RES MILITARIS
Social Science Journal

OPTIMIZED ADAPTIVE NOISECANCELLATION ON FPGA

Umavathi.M.
Department of Electrical and Electronics Engineering
B.M.S College of Engineering Bengaluru, India
umavathim.eee@bmsce.ac.in

Vinayak Mallikarjun Kolavi
Department of Electronics and Communication Engineering BMS College of Engineering Bengaluru, India
vinayakmk.ec21@bmsce.ac.in

Vishrut Sateesh Mokashi
Department of Electronics and Communication Engineering BMS College of Engineering Bengaluru, India
vishrut.ec21@bmsce.ac.in

Vishnu Prakash Bharadwaj
Department of Electronics and Communication Engineering BMS College of Engineering Bengaluru, India
vishnu.ec21@bmsce.ac.in

Mrs Ashwini V
Department of Electronics and Communication Engineering BMS College of Engineering
Bengaluru, India ashwiniv.ece@bmsce.ac.in

Vamshi B
Department of Electronics and Communication Engineering BMS College of Engineering
Bengaluru, India
vamshi.ec21@bmsce.ac.in

***Abstract*—. External sounds are omnipresent, particularly incritical environments like vehicles and aircraft, where the imperative is to minimize external noise impacting the system. To accomplish this, we've employed an adaptive filter utilizing the fast-LMS algorithm. Additionally, we've interfaced the FPGA with hardware peripherals, for capturing and playing back the reduced noise, respectively. The audio codec on the FPGA is leveraged to process the sound signals in our implementation.**

Keywords—FPGA, ANC, LMS, ADC, DAC

1. INTRODUCTION

Noise cancellation requires quick and efficient signal processing to function effectively in real-time. Our project revolves around the real-time implementation of an adaptive noise canceller designed to effectively mitigate environmental noise. The omnipresence of external sound poses a significant challenge, especially in environments like vehicles and aircraft, where minimizing external noise is crucial. To address this, we've employed an adaptive filter based on optimized algorithms and interfaced it with FPGA hardware. The FPGA's built-in audio codec is leveraged to process the sound signals, enhancing the efficiency of our adaptive noise cancellation system. This project is geared towards providing practical solutions for minimizing unwanted external noise in dynamic environments.

Implementing noise cancellation on a Field- Programmable Gate Array (FPGA) offers several advantages, making it a compelling choice for various applications. FPGAs excel in real-time processing due to their inherent capability for parallel processing. This feature is crucial for noise cancellation systems that demand rapid adaptation to changing environmental conditions. The parallel processing capabilities of FPGAs enable the simultaneous execution of complex algorithms, allowing for efficient and real-time implementation.

In addition to real-time processing, FPGAs provide low- latency capabilities, making them suitable for applications where minimizing the delay between noise detection and cancellation is essential. This is particularly beneficial in live audio processing scenarios. The customization potential of FPGAs is a key advantage. These devices allow for the tailoring of hardware architectures to match the specific requirements of the noise cancellation algorithm. This flexibility is especially advantageous when dealing with unique or proprietary algorithms, providing the ability to optimize performance for specific applications.

FPGAs offer dedicated digital signal processing (DSP) blocks, which can be efficiently utilized for implementing filtering operations. This results in resource-efficient designs, a crucial factor in real-time applications where efficient resource usage is paramount for optimal performance. The parallel processing capabilities of FPGAs can be further exploited for noise cancellation by simultaneously executingmultiple tasks. This can include processing multiple input channels or implementing parallel adaptive filters to enhance cancellation performance. Adaptability is a critical aspect of noise cancellation systems, which often require adaptive algorithms to continuously adjust to changing noise characteristics. FPGAs can be reconfigured on the fly, allowing for dynamic adaptation and optimization of algorithm parameters based on the real-time noise environment.

FPGAs can be power-optimized for specific tasks, strikinga balance between performance and energy efficiency. This isparticularly advantageous in applications where powerconsumption is a critical consideration, contributing to the overall efficiency of the noise cancellation system. FPGAs facilitate seamless digital integration by easily interfacing with other digital components, such as analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). This integration capability enhances the overall performance and functionality of the noise

cancellation system within digital signal processing systems. Scalability is another noteworthy feature of FPGAs. They offer the flexibility to implement noise cancellation systems with varying degrees of complexity, making them suitable for a wide range of applications. This scalability allows for the adaptation of noise cancellation techniques from simple reductions in consumer electronics to advanced implementations in industrial settings.

FPGAs provide an excellent platform for prototyping and development. Designers can iterate quickly, experiment withdifferent algorithms, and optimize performance beforecommitting to a final implementation. This accelerates the development process and ensures a more robust and tailored noise cancellation system. Integration of noise cancellation on an FPGA can often lead to a reduction in the overall component count in a system. This can simplify the design and potentially reduce costs, making FPGAs an economically viable solution for noise cancellation implementations.

2. LITERATURE SURVEY

A. Adaptive Noise Cancellation

Adaptive Noise Cancellation (ANC) is a fascinating fieldthat addresses the challenge of minimizing or eliminating unwanted ambient noise during audio transmission. In an idealcommunication scenario, the transmitted message should be reproduced faithfully at the destination without any additionalinterference. However, achieving such an ideal setup is challenging due to the omnipresence of environmental noise [2].

In the context of this project, the focus is on audio transmission, and the technique chosen is Noise Cancellation,specifically Adaptive Noise Cancellation. This method aims to provide optimal audio signal transmission by minimizing data loss or disruption caused by extraneous noise. Since noise is inherent in the environment, noise cancellation becomes crucial for maintaining the integrity of the transmitted audio signal.

Adaptive Noise Cancellation involves the generationof anti-noise [4], which theoretically cancels out ambient noise. In this specific Noise Cancellation setup, a singlemicrophone is used, along with audio output devices such as speakers or headphones. The system is modeled such that themicrophone captures the environment's ambient noise, serving as the input to the system. The system then generates anti- noise, which is superimposed on the input noise signal, leading to partial cancellation of the noise. The result is a reduced noise signal at the output, contributing to improved audio signal transmission quality.

B. LMS Algorithm

In our project, we opted for the Least Mean Square (LMS)Algorithm due to its computational simplicity and ease of implementation. This algorithm is based on the concepts of Least Mean Squares and the Steepest Descent Algorithm [1].Notably, it streamlines the process by eliminating the requirement for precise measurements of the gradient vector and avoids the complexity of matrix inversion. The LMS algorithm belongs to the adaptive filter class, aiming to replicate a desired filter by dynamically adjusting filter coefficients. This adjustment is orchestrated to minimize the least mean square of the error signal. In essence, if we conceptualize the error as a cost function, the LMS algorithm seeks filter coefficient values that minimize this cost function. The error signal, in this context, is the disparity between the desired and actual signals.

Our project implements an adaptive filter using an optimized variant of the LMS algorithm. This optimization facilitates real-time updates to the filter weights while concurrently reducing the computational load on the FPGA. The focus on efficiency ensures that our adaptive noise cancellation system performs effectively, addressing the dynamic nature of the real-time noise reduction process.

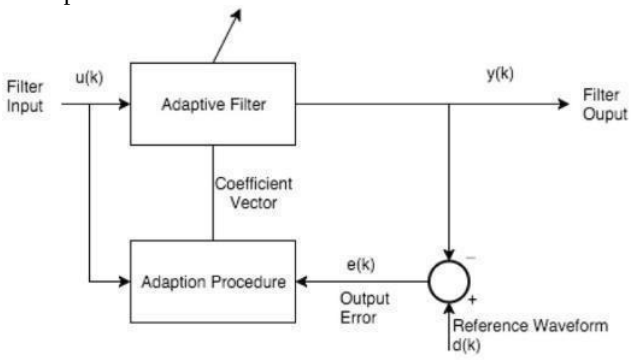

Fig. 1

The signals illustrated in Fig. 1 play distinct roles in thefunctionality of our adaptive noise cancellation system:

i. u(k): This represents the input vector, essentially a purenoise signal introduced into the system.

ii. d(k): The reference signal, depicting the contaminatedinput signal that needs noise reduction.

iii. y(k): The output of the adaptive filter, providing the anti-noise or reduced noise signal.

iv. e(k): The estimated error, obtained by subtracting the anti-noise from the contaminated signal. This serves as the final reduced-noise output of the system.

The LMS algorithm, governing the adaptive filter'soperation, follows these equations [3]:

i. Filter Output: y(k) = uT W

W: Vector of weights applied to the filter coefficientsT: Transpose operation

ii. Estimated Error: e(k) = d(k) - y(k)

iii. Weight Update: Wk+1 = Wk+2 × μ × e(k) × μk

The adaptive procedure incorporates the estimated errorand the input noise vector as feedback.

The step size, denoted by μ, is crucial in determining the convergence behavior. It is inversely proportional to the settling time constant. Smaller step sizes result in a slower adaptive process but minimize the mean-square error, leading to better results over an extended convergence period. Conversely, larger step sizes accelerate the adaptive process, but the achieved results may not be as optimal as those obtained with smaller step sizes. This trade-off between convergence speed and final performance is a critical consideration in configuring the LMS algorithm for effectiveadaptive noise cancellation.

C. Audio CoDec

Building an interface to the audio codec on the FPGA is essential for recording and playing audio, facilitating bidirectional communication with hardware peripherals like microphones and speakers.

When audio travels through the air as pressure waves, the microphone captures these waves by detecting areas of compression and rarefaction. It translates the pressure variations into an electrical signal, using voltage levels to represent points in the wave. However, these voltage levels are continuous in time, creating analog signals [5]. Since the FPGA operates on digital quantities at a fixed clock rate, it cannot process analog signals directly. To bridge this gap, the audio codec on the FPGA intervenes. It converts the analog voltage values to discrete quantities by sampling the voltage levels at regular intervals based on a predefined sampling rate. This process involves Analog-to-Digital Conversion (ADC), discretizing the continuous analog signal into a digital formatthat the FPGA can handle.

Conversely, when playing an audio signal from the FPGA through speakers, the codec executes Digital-to-Analog Conversion (DAC). This process involves converting thediscrete digital values back into analog voltage levels, reconstructing a continuous signal. However, it's important to note that this reconstruction may result in a slight loss of information due to the discrete nature of the original digital values.

3. Problem Analysis & Solution

A. Problem Definition

Employing real-time noise cancellation of sound on FPGA can pose a variety of challenges:

- Delay due to processing: Processing with the LMS algorithm can create a noticeable delay between the noisy input and clean output of sound.
- Integrating audio CoDec: Microphone and speaker/headphones have to be working in sync with the FPGA.
- Floating Point Coefficients: The filter coefficients are precisely described in floating point that the FPGA has todeal with.

B. Proposed Solution

- Delay due to processing: In our implementation of the adaptive filter, we employ the Fast-LMS Algorithm, characterized by the following weight updating equation:

Wk+1 = Wk + e(k) × sign(u(k)) >> n

In the Fast-LMS Algorithm, a notable departure from traditional methods is the replacement of the step size with a shift operation. Here, n denotes the number of shifts. Unlike conventional approaches, Fast-LMS exclusively utilizes the sign bit of the reference input u(k), disregarding the actual value of the input. This adaptation streamlines the computation process, reducing the computational load and enhancing the algorithm's efficiency for real-time applications.

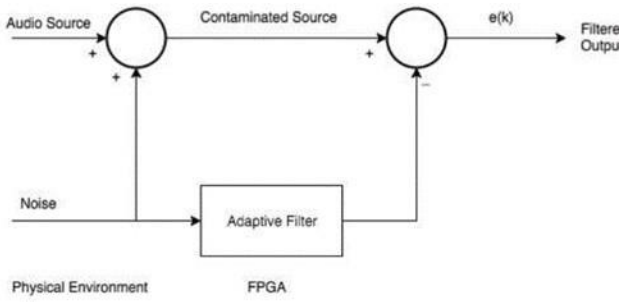

Fig. 2

The essence of Fast-LMS lies in its ability to updatefilter weights with reduced computational complexity, making it well-suited for scenarios such as real-time applications in adaptive noise cancellation systems.

- Integrating audio CoDec: We can make use of the I2C communication protocol to synchronize the peripheral hardware with the FPGA.
- Floating Point Coefficients: We can consider weights asfixed-point arithmetic values by introduction of slight loss of data.

4. METHODOLOGY & IMPLEMENTATION

A. Block Diagram

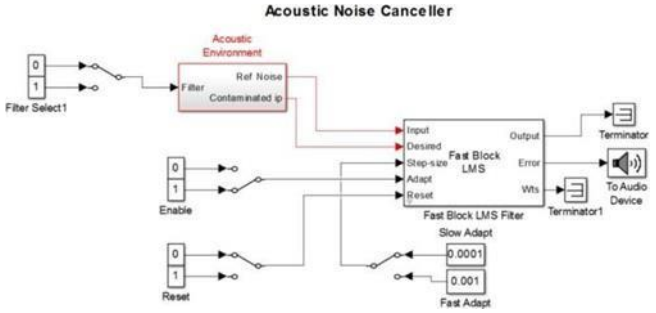

Fig.3

B. Pictorial Representation

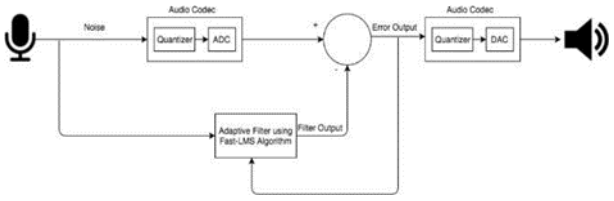

Fig.4

Opting for Fast-LMS over LMS in our project was a strategic choice because Fast-LMS relies solely on the sign bit of the input in its weight update formula. This characteristic significantly decreases the number of required multiplications, streamlining the implementation of the LMSfilter in hardware. Furthermore, Fast-LMS simplifies the algorithm by replacing the step size with a shift operation.

To accommodate our system's noise input, recorded through a microphone, we've adjusted the block diagram of Adaptive Noise Cancellation. The Adaptive Filter receives thenoisy input and produces a reduced noise, represented as the Filter Output in the diagram. The next step involves subtracting this Filter Output from the Noisy Input to generatethe error output, which represents the reduced noise signal. This error output is then fed back to the adaptive filter, facilitating the updating of filter coefficient weights. Finally, the reduced error output is transmitted through a speaker.

C. Structure

- *Overview of Artix 7 Basys 3 FPGA:* The Artix-7 FPGAis designed for optimal high-performance logic and boasts increased capacity, superior performance, and more resources compared to earlier designs. Key features of the Artix-7 35T include:

- 33,280 logic cells distributed across 5200 slices, where each slice incorporates four 6-input LUTs and 8 flip- flops.
- 1,800 Kbits of fast block RAM.
- Five clock management tiles, each equipped with aphase-locked loop (PLL).
- 90 DSP slices.
- Internal clock speeds surpassing 450MHz.
- On-chip analog-to-digital converter (XADC)

The Basys 3 further enhances its capabilities with animproved array of ports and peripherals, encompassing:

- 16 user switches.
- 16 user LEDs.
- 5 user pushbuttons.
- 4-digit 7-segment display.
- Three Pmod ports.
- Pmod for XADC signals.
- 12-bit VGA output.
- USB-UART Bridge.
- Serial Flash.
- Digilent USB-JTAG port for FPGA programming andcommunication.
- USB HID Host, accommodating mice, keyboards, andmemory sticks.

a. Vivado Software Overview: Vivado stands as a comprehensive software suite crafted by Xilinx, aimed at the design and programming of Field-Programmable Gate Arrays (FPGAs), System on Chips (SoCs), and programmable devices. It furnishes an integrated development environment (IDE) equipped with tools spanning various stages of the FPGA design process, encompassing design, verification, synthesis, implementation, and debugging. Here's a concise overview of Vivado:

- Design Entry: Vivado provides diverse methods for entering and describing designs. Users can employ schematics, Hardware Description Languages (HDLs) like Verilog or VHDL, or high-level synthesis (HLS) languages such as C/C++ or MATLAB/Simulink to articulate the desired functionality.
- Synthesis and Optimization: Vivado's synthesis tool translates the design description into a netlistrepresenting the logic circuit. It conducts optimizations to enhance performance, minimize resource usage, and adhere to timing constraints. Users can specify design constraints, such as clock frequencies, input/output requirements, and timing constraints to guide synthesis.
- Implementation: Following synthesis, Vivado engages inplace-and-route, mapping the synthesized logic onto the FPGA's resources and routing interconnections betweenthem. It optimizes placement to meet timing requirements and minimize resource utilization. Vivado supports various FPGA architectures from Xilinx, including Artix, Kintex, Virtex, and Zynq families.
- Debugging and Validation: Vivado offers advanced debugging features for identifying and resolving design issues. Its integrated logic analyzer and debugging interfaces enable users to probe signals, set breakpoints, and analyze waveforms to debug designs running on theFPGA.
- Programming and Configuration: Once the design is validated, Vivado generates configuration files (bitstreams) to program the FPGA and implement the desired functionality. These bitstreams can be downloaded onto the FPGA using Xilinx programming tools such as Vivado Hardware Manager.
- In summary, Vivado stands out as a potent and versatile software suite that facilitates the entire FPGA design process, from conceptualization to final implementation and debugging. Its comprehensive features, advanced optimization algorithms, and integration capabilities make it a preferred choice among FPGA designers for developing intricate and high-performance digital systems.

D. Flow of Program

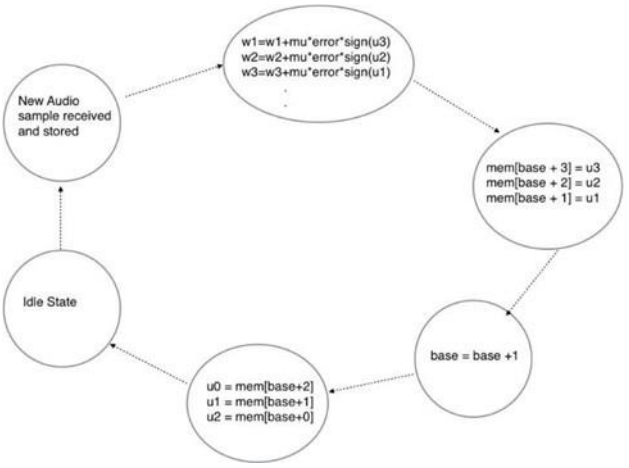

Fig.5

A state machine was devised to manage the various sequences of processes involved, consisting of six states. The transitions in the state machine occur on the positive edge ofthe main clock, operating at 100 MHz. However, the triggering of the state machine is synchronized with thepositive edge of the DAC clock (the Idle state awaits the positive level of the DAC clock).

The six states are as follows:

1. First State: Awaits noise input sample.
2. Second State: Computes filter weights.
3. Third State: Stores noise input in a circular buffer.
4. Fourth State: Increments the base address pointer of thecircular buffer.
5. Fifth State: Restores the previous history of noise samplesfrom the circular buffer for each tap.
6. Sixth State: Idle State. Error values are calculated through continuous assignment statements, instantly computing values whenever any of their inputs change. The output values for the DAC are written at the negative edge of the DAC clock, ensuring that the data isready for pushing to the DAC in the audio codec during the subsequent clock cycle.

4. RESULTS & DISCUSSION

Noisy input is taken from a source, say a sample signal on cell phone, that is played near the microphone. The microphone converts the sound signal to electrical signals. The ADC interface converts the electrical signal to digital data. The FPGA processes the input to adjust filter coefficients. The data is then passed through adjusted coefficient filter. The output of the filter is interfaced with anDAC. The output of DAC is connected to a speaker. The input sound is heard from the speaker without the noise, withas littledelay as possible between providing the input sound at microphone and obtaining the output sound at the speaker. The power utilization of the chip is summarized as follows inFig.6.

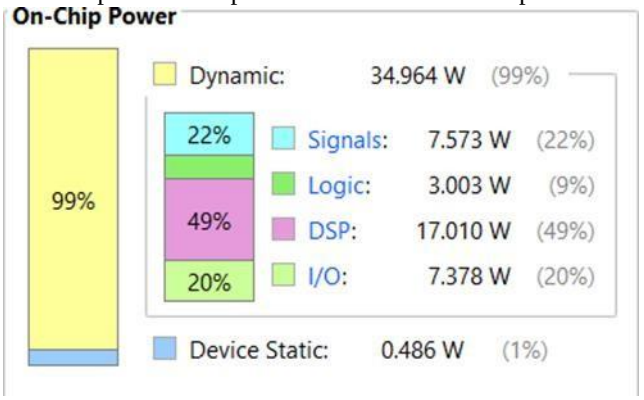

Fig.6

The cell usage report as produced by Vivado is as given inFig.7. We can observe DSP blocks being used.

Report Cell Usage:

	Cell	Count
1	BUFG	1
2	CARRY4	36
3	DSP48E1	17
7	LUT2	400
8	LUT3	4
9	LUT4	7
10	FDCE	3
11	FDRE	272
12	IBUF	34
13	OBUF	16

Fig.7

The design elaborated using the FPGA cells is depicted inFig.8.

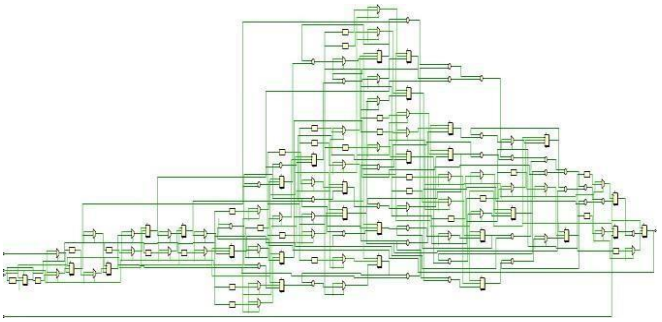

Fig.8

5. FUTURE TRENDS AND CONCLUSION

A. Conclusion

In conclusion, our project focuses on the intricate domain of Adaptive Noise Cancellation (ANC), employing the powerful Least Mean Square (LMS) algorithm on an Artix-7 Basys 3 FPGA platform. The endeavor is driven by the compelling need to address the pervasive challenge of environmental noise during audio transmission, striving for optimal audio signal quality. The application of Adaptive Noise Cancellation involves the generation of anti-noise to counteract ambient noise, ensuring a refined audio signal at the output. The utilization of the LMS algorithm enhances the adaptive filter's efficiency, with a particular emphasis on real- time noise reduction. The algorithm's adaptive nature, governed by equations that dynamically adjust filter coefficients, offers a balance between convergence speed andfinal performance. This is crucial for effective noisecancellation, considering the dynamic and real-time nature ofour application.

To facilitate bidirectional communication withhardware peripherals like microphones and speakers, we integrate an audio codec on the FPGA. This crucial interface ensures the seamless processing and conversion of analog signals to digital and vice versa, bridging the gap between thecontinuous analog world and the digital realm of the FPGA. Despite the potential challenges associated with real-time noise cancellation on an FPGA, including processing delays, audio codec integration, and the handling of floating-point coefficients, we propose a comprehensive solution. Our implementation leverages the Fast-LMS Algorithm, which introduces efficiency-enhancing modifications to the weight updating equation, reducing computational complexity. Additionally, we propose using the I2C communication protocol for synchronizing peripheral hardware with the FPGA and mitigating the impact of floating-pointcoefficients by adopting fixed-point arithmetic. In the execution phase, we detail the steps taken to set up ourproject, including practical testing with a microphone, speaker, and noise source. The expected results outline the seamless flow of noisy input through the system, highlightingthe successful reduction of noise and minimal delay betweenproviding inputand obtaining output sound. In essence, our project not only delves into the technical intricacies of ANC,LMS algorithms, and FPGA interfacing but also seeks to address practical challenges, ensuring a comprehensive and effective solution for real-time noise cancellation in audio applications.

B. Future Trends

The existing configuration can be enhanced by incorporating a more optimized iteration of the Fast LMS algorithm or an

alternative optimized algorithm, resulting in an improved noise cancellation effect. The current model processes ambient noise as an input and generates a diminished versionof the noise signal using a solitary microphone and a single audio output. Presently, the system inherently combines silence with the ambient noise signal to create the input. Thissetup has the potential to be expanded to accommodate any audio signal mixed with ambient noise, delivering a noise- reduced or cancelled audio signal as the output. Moreover, toaugment flexibility and broaden the system's operational scope, an enhancement can be implemented to support mp3 format audio files.

REFERENCES

[1] Goel, P., Chandra, M. (2019) "FPGA Implementation of Adaptive Filtering Algorithms for Noise Cancellation— A Technical Survey". In: Nath, V., Mandal, J. (eds) Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems. Lecture Notes in Electrical Engineering, vol 556. Springer, Singapore.
[2] Lawrence Rabiner, Emmanuel C. Ifeachor, "Digital Signal Processing: A Practical Approach", Addison- Wesley.
[3] Mohamed Salah, Abdel-Halim Zekry, Mohammed Kamel, "FPGA Implementation of LMS Adaptive Filter", 28th NATIONAL RADIO SCIENCE CONFERENCE (NRSC 2011) April 26-28, 2011,National Telecommunication Institute, Egypt.
[4] D. B. Bhoyar, S. Bera, C. G. Dethe and M. M. Mushrif, "FPGA implementation of Adaptive filter for Noise Cancellation", 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2014, pp. 1-5, doi: 10.1109/ECS.2014.6892681.
[5] Dimitris G. Manolakis, Vinay K. Ingle, "Applied DigitalSignal Processing", Cambridge University Press.
[6] Huang, M., Wang, C., Liu, B., Wang, F. and Wang, J., 2020. Quadrature Kalman filter–based state of charge estimation for lithium-ion battery. *Advances in Mechanical Engineering*, *12*(7), p.1687814020942696.
[7] Cheng, Chin, Zuchang Gao, A123's Lithium Iron Phosphate(ANR26650M1-B) Battery Cell Data, IEEE Data Port (2019)
[8] Aung, Htet, Kay Soon Low, and Shu Ting Goh. "State- of-charge estimation of lithium-ion battery using square root spherical unscented Kalman filter (Sqrt-UKFST) innanosatellite." IEEE Transactions on Power Electronics 30.9 (2014): 4774-4783.
[9] Lavanya, N., Fazio, E., Neri, F., Bonavita, A., Leonardi, S.G., Neri G. and Sekar, C., 2016. Electrochemical sensor for simultaneous determination of ascorbic acid, uric acid and folic acid based on Mn-SnO2 nanoparticles modified glassy carbon electrode. *Journal of Electroanalytical Chemistry*, *770*, pp.23-32.
[10] M.A. Hannan, M.S. Lipu, A. Hussan, et al., A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations, Renew. Sustain. Energy Rev. 78 (2017) 834–854.