International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Investigation of FIR Filter Design with Computational Tools for Audio Signal Processing

[1]Gadug Sudhamsu, [2]Dr Chandrasekar Shastry

Submitted: 27/01/2024 **Revised**: 05/03/2024 **Accepted**: 13/03/2024

Abstract: In This paper "Investigation of FIR Filter Design with Computational Tools for Audio Signal Processing" explores the synergies between Finite Impulse Response (FIR) filter design techniques and modern computational tools like MATLAB and Octave. The study emphasizes the fundamental role of FIR filters in shaping the frequency response of audio signals and discusses various design methods. By seamlessly integrating these techniques with computational tools, the paper demonstrates a versatile platform for efficient implementation and real-time analysis in audio signal processing.

Practical demonstrations using representative audio signals illustrate the impact of FIR filter design on crucial audio parameters. Octave is a powerful open-source programming language primarily used for numerical computations and data analysis, similar to MATLAB. It provides a user-friendly environment for scientific and engineering applications. Octave supports various data types, including matrices, vectors, and scalars, making it suitable for handling complex mathematical operations efficiently. Its syntax is largely compatible with MATLAB, allowing users familiar with MATLAB to transition smoothly to Octave. The study highlights the collective potential of FIR filter design and computational tools to advance audio signal processing, providing valuable insights for researchers, practitioners, and educators in the field.

Keywords— *Audio signal, signal processing, FIR filter, Matlab, octave*

I. Introduction

Multimedia represents the amalgamation of diverse media elements, including text, audio, images, videos, and animations, converging to create a seamless digital experience. This integration aims to transcend traditional communication boundaries, fostering interactive and captivating content accessible across an array of platforms, from computers and mobile devices to the vast expanse of the internet. The overarching goal of multimedia is to revolutionize communication by delivering information dynamically and compellingly. By intertwining multiple media forms, multimedia unfolds a richer and more immersive experience for users, diverging from conventional communication reliant on a solitary medium.

Multimedia represents the amalgamation of diverse media elements, including text, audio, images, videos, and animations, converging to create a seamless digital experience. This integration aims to transcend traditional communication boundaries, fostering interactive and captivating content accessible across an array of platforms, from computers and mobile devices to the vast expanse of the internet. The overarching goal of multimedia is to revolutionize communication by delivering information dynamically and compellingly. By intertwining multiple media forms, multimedia unfolds a richer and more immersive experience for users, diverging from conventional communication reliant on a solitary medium.

Text, the fundamental building block, serves as a conduit for information and ideas through written language. Augmenting this, audio elements, encompassing music, speech, or sound effects, add a layer of communication that resonates with the auditory senses, creating a multi-sensory engagement. Visual representation and emotional resonance find expression through images and graphics, conveying intricate concepts and eliciting profound emotions. Meanwhile, videos inject movement, action, and storytelling into multimedia experiences, while animations breathe life into visuals, fostering dynamic and interactive engagement.

The applications of multimedia span diverse fields and industries, each benefiting from its transformative capabilities. In education, multimedia facilitates interactive learning experiences by seamlessly presenting information through a fusion of text, images, and videos. The educational landscape has been revolutionized, with multimedia enabling dynamic lessons and engaging educational content. In the realm of entertainment, multimedia emerges as the cornerstone for creating immersive experiences, leaving its mark in movies, video games, and the burgeoning field of virtual reality. The synergistic blend of various media elements elevates storytelling, transporting audiences to new realms of immersion.

Multimedia's prowess extends into advertising and marketing, where it becomes a strategic tool for captivating

[1]*Department of Computer Science and Engineering,*
Jain Deemed to be University
Email: s.gadug@jainuniversity.ac.in
[2]*Department of Electronics and Communication Engineering,*
Jain Deemed to be University
Email: bs.chandrasekar@jainuniversity.ac.in

audiences and effectively conveying brand messages. The marriage of visuals, audio, and interactive elements in multimedia campaigns ensures a compelling and memorable brand experience. As digital technology advances, the possibilities within the realm of multimedia expand exponentially. High-speed internet proliferation empowers users to seamlessly access and share multimedia content globally. Moreover, the development of robust software tools and multimedia authoring platforms democratizes content creation, making it accessible for both individuals and organizations to craft and disseminate their multimedia narratives.

In essence, multimedia serves as a catalyst for transformative communication, transcending the limitations of traditional mediums. Its versatility and power lie in the harmonious integration of diverse media forms, delivering a tapestry of experiences that captivate, educate, and inspire. The continuous evolution of digital technology ensures that the trajectory of multimedia remains on a dynamic and innovative course, promising ever-expanding horizons for creative expression and communicative impact.

The term "audio signal" encapsulates the transformative electrical representation of sound waves, a fundamental cornerstone of our daily experiences, facilitating perception, communication, and interaction through sound. Originating as variations in air pressure over time, these sound waves undergo a pivotal metamorphosis, being converted into electrical signals by specialized devices like microphones. This conversion process sets the stage for a journey through the intricate realm of audio signal processing, a domain that reverberates across diverse fields, shaping telecommunications, entertainment, music production, broadcasting, and cutting-edge technologies like speech recognition.

In Octave, digital signal processing techniques play a crucial role in telecommunications, particularly in handling audio signals. With Octave, users can implement complex algorithms for encoding and decoding audio data, facilitating efficient transmission over communication networks. Functions like audioread and audiowrite enable the import and export of audio files, while built-in signal processing functions allow for the manipulation and analysis of audio signals. Additionally, Octave provides tools for designing and implementing audio compression algorithms, such as those used in codecs like MP3 or AAC, ensuring optimal balance between data size and audio quality. Through Octave's capabilities, telecommunications engineers can orchestrate the intricate processes involved in audio signal transmission, contributing to the seamless operation of telephony systems, video conferencing platforms, and internet audio streaming services. This integration of Octave into telecommunications workflows exemplifies its significance in supporting the fidelity and reliability of modern auditory interactions across diverse communication channels.

Entertainment and media owe much of their immersive prowess to the strategic utilization of audio signals. In cinematic experiences, television shows, and video games, audio signals augment visual content, delivering realistic sound effects, evocative background music, and compelling dialogue. Behind these auditory marvels, sound engineers and designers manipulate and blend audio signals, crafting a symphony of effects to enrapture audiences and heighten the overall sensory experience.

The digital landscape has ushered in a new era of possibilities for audio signals, allowing for synthesis, digital manipulation, and the creation of electronic music or special effects. The study and understanding of audio signals delve into nuanced concepts such as amplitude, representing loudness, frequency, denoting pitch, phase, and waveform. These fundamental principles underpin the rich tapestry of auditory experiences that characterize our daily lives.

Signal processing, a broader field enveloping audio signals, extends its influence across diverse domains, spanning various signal forms, including audio, images, video, biomedical data, and radar signals. The crux of signal processing lies in its ability to analyze, modify, and interpret signals, extracting valuable information or enhancing their quality. A signal, in essence, becomes a dynamic representation of information that fluctuates over time or space, with audio signals encapsulating the vibrational nuances of sound.

Signal processing can be classified into two domains: analog and digital. Analog signal processing traverses the continuous realm, utilizing electrical or analog circuits for manipulation, employing techniques like filters, amplifiers, and modulation. Conversely, digital signal processing (DSP) operates in the discrete realm, representing signals as numerical sequences and leveraging the computational prowess of computers or digital devices. The ascendancy of digital technology has propelled the widespread adoption of DSP techniques, offering flexibility, accuracy, reproducibility, and the ability to handle copious amounts of data.

II. LITERATURE SURVEY

Preeti Rao et al. [1] The human auditory system is adept at processing complex soundscapes, distinguishing between different sound sources, such as birds chirping, traffic noise, and music, through a process known as auditory scene analysis. This capability allows humans to form high-level abstractions of their environment by analyzing and grouping sensory inputs. Emulating this functionality in machines, particularly for sound source separation and classification, presents significant potential benefits for applications like speech recognition in noisy settings, automatic music transcription, and multimedia data search and retrieval. Developing effective machine listening systems requires models informed by sound production as well as perception and cognition. However, the vast diversity of audio signals poses challenges to generalizing audio processing models beyond those used in speech processing systems. Digital audio signal processing has critical applications in areas such as audio data compression, audio effect synthesis, and audio classification. While audio compression has historically been a prominent application, the increasing importance of managing multimedia content has expanded the use of signal processing for audio segmentation and classification. This is particularly useful in managing digital libraries, professional media production, education, entertainment, and surveillance. Beyond traditional problems like speech and speaker recognition, the growth of digital music archives has sparked interest in improving nonlinear browsing and retrieval methods, emphasizing good indexing for efficient audiovisual navigation. Audio classification also enhances the efficiency of audio compression systems by tailoring the compression method to match the audio type, thereby optimizing coding and transmission processes.

Padmapriya S et al. [2] The design and optimization of Finite Impulse Response (FIR) filters play a pivotal role in digital audio signal processing. These filters are integral for mitigating unwanted noises and enhancing sound quality in various audio-related applications. Recent research efforts have concentrated on enhancing the performance and efficiency of FIR filters through innovative approaches. Introduced an advanced architecture for an adaptive FIR filter, specifically tailored for processing speech signals. This new design focuses on minimizing the delay in adaptation and improving the efficiency metrics of area, delay, and power. The innovation represents a significant leap forward in making speech signal processing systems more efficient and less resource-intensive. Delved into the foundational aspects of acquiring and processing multimedia vocal signals. FIR filters implemented via MATLAB for comprehensive filter processing tasks. This exploration underscores the versatility and effectiveness of using FIR filters, along with IIR filters, in refining the quality of multimedia vocal signals through precise filtering techniques. Together, these studies illuminate the ongoing advancements in FIR filter design, highlighting their critical role in the realm of audio signal processing and the continuous pursuit of efficiency and performance enhancements.

Xiuqin Han et al. [3] This paper focuses on the techniques of collecting and manipulating audio signals using digital signal processing (DSP) techniques. The authors explain various methods of collecting audio signals, including using a microphone, recording from a synthesized source, or obtaining audio from a pre-existing recording. They also discuss the process of adding noise to audio signals, including different types of noise such as Gaussian noise, pink noise, and white noise. The authors then conduct spectral analysis on both noise-free frequency signals and noise-added frequency signals using MATLAB, a programming tool commonly used in DSP. Explained that spectral analysis involves decomposing a signal into its constituent frequencies, known as the frequency spectrum, and analyzing the distribution of energy in the frequency spectrum. The authors also outline the process of filter design using IIR (infinite impulse response) and FIR (finite impulse response) digital filters, which are commonly used in DSP applications. Provided example of how to design these filters using MATLAB and demonstrate their use through simulations. Throughout the paper, the authors illustrate the applications of these techniques through practical simulations, using MATLAB to verify the conclusions drawn from theoretical derivations. The paper provides a comprehensive overview of the methods and techniques used in digital signal processing, with a focus on audio signal processing and filtering.

P. Jubair Ahamed et al. [4] The application of Field Programmable Gate Array (FPGA) software in digital signal processing (DSP), highlighting its utility across various domains such as voice and audio signal processing, image processing, information systems, and system control. Specifically, the focus is on the use of digital filters, providing a detailed methodology for applying filter algorithms through design steps and final simulation validation. The document introduces an innovative design based on a recently published low-level parallel filter Finite Impulse Response (FIR) structure, aimed at reducing device complexity. This new design notably increases the storage capacity of the system, exemplified by the ability to accommodate a larger number of devices in FIR filters with varying levels of parallelism. Filters see a significant scaling in capacity and computational resources when parallelism levels are adjusted, demonstrating both a reduction in complexity and an efficient multiplier aggregation structure conducive to efficient implementation, The article highlights the use of hybrid programming techniques, combining VB (Visual Basic) and Matlab, to design digital filters for FPGA-based systems. This innovative approach enables the efficient implementation of DSP algorithms in FPGA devices, providing a novel solution for signal processing applications." (Octava) "Matlab and VB are combined in the article to design digital filters for FPGA systems, presenting an innovative approach to DSP algorithm implementation in FPGA devices." (Matlab)

Zain Ali et al. [5] The authors explore the use of Particle Swarm Optimization (PSO) and its variants for designing Digital Filter, specifically FIR (Finite Impulse Response) filters. The authors highlight the importance of digital filters in various applications, such as control systems, audio/video processing, and communication systems, and explain why

FIR filters are preferred due to their frequency stability and linearity in phase response. The authors note that designing FIR filters using PSO and its variants involves multi-modal optimization problems, making these techniques suitable for filter design. They explain that PSO and its variants work by updating the velocity and position of particles in the search space, providing an efficient approach for optimizing filter coefficients. The authors demonstrate the effectiveness of PSO and its variants in designing FIR filters by using them to optimize filter coefficients in MATLAB. They compare the results obtained using PSO and ARPSO (Ant Colony Optimization PSO) algorithms with those obtained using the traditional particle swarm optimization (PM) algorithm, and show that the CRPSO (Collaborative Responsive PSO) algorithm outperforms the PM algorithm in terms of frequency spectrum and RMS error. Overall, the paper provides a comprehensive overview of the use of PSO and its variants for designing FIR filters, highlighting their advantages and the optimal design techniques. The authors conclude that PSO and its variants are effective optimization techniques for designing digital filters, particularly FIR filters, and can provide better results than traditional optimization methods in certain cases.

CS Murthy, K Sridevi et al. [6] The design of a high-performance Finite Impulse Response (FIR) filter for Software-Defined Radio (SDR) applications using the Residue Number System (RNS). The RNS offers advantages over traditional FIR implementations due to its parallelism and data partitioning mechanism, but increased bit width can result in performance trade-offs. To overcome this, the authors propose an optimized RNS arithmetic design that includes distributed arithmetic-based residue computation during RNS multiplication and speculative delay-optimized reverse computation to mitigate the trade-off characteristics with filter length. The proposed design utilizes built-in RAM blocks in FPGA devices to store pre-computational values and reduce hardware complexity overhead. The authors evaluate the functionality of the FIR filter core and FPGA hardware synthesis for various input word sizes and FIR lengths. Experimental results show that the proposed optimized RNS system narrows the trade-off between conventional RNS FIR filters and filter length, while achieving considerable complexity reduction. The article highlights the significance of the proposed design in achieving high-performance FIR filter implementation for SDR applications. The use of RNS arithmetic and optimized reverse computation helps to minimize hardware complexity overhead while improving operating speed. The effectiveness of the proposed design in reducing the trade-off between filter length and complexity, making it an attractive option for FIR filter implementation.

Mohammed Abdulzahra Ahmed Al-Dulaimi et al. [7] The design and implementation of a digital Finite Impulse Response (FIR) filter for processing audio signals using the Field-Programmable Gate Array (FPGA) platform, specifically the Altera DE1 board. The design is divided into three primary blocks: the S2P Adapter block, Codec initialization block, and FIR filter block. These blocks are interfaced together and compiled/simulated to obtain accurate results. The whole system is built to be functional, and the frequency response is obtained. The authors achieved 100% accuracy and high-quality implementation of the filter in the FPGA platform by programming it in VHDL. The use of FPGA technology enables rapid prototyping and testing of digital signal processing algorithms, making it an ideal platform for implementing DSP techniques in telecommunications. The authors' design and implementation demonstrates the potential of FPGA-based DSP systems for high-speed and accurate signal processing.

III. System Requirments [Table 1]

Sl. No.	Particulars	Description Matlab	Description Octave
1	Operating System	Windows 11 Windows 10 (version 20H2 or higher) Windows Server 2019 Windows Server 2022	Windows7, 8 10 Mac OS Linux OS
2	Processor	Minimum: Any Intel or AMD x86–64 processor. Recommended: Any Intel or AMD x86–64 processor with four logical cores and AVX2 instruction set support. Note: A future release of MATLAB will require a processor with AVX2 instruction set support.	Intel or AMD Processor
3	RAM	Minimum: 4 GB Recommended: 8 GB	8GB large datasets 16 GB or 32 GB
4	Storage	3.8 GB for just MATLAB 4-6 GB for a typical installation 23 GB for an all products installation. An SSD is strongly recommended.	2.0 GB or 3.0 GB
5	Graphics	No specific graphics card is required, but a hardware accelerated graphics card supporting OpenGL 3.3	It comes with a set of built-in plotting functions that

		with 1GB GPU memory is recommended. GPU acceleration using Parallel Computing Toolbox requires a GPU with a specific range of compute capability.	allow users to create 2D and 3D Plots. Such as FLTK, Gnuplot, and OpenGL

IV. Research Methodology

The Parks-McClellan algorithm, implemented by the fir1 function in MATLAB, is a powerful method for designing Finite Impulse Response (FIR) filters with arbitrary frequency response specifications. This algorithm falls under the category of optimal equiripple algorithms, aiming to minimize the maximum deviation between the desired frequency response and the actual response of the designed filter.

Algorithm Overview:

The Parks-McClellan algorithm employs the Remez exchange algorithm, a mathematical optimization technique that iteratively adjusts the filter coefficients to minimize the error between the desired and actual frequency responses. Unlike other FIR design methods, Parks-McClellan achieves an equiripple response, distributing the error evenly across the specified frequency bands.

Preprocessing:

Utilise a microphone or other audio input device to capture or record the audio signal.

Sampling: By doing a specified number of times (or samples per second), you can turn a continuous analogue audio stream into a discrete digital representation.

The sampled values should be quantized into a limited number of discrete amplitude levels (bit depth).

Processing in the time domain:

Apply filters to the signal to reduce undesired noise or change its frequency makeup (e.g., high-pass, low-pass, bandpass filters).

Adjust the audio's frequency response by equalisation.

Dynamics processing: Adapt the audio signal's dynamic range using strategies including compression, expansion, or limitation.

Processing in the frequency domain

Fourier Transform: Use methods like the Fast Fourier Transform (FFT) to transform the audio signal from the time domain to the frequency domain.

Analyse the signal's frequency content using a spectral analysis technique to pinpoint specific frequency components or features.

Utilise methods like filtering, spectral shaping, or spectral effects to alter the frequency components.

Impacts and Improvements:

Reverberation: By incorporating reflections and decay into the audio input, this effect simulates the acoustic environment of a space.

To produce echoes or spatial effects, add temporal delays between audio streams.

Modulation: Use effects like chorus, flanging, or phasing to give the audio a variety of textures and motion.

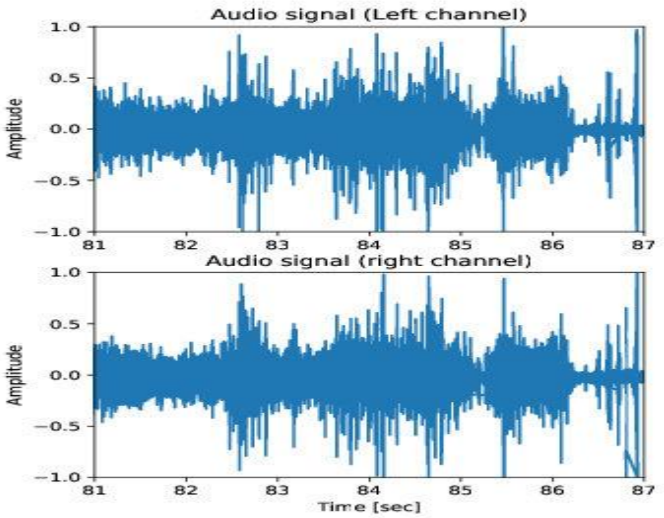


Fig 1 sample audio signal

The discrete Fourier transform (DFT) of a series of complex or real numbers can be quickly computed using the Fast Fourier Transform (FFT) algorithm. It reveals information about the frequency content of a signal by breaking down a signal into its individual frequency components.

Discrete Fourier Transform (DFT):

$$X[k] = \sum[n=0 \text{ to } N-1]\ x[n] * e^{(-j2\pi n*k/N)}$$

where:

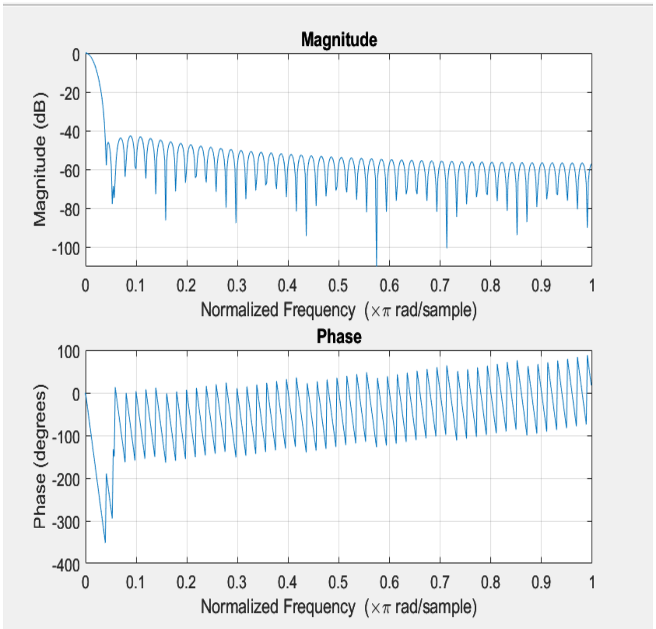
$X[k]$ is the kth frequency component of the transformed signal

$x[n]$ is the nth sample of the input signal

N is the total number of samples in the input signal

The FFT makes the DFT useful for real-time and fast signal processing by lowering its computational complexity from O(N2) to O(N log N). The FFT is an essential tool in many audio and signal processing applications because it allows for tasks like spectral analysis, filtering, and audio signal creation by translating signals from the time domain to the frequency domain.

Fig 2 Magnitude and Phase of audio signal



V. MAT LAB FOR SIGNAL PROCESSING

The provided code snippet demonstrates the process of recording audio using the MATLAB audio recorder function, visualizing the audio signal in both the time and frequency domains, and saving the recorded audio as a WAV file

Five speech Samples have been used for experimentation. This could be a conversation, a speech, or any spoken content. It will provide a diverse range of frequencies and amplitudes.

Instrumental piece as the audio signal. Music contains a wide variety of frequencies and complex waveforms, which can produce interesting patterns in the time and frequency domains. Generate a white noise signal using a random number generator. White noise contains all frequencies at equal intensity and can be used for testing purposes or as a source of background noise. Sine Wave: Generate a pure sine wave of a specific frequency, such as 440 Hz (A4), using a sinusoidal function. This will produce a simple and regular waveform with a single frequency component.

VI. RESULTS AND DISCUSSION

The x-axis label is set to "Frequency (Hz)", the y-axis label is set to "Amplitude", and the title of the subplot is set as "Frequency Domain Plot of the Audio Signal".

The time domain plot in the program provides a visual representation of how the amplitude of the audio signal (x) changes over time. The x-axis represents time in seconds, and the y-axis represents the amplitude of the signal. By plotting the audio signal in the time domain, you can observe the variations and patterns in the waveform. Peaks and valleys in the plot indicate changes in the signal's amplitude, while the overall shape of the waveform represents the characteristics of the audio signal.

The frequency domain plot, on the other hand, illustrates the distribution of frequencies present in the audio signal. It is obtained by performing the Fast Fourier Transform (FFT) on the audio signal, which converts it from the time domain to the frequency domain. The x-axis of the frequency domain plot represents frequency in Hertz (Hz), and the y-axis represents the amplitude of each frequency component.

Same signal – 60 sec has been provided to Matlab & octava for analysis. It is observed that both the outputs are similar but in octave the minimum difference in octava

By analyzing the frequency domain plot, you can identify the dominant frequencies or frequency components present in the audio signal. Peaks in the plot indicate the presence of specific frequencies, while the overall shape and spread of the spectrum provide information about the frequency content of the signal. The magnitude spectrum (AY_0) represents the amplitudes of the frequency components.

Together, the time domain and frequency domain plots provide complementary insights into the audio signal. The time domain plot gives you information about the temporal characteristics of the signal, while the frequency domain plot reveals the spectral composition and frequency distribution. These plots help in understanding and analyzing the properties of the audio signal in both the time and frequency domains.

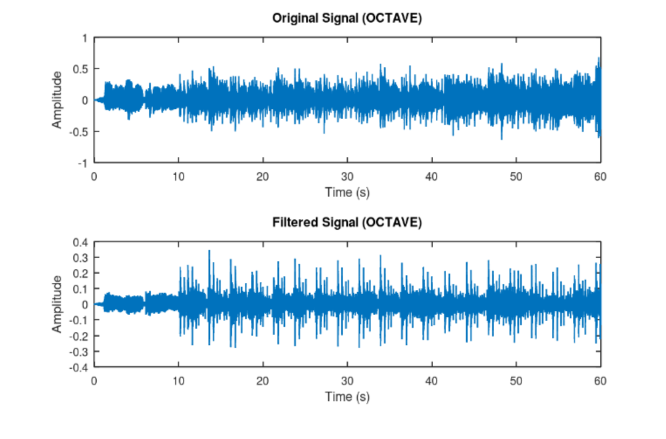
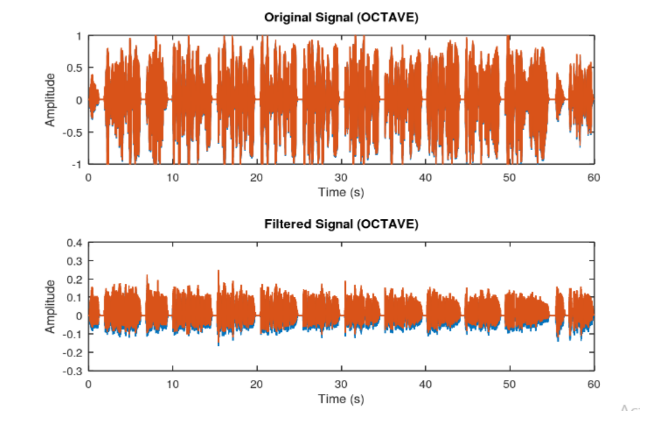
Fig 1	
Fig 2	

Fig 1

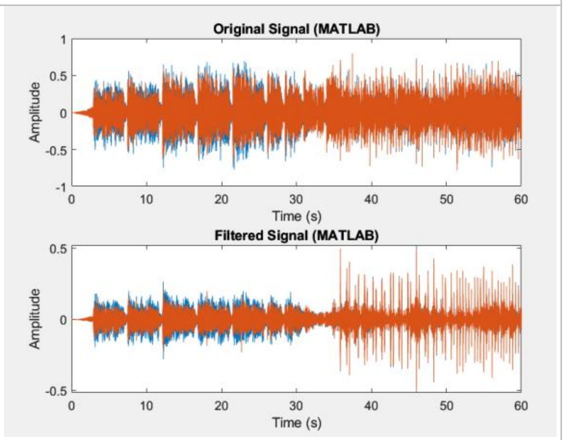


Fig 2

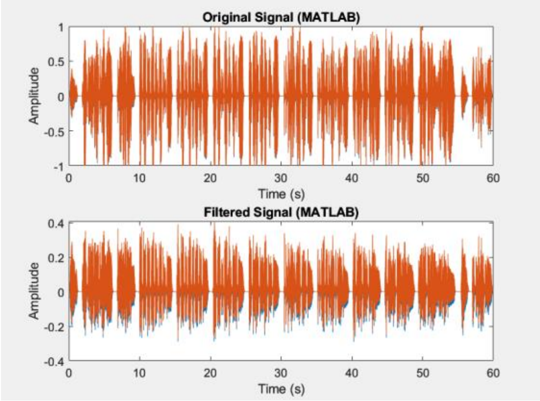


Fig 3

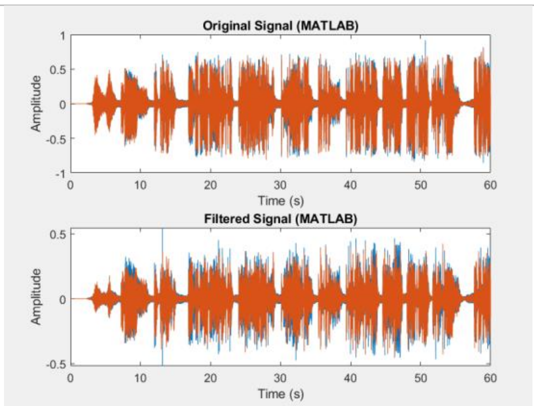
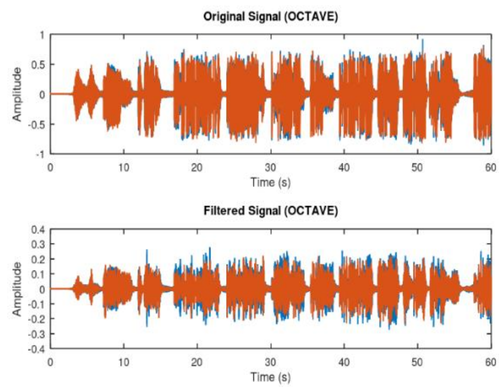


Fig 3



Conclusion

In conclusion, the FIR filter design for audio signal processing is a crucial aspect of signal processing in various fields. The design of FIR filters involves several computational tools and techniques that help in achieving the desired frequency response. In this investigation, Have discussed the basics of FIR filters, their design techniques, and the computational tools used in their implementation. We have started by explaining the basics of FIR filters, including their definition, types, and characteristics. Next, have discussed the design techniques for FIR filters, including the use of transfer functions, frequencies, and impulse responses. We have also explained the different design methods, such as the window method, the least squares method, and the Parks-McClellan algorithm. Furthermore, discussed the computational tools used in FIR filter design, including MATLAB and OCTAVA. Provided examples of how to design and implement FIR filters using these tools. The advantages of using computational tools in FIR filter design are numerous. Firstly, they provide a more accurate and efficient design process. Secondly, they allow for the design of filters with complex specifications and capabilities. Thirdly, they provide a visual representation of the filter design, making it easier to understand and analyze. However, there are also some limitations to using computational tools in FIR filter design. One of the main limitations is the computational complexity of the algorithms used, which can lead to longer design times and larger computation requirements. Despite these limitations, the use of computational tools in FIR filter design is essential for achieving high-quality filter designs. In modern signal processing applications, computational power and speed are critical factors, and the use of computational tools can help to optimize filter design for faster and more efficient processing. In conclusion, FIR filter design is a crucial aspect of audio signal processing, and computational tools play a vital role in the design process. The use of computational tools can help to optimize filter design for faster and more efficient processing, and can provide a more accurate and efficient design process. References:

References

[1] Rao, Preeti. "Audio signal processing." Speech, Audio, Image and Biomedical Signal Processing using Neural Networks. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 169-189.

[2] Padmapriya, S. "Design of an optimized twin mode reconfigurable adaptive FIR filter architecture for speech signal processing." Informacije MIDEM 49.4 (2019): 241-254.

[3] Han, Xiuqin. "Acquisition and its basic processing technology of multimedia vocal signal." International Journal of Pattern Recognition and Artificial Intelligence 34.08 (2020): 2058009.

[4] Ahamed, P. Jubair, and Mohammed Abdul Haseeb. "Implementation of digital IIR filter design based on field programmable gate array." Materials Today: Proceedings 45 (2021): 2573-2577.

[5] Ali, Zain, et al. "Digital FIR filter design by PSO and its variants attractive and repulsive PSO (ARPSO) & craziness based PSO (CRPSO)." Int. J. Recent Technol. Eng.(IJRTE) 9.6 (2021).

[6] Murthy, C. Srinivasa, and K. Sridevi. "FPGA implementation of high speed-low energy RNS based reconfigurable-FIR filter for cognitive radio applications." WSEAS Transactions on Systems and Control 16 (2021): 278-293.

[7] Al-Dulaimi, Mohammed Abdulzahra Ahmed, Husam A. Wahhab, and Ahmed Abdulhussein Amer. "Design and Implementation of Communication Digital FIR Filter for Audio Signals on the FPGA Platform." J. Commun. 18.2 (2023): 89-96.

[8] Spanias, Andreas, Ted Painter, and Venkatraman Atti. Audio signal processing and coding. John Wiley & Sons, 2006.

[9] Lerch, Alexander. An introduction to audio content analysis: Applications in signal processing and music informatics. Wiley-IEEE Press, 2012.

[10] Smith, Julius Orion. Introduction to digital filters: with audio applications. Vol. 2. Julius Smith, 2007.

[11] Härmä, Aki, et al. "Frequency-warped signal processing for audio applications." Journal of the audio engineering society 48.11 (2000): 1011-1031.

[12] Tsoukalas, Dionysis E., John Mourjopoulos, and George Kokkinakis. "Perceptual filters for audio signal enhancement." Journal of the audio Engineering Society 45.1/2 (1997): 22-36.

[13] Li, Qi. "An auditory-based transfrom for audio signal processing." 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. IEEE, 2009.

[14] Zeng, Yuni, et al. "Spectrogram based multi-task audio classification." Multimedia Tools and Applications 78 (2019): 3705-3722.

[15] McClellan, James H., and Thomas W. Parks. "A personal history of the Parks-McClellan algorithm." IEEE signal processing magazine 22.2 (2005): 82-86.

[16] Sahu, Rahul Kumar, and Vandana Vikas Thakare. "Comparative Designing of Optimal FIR Filter using Parks-McClellan & Genetic Algorithm." Evaluation 300 (2015): 400.

[17] Sudhamsu, Gadug, and BS Chandrasekar Shastry. "Audio Signal Processing Using MATLab." 2023 International Conference on Network, Multimedia and Information Technology (NMITCON). IEEE, 2023

[18] Sharma, Garima, Kartikeyan Umapathy, and Sridhar Krishnan. "Trends in audio signal feature extraction methods." Applied Acoustics 158 (2020): 107020.

[19] Maguolo, Gianluca, et al. "Audiogmenter: a MATLAB toolbox for audio data augmentation." Applied Computing and Informatics ahead-of-print (2021).

[20] Zölzer, Udo. Digital audio signal processing. John Wiley & Sons, 2022.