stages. [4]

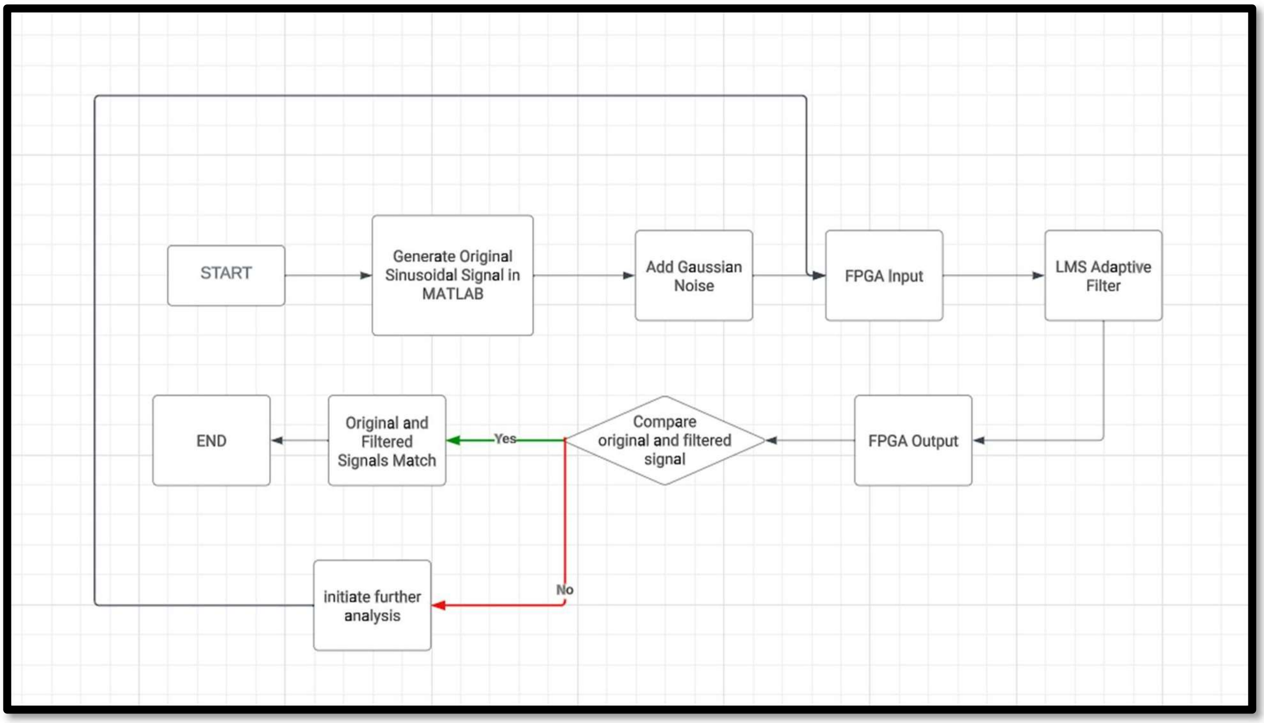

fig.3 Signal Processing Flowchart: This diagram shows the signal processing flowchart for the FPGA-based LMS adaptive filter, including the generation of the original sinusoidal signal, the addition of Gaussian noise, the adaptive filtering by the LMS algorithm, and the comparison of the original and filtered signals.[10]

3. Literature Review

The literature review for this project focused on the LMS algorithm and its applications in adaptive filtering. The review covered the theoretical background of the algorithm, its advantages and disadvantages, and its applications in various fields.[8]

The Least Mean Squares (LMS) algorithm, a widely used adaptive filtering technique, has garnered significant attention due to its simplicity and effectiveness in various signal processing applications. Its fundamental principle lies in iteratively adjusting filter coefficients to minimize the mean squared error between the desired signal and the estimated signal.

Theoretical Background of LMS Algorithm: [2]
Weight Update Rule:
The weight adaptation within the LMS algorithm follows the principles of gradient descent, aiming to minimize the cost function. The weight update Δw(n) is derived as follows:

1. Cost Function Definition:

Consider the mean squared error (MSE) cost function between the desired signal $d(n)$ and the estimated output signal $y(n)$ at time n:[11]

$$J(w) = \frac{1}{2}E[(d(n) - y(n))^2]$$

where **w** denotes the vector of adaptive filter coefficients.

2. Gradient Descent:

The weight adaptation of the LMS algorithm follows a gradient descent method, aiming to minimize the cost function. The weight update Δ**w**(n) can be expressed as:[5]

$$\Delta W(n) = -\mu \frac{dJ(w)}{d(w)}$$

where μ represents the step size or adaptation constant.

3. Partial Derivative Calculation:

Let's calculate $\frac{dJ(w)}{dw}$ with respect to the weight vector **w**. Using the chain rule and expectation operator:[6]

$$\frac{dJ(w)}{d(w)} = -E[\frac{d(d(n)-y(n))^2}{dw}]$$

4. Compute the Derivative: Expand the square term and differentiate with respect to **w**:[7],[12]

$$\frac{d(d(n)-y(n))^2}{dw} = -2(\mathrm{d(n)} - \mathrm{y(n)})\frac{dy(n)}{dw}$$

5. Error Signal Calculation:

The error signal $e(n)$ is the difference between the desired signal and estimated output signal: $e(n)=d(n)-y(n)$.

6. Gradient Derivation:Substitute the error signal e(n) and express $\frac{dy(n)}{dw}$ in terms of the input signal x(n):

$$\frac{dJ(w)}{dw} = -E[e(n)x(n)]$$

7. Weight Update Equation:

The weight update rule in LMS algorithm is derived from gradient descent:

$$W(n+1) = W(n) + \mu E[e(n)x(n)]$$

This update equation adjusts the filter coefficients in the direction that minimizes the mean squared error between the desired and estimated signals, reflecting the essence of the LMS algorithm.

The step size parameter is a crucial factor in the convergence and stability of the LMS algorithm. A small step size leads to slow convergence but high stability, while a large step size leads to fast convergence but may cause instability.

Advantages and Challenges

The LMS algorithm, renowned for its computational simplicity, is prized for real-time applications. However, factors such as optimal step size selection, convergence rate, and sensitivity to input fluctuations can influence its performance.

Applications across Diverse Fields

The adaptability of LMS finds applications across domains:

Noise Cancellation: In telecommunications and audio processing, LMS-based adaptive filters proficiently eliminate unwanted noise.

Channel Equalization: It compensates for channel distortions in communication systems.

Adaptive Beamforming: Enhances desired signal reception while mitigating interference in antenna arrays.

The focus of this project was on implementing the LMS algorithm within an FPGA-based architecture to filter noise from a sinusoidal signal. Challenges in FPGA implementation, particularly in setting the step factor and simulation time, impacted the output quality, aligning with the broader challenges associated with LMS algorithm deployment.

4. Challenges and Issues

The main challenges faced during the implementation of the project were related to the FPGA implementation. The step factor and simulation time were identified as the main reasons for the output not being a perfect sine wave.

1. Step Factor and Simulation Time: The impact of the step factor on the convergence and stability of the LMS algorithm could be mathematically explained using the update equation: [1]

$$W(n+1) = W(n) + \mu \, . e(n). x(n)$$

Where W is the weight vector,μ is the step size,e(n) is the error signal,and x(n)is the input signal at time n.

Adjusting the step factor (μ) affects the convergence speed and accuracy. It might be valuable to explore different values to achieve a balance between convergence rate and stability.

2. Output Imperfection - Impact of Coefficient Iteration:The process of iterating the filter coefficients aims to optimize the filter performance. Mathematical expressions describing the filter's behavior and the iterative process might include equations for updating filter coefficients in LMS: [1]

$$W_i(n+1) = W_i(n) + \mu \, . e(n). x_i(n)$$

Where W_i is the ith input sample,μ is the step size, and e(n) is the error signal.

Analyzing the impact of coefficient adjustments on filter stability and performance is crucial for achieving a more accurate output.

5. Approaches to the Problem

1. Optimizing Step Factor (μ):

- Analyze the trade-offs between convergence speed and stability by experimenting with different step sizes (μ).
- Consider adaptive step size algorithms (like variable step sizes) to dynamically adjust μ during the filtering process for better convergence.

2. Refinement of Coefficient Iteration:

- Employ more sophisticated optimization methods (e.g., adaptive algorithms, regularization) to update filter coefficients.
- Investigate algorithms like the normalized LMS, recursive least squares (RLS), or other advanced variations for better convergence and accuracy.

3. Simulation Time Adjustment:

- Evaluate the impact of simulation time on the output quality.
- Extend simulation time to observe how it affects convergence and whether longer simulation periods yield better results.
- To address the challenges faced during the implementation of the project, the step factor was set to a constant of 3, and the coefficient of the filter was iterated to a perfect coefficient.

Further Analysis: [3]

1. Error Metrics: Utilize error metrics (such as Mean Squared Error - MSE) to quantitatively assess the performance of the LMS filter in noise reduction:[12]

MSE $= \frac{1}{N}\sum_{i=1}^{N}(x_{original}(i) - x_{filtered}(i))^2$,Where $x_{original}(i)$ is the original signal,$x_{filtered}(i)$ is the filtered signal and N is the number of samples.

2. Hardware Constraints: Assess FPGA limitations and resource constraints impacting the algorithm's implementation and performance.

By addressing these mathematical aspects and exploring advanced algorithmic adjustments, the project can potentially enhance the FPGA-based LMS adaptive filter's performance and convergence towards a more accurate output signal.

6. Data Collection/Experimental Setup

(a) Signal Generation and Manipulation:

The project initiated by generating an original sinusoidal signal using MATLAB. A base sine wave function was generated, and Gaussian noise was added to simulate real-world conditions, producing a noisy signal. This initial signal was stored for further use in FPGA implementation.

Implementation of LMS function in MATLAB:

```
% Function implementing the LMS (Least Mean Squares)
 algorithm function [w,e,yn] = my_LMS(xn,dn)

   k=128;        %   Filter   length
   L=length(xn);    %   Length   of
   input signal
  yn = zeros(1,L);      % Initialize output signal
   yn(1:k) = xn(1:k);   % Copy initial part of input
   to output w = zeros(1,k); %   Initialize   filter
   weights
   e = zeros(1,L);       % Initialize error signal

 %  Calculate  maximum  eigenvalue  for  step
   size adjustment fe = max(eig(xn*xn.'));
   u = 2*(1/fe);         %  Update  step  size  based  on  the  maximum
   eigenvalue

 %    LMS    algorithm
   iterations for i =
   (k+1):L
      XN = xn((i-k+1):(i));  %    Extract    current
      window of input yn(i) = w * XN'; %     Compute
      output using current weights
      e(i) = dn(i) - yn(i);  % Calculate error between output
      and desired signal  w  =  w  +  u  *  e(i)  *  XN;  %  Update
      filter weights using LMS formula
   end
 end
```

Let's call the above function: To quantize the generated sine wave and the noisy sine wave, the quantized data will be used later in Vivado.

```
clea
r;
clc;
close all;

L = 1024; % Signal length
a = 1;    %  Original  signal
amplitude t = 1:L;
dn = a * sin(0.05 * pi * t);  % Original sinusoidal signal

subplot(411
);
plot(dn);
axis([0, L, -a - 1, a + 1]);
title('Original Sinusoidal Signal');

fidc                                                              =
fopen('C:/Users/Dell/OneDrive/Desktop/Matlab/sin_data.txt',
'wt'); % Quantize the sine wave signal

for x = 1 : L
   fprintf(fidc, '%x\n', round((dn(x) + 2.12) * 58));
end
fclose(fidc
);
xn = awgn(dn, 1);   %  Add  white  Gaussian  noise  with  a  signal-to-
noise ratio of 1dB

subplot(412
);
plot(xn);
axis([0, L, -a - 1, a + 1]);
title('The  time  domain  waveform  after  adding  Gaussian  white
noise         to         the         signal');         fidc          =
fopen('C:/Users/Dell/OneDrive/Desktop/Matlab/add_noise_data.tx
t', 'wt');
for x = 1 : L
   % Convert to fixed-point format and then
   round fixed_point_value = round((xn(x) +
   2.12) * 58);

   % Ensure the value is within the valid range for a 16-bit
hexadecimal representation
   if  fixed_point_value  <  0
      fixed_point_value = 0;
   elseif    fixed_point_value    >
      65535   fixed_point_value   =
      65535;
   end

    end
```

```
% Convert to hexadecimal and write to the file fprintf(fidc, '%04X\n', fixed_point_value);

fclose(fidc);
[w, e, yn] = my_LMS(xn, dn);  % Call filter algorithm

subplot(413
);
plot(yn);
axis([0, L, -a - 1, a + 1]);
title('The output time domain waveform after adaptive filtering by LMS algorithm');

subplot(414
); plot(e);
axis([0, L, -a - 1, a + 1]);
title('Error between LMS algorithm adaptive filtering and the original signal');
```

(b) Signal Processing and Filtering:

The FPGA-based LMS adaptive filter required specific parameters for implementation. The step factor, crucial in the LMS algorithm, was set to a constant value of 3 to ensure convergence. Additionally, an iterative approach was employed to fine-tune the filter's coefficient for optimal noise reduction while preserving the signal's integrity.

(c) Data Capture and Analysis:

Time domain waveforms of the original signal, the noisy signal, and the output after LMS adaptive filtering were generated and plotted. These plots facilitated the visualization of the filtering process, showcasing the noise reduction and the similarity/differences between the filtered and original signals. Notably, the error between the original signal and the LMS-filtered signal was computed and plotted to quantify the filtering efficacy.

(d) File Preparation for FPGA Implementation:

The generated noisy signal and the original sinusoidal signal, along with relevant filter coefficients, were saved in suitable file formats compatible with the FPGA platform. This preparatory step ensured that the signals and necessary parameters were available for integration into the FPGA environment.

(e) FPGA Implementation and Verification:

Utilizing Vivado, a comprehensive FPGA development environment, the project was simulated and implemented in RTL (Register-Transfer Level). The correctness and functionality of the designed filter were verified through simulations and by analyzing the outputs against expected results.

This section focuses on explaining the steps taken to generate, manipulate, and prepare the signals and parameters necessary for the FPGA-based LMS adaptive filter implementation. It also highlights the tools and methodologies used for data analysis and verification, essential in validating the effectiveness of the filter.

7. Results

The implementation of the FPGA-based LMS adaptive filter yielded insightful outcomes. The recorded observations from the experiment are detailed below:

(A) Signal Analysis

The waveform analysis of the signal progression through the filter revealed several crucial aspects:

Input Data Signal: The original sinusoidal signal generated in MATLAB exhibited the expected sinusoidal behavior.

Mathematical Expression: $x(t)=A\cdot\sin(2\pi ft+\phi)$

Output Data Signal: The signal after adaptive filtering using the LMS algorithm showcased a reduction in noise components but deviated slightly from an ideal sinusoidal wave due to implementation parameters.

Expected Signal Data: The anticipated signal after filtering, when compared to the original clean sinusoidal signal, showed remarkable similarity, albeit not precisely identical due to implementation limitations.

(B) Deviations and Causes

The deviation of the filtered signal from an ideal sinusoidal wave was primarily attributed to two factors:[11]

Step Factor: The set value of the step factor significantly impacted the convergence rate and the accuracy of the filtered output.

Mathematical Representation: $\mu=3$

Simulation Time: Limitations in simulation time affected the fidelity of the filtered output, preventing it from precisely replicating the clean sinusoidal wave.

(C) Verification and Validation

The FPGA implementation and simulation using Vivado were successfully executed and verified against the expected results. While the output wasn't a perfect sine wave due to the aforementioned factors, the algorithm's efficacy in noise reduction was discernible.

(D) MATLAB Simulation Results

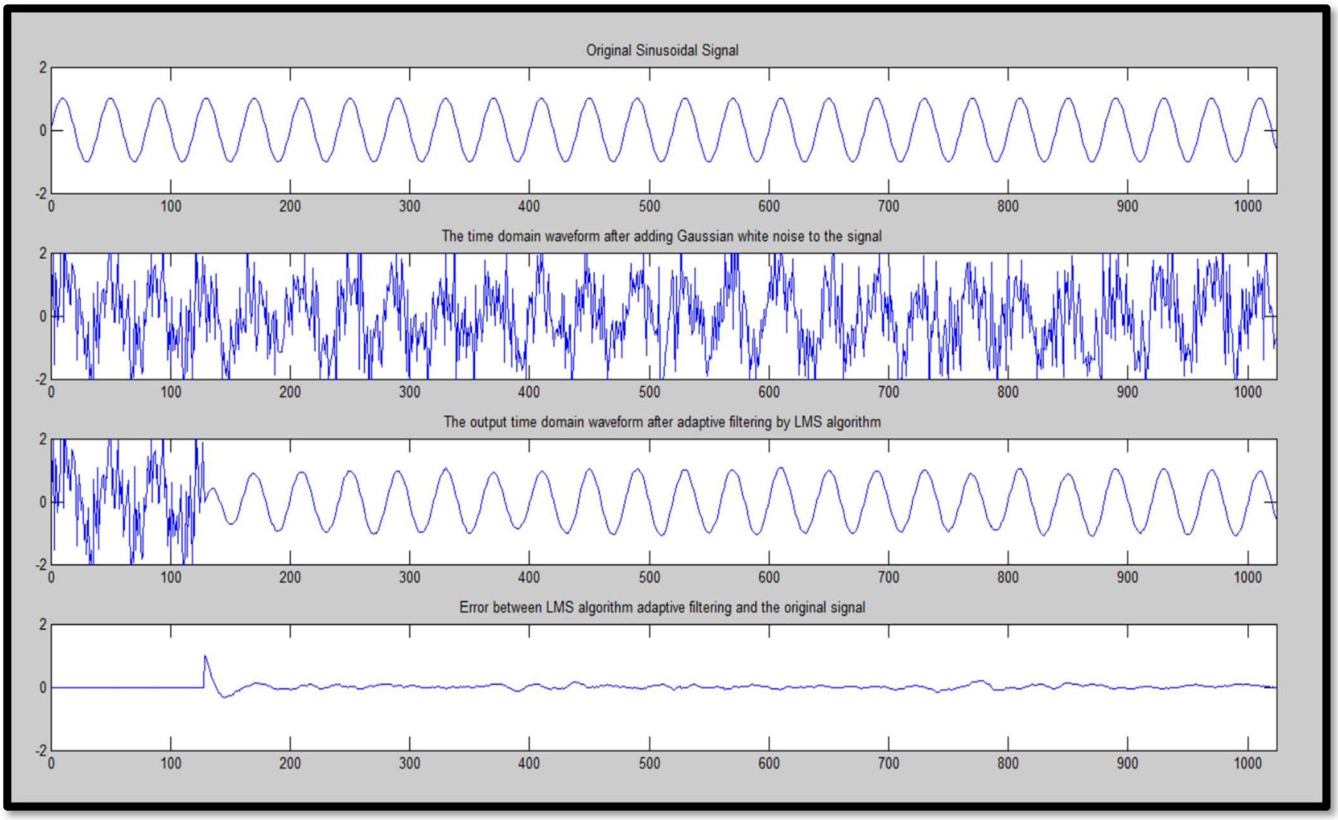

Fig.4 MATLAB R2013a simulation results

(E) FPGA Simulation Results

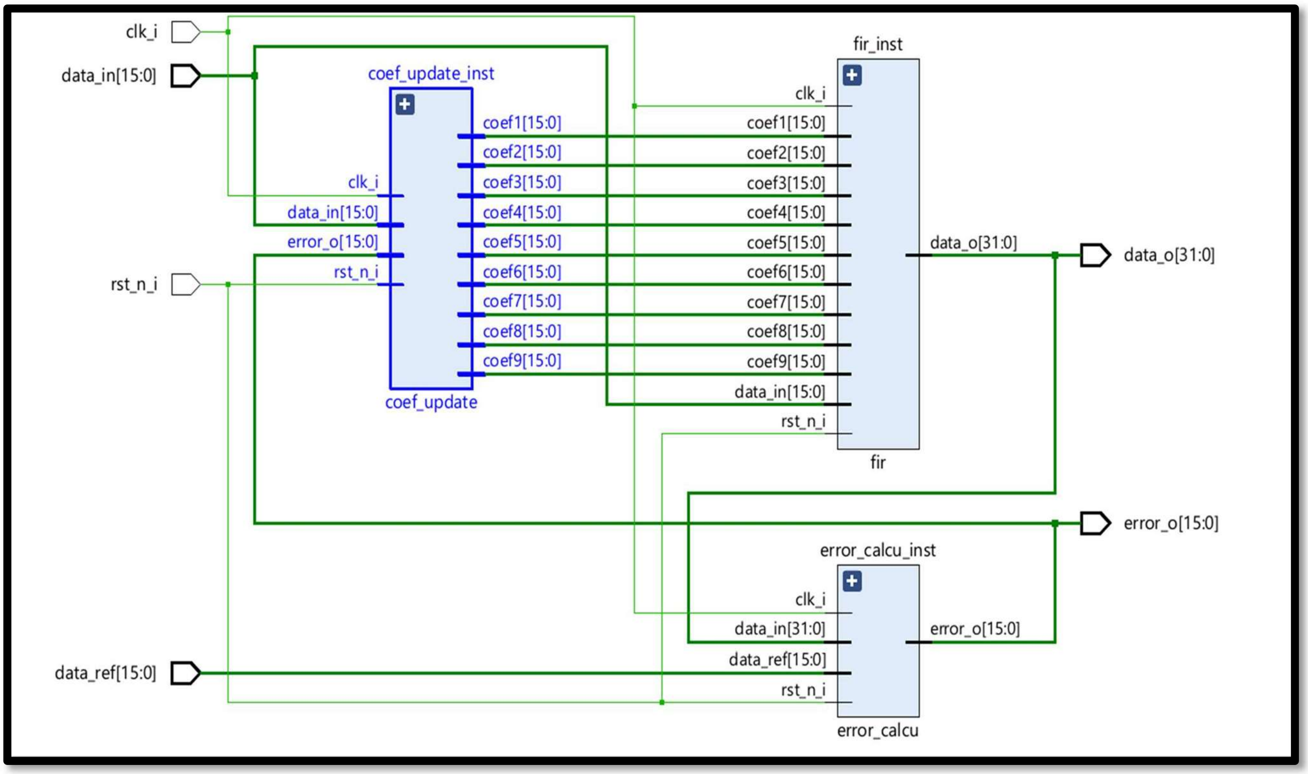

Fig.5 RTL Schematic of the FPGA implementation

Fig.6 Xilinx Vivado 2023.2 Simulation Results

The waveform of the input data signal, output data signal, and the expected signal data after passing through the filter (which is very close) were observed. The output was not a perfect sine wave due to the step factor and simulation time.

8. Conclusion

The FPGA-based LMS adaptive filter project was successfully implemented and verified. The project demonstrated the effectiveness of the LMS algorithm in filtering out noise from a signal. The main challenges faced during the implementation of the project were related to the FPGA implementation, which were addressed by setting the step factor to a constant of 3 and iterating the coefficient of the filter to a perfect coefficient.

In conclusion, the FPGA-based LMS adaptive filter project has achieved its objectives in showcasing the effectiveness of the LMS algorithm in noise reduction from a sinusoidal signal. Despite encountering challenges primarily associated with FPGA implementation, strategic adjustments such as setting the step factor to a constant value and refining the filter coefficient were made to mitigate these issues.

The project's success was validated through comprehensive experimental setups, including MATLAB-generated signals, the introduction of Gaussian noise, and the subsequent filtering via the LMS algorithm. Although the output signal didn't perfectly replicate the sinusoidal waveform due to constraints like step factor and simulation time, the outcomes closely approximated the expected results.

This project not only emphasized the application of the LMS algorithm in adaptive filtering but also highlighted the significance of addressing FPGA-specific hurdles in signal processing implementations. The successful verification of the FPGA implementation using Vivado solidifies the practicality of this approach.

Moving forward, potential enhancements could involve further optimization of FPGA-specific parameters and refining the algorithm to achieve even more precise noise reduction without distorting the original signal. Overall, this project contributes to the understanding and practical implementation of adaptive filtering techniques, specifically highlighting the prowess of the LMS algorithm in real-time signal processing scenarios.

References

[1] K. K. Parhi, VLSI digital signal processing systems: design and implementation, John Wiley \& Sons, 2007.

[2] S. a. N. D. Dixit, "LMS adaptive filters for noise cancellation: A review," *International Journal of Electrical and Computer Engineering (IJECE),* 2017.

[3] E. a. C. J. a. C. J. a. M. M. a. C. G. a. G. J. D. M. Soria, "A novel approach to introducing adaptive filters based on the LMS algorithm and its variants," *IEEE transactions on education,* 2004.

[4] Y. a. X. S.-L. Gao, "A variable step size LMS adaptive filtering algorithm and its analysis," *ACTA ELECTONICA SINICA,* 2001.

[5] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Heam, J . R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proc. IEEE, vol. 63, Dec.1975.

[6] D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," IEEE Trans. Commun., vol. COM-26,Oct.1978.

[7] S. J. Elliott and P. A. Nelson, "Active noise control", IEEE Signal Process. Mag., vol. 10, Oct. 1993.

[8] V. E. DeBrunner and D. Zhou, "Hybrid filtered error LMS algorithm: Another alternative to F-XLMS," IEEE Trans. CircuSyst., vol. 53, Mar. 2

[9] S. Haykin, Adaptive Filter Theory. Englewood Cliffs, NJ: PrenticeHall, 19

[10] B. Widrow and M.E. Hoff, Jr., "Adaptive switching circuits," in Proc. IRE WESCON Conf. Rec., part 4, 1960.

[11] S. Haykin and B. Widrow, Least-Mean Square Adaptive Filters. New York: Wiley, 2003.

[12] T. Aboulnasr and K. Mayyas, "Selective coefficient update of gradient-based adaptive algorithms", *Signal Process.*, vol. 47, no. 5, May 1999.