An Innovative Embedded Processor-Based Signal Phase Shifter Algorithm

Aven R. Hamza[1] and Mohammed A. Hussein[2]

[1]Department of Communication Engineering, Sulaimani Polytechnic University,
Sulaimani, Kurdistan region – F.R. Iraq

[2]Department of Electrical Engineering, University of Sulaimani,
Sulaimani, Kurdistan region – F.R. Iraq

Abstract—**Digital filtration is widely used today in many application fields, and with the increased use of low-cost embedded processors, it can be applied to vast areas. A drawback of digital filtration algorithms is the introduction of phase angle shifts in the filtered signals, thereby creating undesirable characteristics in many application fields. In this work, low-pass filters of finite impulse response and infinite impulse response types are designed with an innovative buffering scheme to delay a digitally low-passed signal by an angle ranging from 0° to 180° for real-time signals. The application of the filtration and buffering scheme on a cost-effective embedded processor with limited signal processing capabilities opens the horizons for its applicability in many signal processing fields. In assessing its practicality, the generated filtered output signal is correlated with the original signal (a low-passed version), revealing correlation values reaching 0.99 in certain instances. The novelty of the proposed approach enables its application to a broad-spectrum area of digital signal filtration.**

Index Terms—**Finite impulse response digital filter, Infinite impulse response Digital filter, Phase angle shifting algorithms, Signal correlation, Signals buffering.**

I. Introduction

Nowadays, most signal processing electronic systems depend on digital signal processing (DSP) systems as they are more effective in handling complex signal manipulation operations compared to analog techniques (Pandey and Pratibha, 2022). Digital filters offer advantages of easy design modifications using software as they do not rely on the environmental parameters such as temperature change and magnetic declination in addition to its cost-effectiveness and capabilities of storing digital data (Ozkan and Saday, 2018). Digital filters play a crucial role in various signal processing implementations, including spectrum analysis, digital image manipulation, audio and video processing, sonar and radar systems, as well as pattern recognition (Wang, 2022; Dallalbashi, 2020). They are operated on discrete-time signals and are classified as linear time-invariant (LTI) systems. Digital filters also have several significant properties, such as causality, stability, and feedback (recursivity), in addition to definability by unit impulse responses in the time domain. The unit impulse response sequence can have a finite or infinite duration, a classification that comprises finite impulse response (FIR) and infinite impulse response (IIR) (Dallalbashi, 2020).

ARO-The Scientific Journal of Koya University
Vol. XII, No. 1 (2024), Article ID: ARO.11358. 10 pages
Doi: 10.14500/aro.11358
Received: 20 August 2023; Accepted: 27 January 2024
Regular research paper: Published: 15 February 2024
Corresponding author's email: aven.rawf@spu.edu.iq

Based on the mentioned properties of LTI systems, FIR filter is a non-recursive filter with a linear phase response, meaning it introduces a constant phase shift for all frequencies without distorting their relative timing. It is always stable due to its FIR (Kockanat and Karaboga, 2015; Hannah and Agordzo, 2020). In contrary, IIR filter is a recursive filter that combines both input and output samples for calculating the output signal a characteristic that can lead to instability. One significant distinction of IIR filter is its non-linear phase response, causing distinct frequency components of the input signal to undergo varying amounts of phase shift, leading to phase distortions between the output and input signals (potentially affecting the timing relationships of different frequencies). As a result, the non-linear phase response of the IIR filter may have implications in applications that depend on accurate timing relationships (Kockanat and Karaboga, 2015; Pal, 2017).

Enhancing digital filtration characteristics is under the scope of many research works. Some samples are surveyed here, and the start is with Lai and Lin (2016), who used two iterative reweighted minimax phase-error algorithms to develop linear phase IIR filters. These algorithms involve two steps in each iteration that results in phase-error function exhibiting equi-ripple characteristics over most of the passband, except for small areas near the passband edges. Furthermore, the filter's group delay response achieves nearly equi-ripple characteristics across the entire passband. In Lai (2009), one- and two-dimensional non-linear phase FIR filters with prescribed phase-error are designed. They focused on constrained least-squares and constrained chebyshev designs. This approach resulted in a minimax filter with the lowest energy of complex errors among the various nonunique minimax solutions, providing an effective solution for achieving desired filter performance.

In Xiao, Olivier and Agathoklis (2001), the authors introduced a new approach to implement linear phase IIR filter. The method is based on the frequency weighted least-square error optimization and integrates the Broyden-Flether-Goldfarb-Shanno (BFGS) method. The method minimizes the weighted least square cost function using the BFGS method and the results demonstrate the effectiveness of this approach in achieving excellent filter design outcomes. The work by Vijay, et al. (2022) proposes an IIR filter model referred to as parallel pipeline, which is built upon a FIR filter including the two-level pipeline-based IIR filter and a look-ahead-based models, both of which have been reported to yield substantial enhancements in filter efficiency.

In Agrawal, Kumar, and Bajaj (2020), an innovative technique is presented to design IIR filters with an almost linear phase response that uses fractional derivative constraints (FDCs). The proposed technique improves the passband response and provides a better transition width; however, there is a slight reduction in stopband attenuation in comparison to non-fractional design methods. The designed filter ensures immunity to word length issues and can be extended to accommodate larger numbers of FDCs. Moreover, the method is applicable for designing multiplier less IIR filters. In Tan and Burrus (2019) employed the Gauss-Newton method to create a linear phase IIR filter. Comparisons with an FIR filter revealed that the designed IIR filter achieves lower magnitude error, a smaller order, and reduced group delay. It maintains a consistent phase response with a linearity of $R^2 \geq 0.99$, making it particularly well-suited for applications demanding an approximately linear phase response.

In this research, a circular buffering technique is employed, exhibiting its applicability in various fields highlighted below:

1. Employing it to introduce phase correction between the original and filtered signals to single-tone signals. Besides, the circular buffering algorithm can be helpful in digitally manipulating the phase angle between transmitted and received signals within the range of 0–180°. This digital phase angle adjustment is particularly valuable in specific applications where precise control over phase relationships is crucial such as noise cancellation, audio processing in music production, communication system, medical imaging, radar, and sonar.
2. In real-time processing, acoustic delays introduce latency. The application of a circular buffering algorithm effectively mitigates this issue by compensating for the acoustic delay and it provides the capability of precisely adjusting the phase angles according to specific application's requirements.

II. Digital Filters

Digital filters that can be categorized as FIR or IIR filters hold a crucial part in the area of DSP, as they are used to attenuate unwanted signals and extract desired signals. One of the most prevalent approaches in designing digital filters is to first design an analog filter and then converts it into a corresponding digital filter. A digital filter is defined by its transfer function or difference equation, which mathematically describes its response to various input signals. The implementation usually involves basic arithmetic operations such as addition, multiplication, and division (Kuo and Lee, 2001).

A. Non-Recursive Filter (FIR Filter)

The essential component within a DSP system is the FIR filter. The unit impulse response is finite (h(n)=0 for $n < 0$ and $n \geq M$) and therefore, the unit sample response is present in the range from 0 to M-1. The transfer function of FIR filter H(z) is determined by Zhao (2022):

$$H\left(z\right) \aleph \aleph \sum_{n=0}^{N-1} \aleph h_n z^{-n} \quad h_0 \quad h_1 z^{-1} \quad h_2 z^{-2} \qquad\qquad h_{N-1} z^{-(N-1)} \tag{1}$$

The coefficient h_n, referred to as zeros is used to design different types of filters where the end of the summation term N-1 is the filter's order. FIRs filters have the advantages of stability and linear phase response, and are non-recursive filters widely used in signal processing. In addition, they employ Fast Fourier Transform (FFT) to optimize their implementation, enabling efficient convolution computation. However, meeting design specifications often necessitates a higher number of coefficients (Nor, et al., n.d.).

B. Recursive Filter (IIR Filter)

IIR filter is a zero-pole filter, and it has an IIR (i.e., h(n)=0 for $n < 0$), hence, the unit sample response persists indefinitely from 0 to ∞. This type of filter is generally realized using feedback structure, so it is called a recursive filter. The transfer function of IIR filter is expressed as follows (Kuo and Lee, 2001)**:**

$$H\left(z\right) = \frac{\sum_{l=0}^{L-1} b_l z^{-l}}{1 + \sum_{m=1}^{M} a_m z^{-m}} \tag{2}$$

In this equation, b_1 and a_m are filter's weight coefficients. Various types of filters, including Butterworth, Chebyshev (Type I and II), and Elliptic filters, are available in which MATLAB software offers functions tailored for each specific filter type (Li, 2022). One notable advantage of IIR filters is their ability to fulfill design specifications using a reduced number of filter coefficients compared to FIR filters. Nevertheless, IIR filters can exhibit instability, potentially leading to unpredictable outputs due to their feedback-based nature. It also introduces non-linear phase shifts, causing different frequency components to experience varying delays that affect the sequential relationships within the signal (Nor, et al., n.d.).

III. Correlation Analysis

Correlation analyses play a key role in various applications, particularly in DSP. It is widely used to assess the similarity between two signals, making it invaluable for tasks such as detecting signals corrupted by noise, measuring time delays between signals, and defining the impulse response of a system. Correlation functions are extensively employed in the analysis of random processes, where the statistical attributes of a stochastic signal such as mean, variance, and correlation functions, commonly vary over time. For a random variable signal x(n), the mean can be formulated by Kuo and Lee (2001):

$$m_x = E\left[x\left(n\right)\right] = \frac{1}{N}\sum_{n=0}^{N-1} x(n) \tag{3}$$

Where m_x is the sample mean of x(n), E [] is the expectation operator, and N is the number of samples in the short-time analysis interval. The variance of random signal x(n) can be determined by Kuo and Lee (2001):

$$\sigma_x{}^2 = E\left[x\left(n\right)\right] = \frac{1}{N}\sum_{n=0}^{N-1}[x\left(n\right) - m_x]^2 \tag{4}$$

Correlation can be characterized into two main types: autocorrelation and cross-correlation. Auto-correlation is employed to measure the correspondence between different segments of the same signal, such as x(n). Autocorrelation can be calculated for the time instances n and k by Kuo and Lee (2001):

$$r_{xx}\left(n,k\right) = E\left[x\left(n\right)x\left(k\right)\right]$$

$$= E\left[x\left(n\right)\right]E(x\left(k\right)] = \begin{cases} 0,\ n \neq k \\ \sigma_x{}^2,\ n = k \end{cases} \tag{5}$$

Cross-correlation is utilized to measure the similarity between two different signals, such as x(n) and y(n) that can be calculated by Kuo and Lee (2001):

$$r_{xy}\left(n,k\right) = E\left[x\left(n\right)y\left(k\right)\right] = E\left[x\left(n\right)\right]E(y\left(k\right)] \tag{6}$$

The correlation coefficient, often referred to as an index of correlation, allows analyzing the connection between two signals, and it ranges from −1 to +1, providing a measure on the strength and direction of the correlation between the signals. The two random variables have a strong positive linear correlation if the correlation coefficient approaches +1 and they have a strong negative linear correlation if the correlation coefficients are nearly −1. In addition, a correlation coefficient of zero indicates that the two signals are not linearly correlated (Kohn, 2006). The calculation of cross-correlation coefficients (ρ_{xy}) between the input signal x(n) and the filtered signal y(n) is accomplished through the following equation (Kohn, 2006):

$$\rho_{xy} = \frac{\frac{1}{N}\sum_{n=0}^{N-1}\left[x\left(n\right) - m_x\right].[y\left(n\right) - m_y]}{\sqrt{\text{üüü}\frac{1}{N}\sum_{n=0}^{N-1} x\left(n\right) - m_x{}^{\;2}\quad y\left(n\right) - m_y{}^{\;2}}} \tag{7}$$

IV. Proposed Circular Buffer Algorithm

A circular buffer, often referred to as a ring buffer, is a simple buffer data structure that follows the First-In-First-Out ordering principle. In a circular buffer, the element that is initially added to the buffer is the first one to be subsequently out from it. Fig. 1 shows the basic mechanism used for buffering P data elements. In general, it uses two pointers: one for writing the sample in the buffer, and the other for reading it. The process begins with the initialization of the buffer, typically represented as an array of fixed-size with cyclic behavior. Initially, all buffer locations are empty and as new data arrives, it is placed at the beginning of the buffer (buf [0]), overwriting the previous data sample stored there. New data are stored in the next buffer location (buf [1]), overwriting what is available there and effectively moving clockwise. This process of replacing the oldest data with the newest data is key to the functioning of circular buffers. By continuously updating the buffer in this manner, only one location in the buffer needs to be modified for each new data sample (Smith, 2013; Tan and Jiang, 2018).

Real-time signal processing, commonly observed in the field of digital signal processors, involves the simultaneous production of the output signal while acquiring the input signal. In this field, real-time applications efficiently utilize algorithms to process groups of input samples, generating corresponding groups of output signals. To implement a real-time FIR or IIR digital filter, as depicted in Fig. 2a and b, it is necessary to have access to a specific number of the most recent samples from both the input and output. For instance, in considering the second-order IIR filter, the values of the coefficients b_0, b_1, b_2, a_0, a_1, a_2 from the input signals x(n), x(n-1), and x(n-2), in addition to the output signals y(n), y(n-1), and y(n-2), need to be known. To achieve this, these coefficients must be stored in memory and continuously updated as new samples are acquired. Circular buffering is typically employed to facilitate this process.

The proposed algorithm for aligning the phase shift of FIR and IIR digital filters involves applying circular buffering to both filters, with a chosen buffering size of 1024 for this research. When implementing circular buffering, it is essential to consider several parameters for effective processing, including:

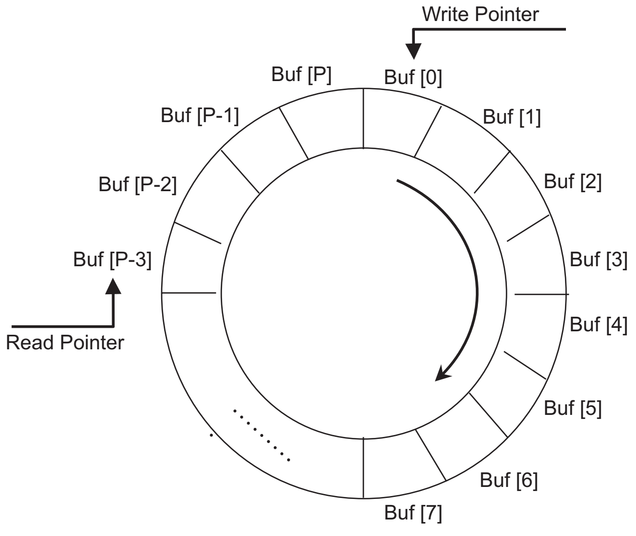

Fig. 1. Circular buffering.

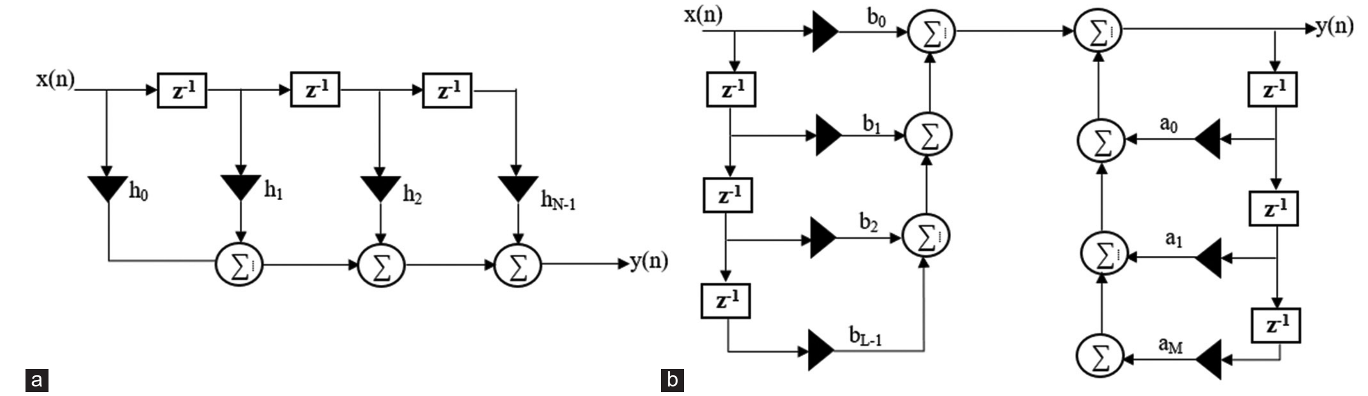

Fig. 2. Digital filter structures for (a) finite impulse response filter and (b) infinite impulse response filter.

1. Buffer Length: The circular buffer holds a total number of samples.
2. Read Pointer: This index indicates the current read position within the circular buffer x(n).
3. Write Pointer: The index is for the current write position within the circular buffer.
4. Step Size: The step size of memory in the algorithm is set to one, indicating that each sample is sequentially stored in memory.
5. Update Frequency: It is the rate at which the circular buffer is updated with new samples.

By carefully managing these parameters, the circular buffering technique enables effective phase correction of signals filtered digitally, allowing accurate signal processing and alignment.

The algorithm used for circular buffering is shown in Fig. 3. The start is with initializing the buffer and inputting signals after it gets conditioned for proper processing, from scaling to DC-offsetting till it reaches the phase of applying the filtration algorithm. The resulting filtered signal is stored in the buffer at the current index. Subsequently, the program proceeds to determine the required phase shift by analyzing signal B and assigning the calculated value to the variable J. To ensure the correctness of the required phase shift, the code checks whether J falls within the desired range. If it does not meet the criteria, corrective actions are taken to make them valid and to obtain a correct shift value. Once the condition is satisfied, signal B advances to proper buffer output position through B(J). The program continuously repeats this process until it reaches the end of the buffer and start again entering new inputs to the buffer at the start location. The filtration algorithm continues till the user ends it.

Fig. 4 shows the graphical abstract that briefly describes the proposed algorithm's methodology in addressing phase delay issues in digital filters. It provides a visual overview of the used approach and allows quick understanding of the algorithm's significance in DSP.

V. Results and Discussion

A digital FIR and IIR low pass were designed using MATLAB and Table I shows the filters parameters.

TABLE I
Digital Filters Parameters

Parameters	FIR low-pass filter	IIR low-pass filter	FIR low-pass filter	IIR low-pass filter
Sampling frequency (Hz)	44100	44100	44100	44100
Cut-off frequency (Hz)	500	500	500	500
Filter order	2	2	4	4
Frequency of the input signal	200	200	200	200

FIR: Finite impulse response, IIR: Infinite impulse response

Different filter orders were used to show performances variations effect in phase correction. The MATLAB function fir1(n,2πfc, ftype) is used to design the FIR low-pass filter. Here 'n' is the filter order, 'fc' is the normalized cut-off frequency, and 'ftype' is the type of the filter (low-pass, high-pass, band-pass, and band-stop filters). Furthermore, for designing IIR low-pass filter, the MATLAB function butter (n, 2πfc, 'low') is used. In this function, 'butter' referred to Butterworth IIR filter type, 'n' represents the order of the digital filter, 'fc' denotes the normalized cut-off frequency, and 'low' signifies that it is a low-pass filter (The MathWorks, Inc., 2023). The obtained results are presented in the next two subsections in which the first is on ones obtained from the MATLAB program, and the next is on outcomes obtained from the embedded processor.

A. MATLAB

Fig. 5a and b shows the comparison between the original signal and the filtered signal using 2nd and 4th orders FIR low-pass filters using MATLAB. The original signal is depicted by the blue line, while the filtered signal is represented by the red line. The input signal has a frequency of 200 Hz, and the filter is designed to have a cut-off frequency of 500 Hz. It is significant to mention that FIR filters have linear phase characteristics, meaning that all frequency components of the input signal experience a consistent time delay. This property ensures that the relative timing relationships within the signal are preserved without introducing significant phase distortions. The small group delay exhibited by the FIR filters contributes to the minimal phase shift between the original and filtered signals.

Fig. 6a and b depict the original and filtered signals for both a 2nd and 4th orders IIR low-pass filter using MATLAB.

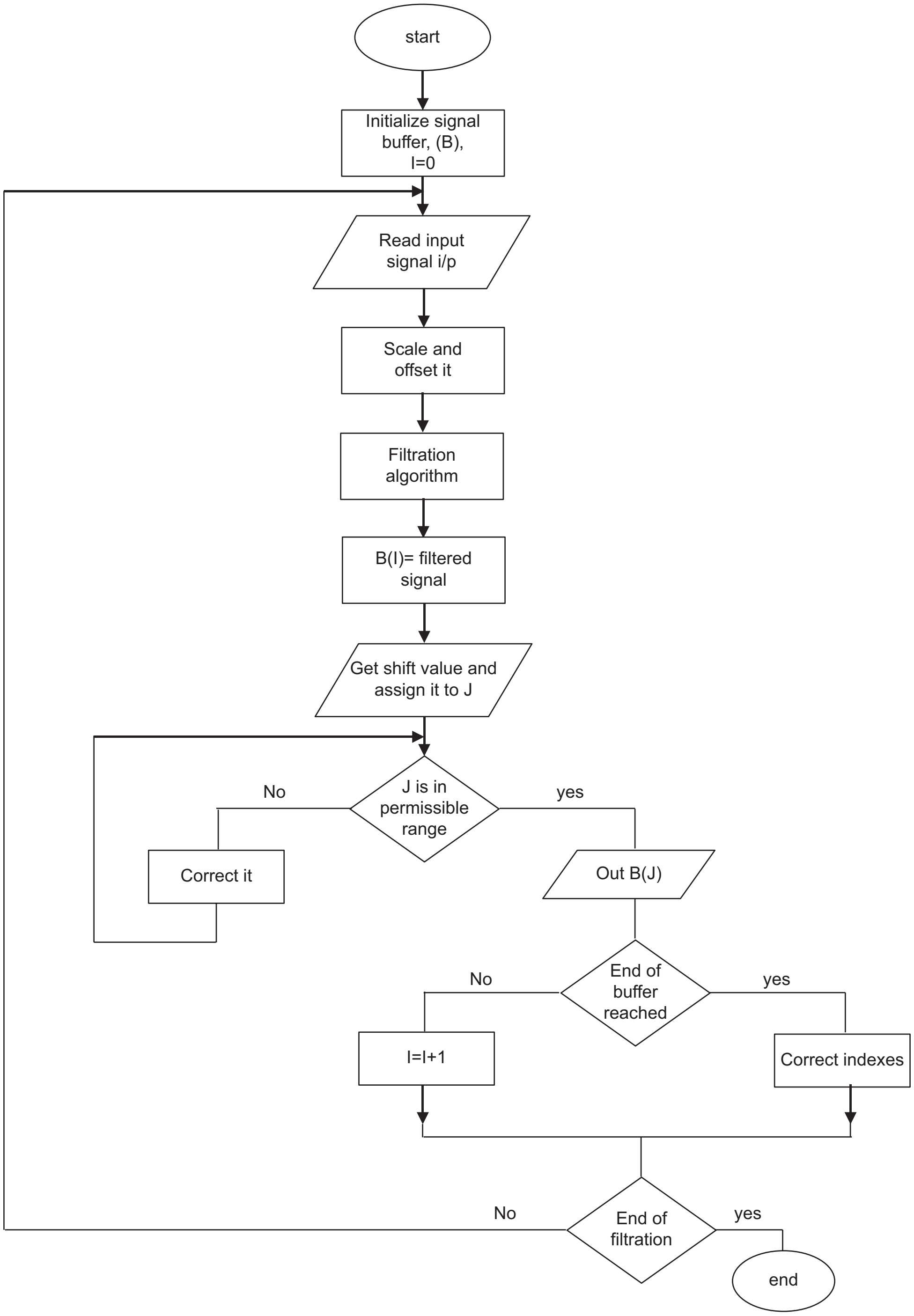

Fig. 3. Flowchart of circular buffering algorithm.

The aim of this analysis is to explore the consequences of employing an IIR low-pass filter on an input signal with a frequency of 200 Hz. Using a sampling frequency of 44100 Hz (to ensure an accurate representation of the signal's characteristics) a Butterworth filter was designed with a cut-off frequency of 200 Hz. From the figure, it is observed that the original signal and the filtered signal exhibited noticeable differences due to the non-linear phase and large group delay properties of the IIR filter. A property causing phase shift between the original and filtered signals and by increasing the filter order the phase shift increases in value.

B. Digital Processor

The connection setup for phase correction of digital filters using an embedded processor is illustrated in Fig. 7. The process can be subdivided into four distinct phases. During the initial phase, the input signals are generated according to the specifications provided in Table II (refer to section 5 results and discussion). In the second stage, a microphone is used to capture the signal. Before passing the signal to the Arduino, it is biased (dc-offset correction) to 1.4 V, as the Arduino is unable to read negative voltage values. Subsequently, in the third stage, the filter coefficients

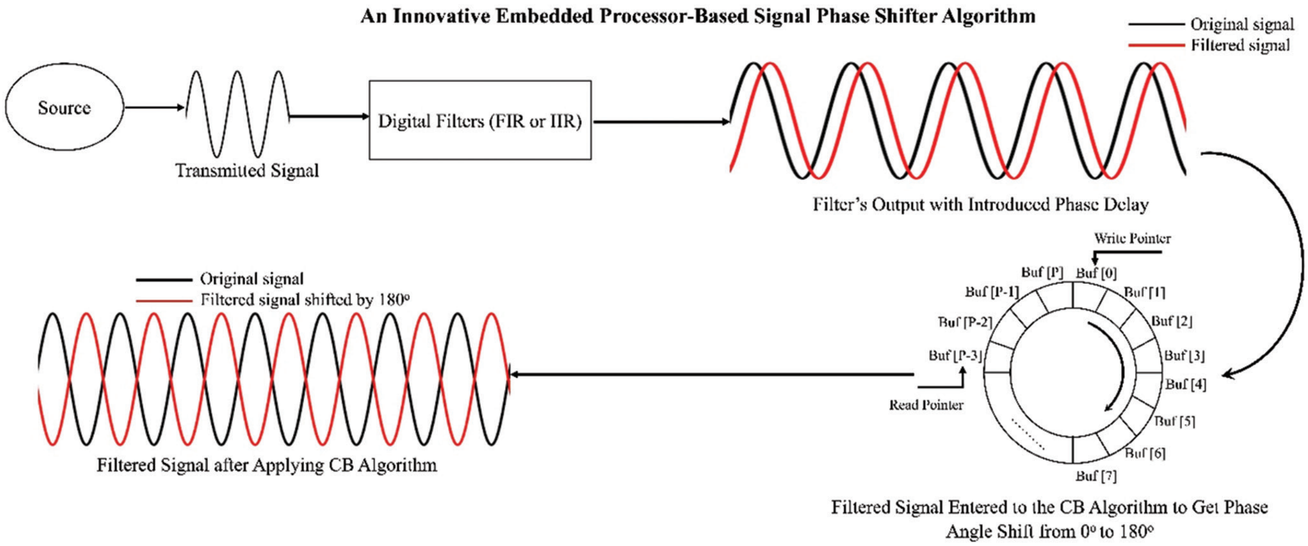
An Innovative Embedded Processor-Based Signal Phase Shifter Algorithm
Original signal
Filtered signal
Source
Transmitted Signal
Digital Filters (FIR or IIR)
Filter's Output with Introduced Phase Delay
Write Pointer
Buf [P]
Buf [0]
Buf [P-1]
Buf [1]
Buf [P-2]
Buf [2]
Buf [P-3]
Buf [3]
Buf [4]
Buf [5]
Buf [6]
Buf [7]
Read Pointer
Original signal
Filtered signal shifted by 180°
Filtered Signal after Applying CB Algorithm
Filtered Signal Entered to the CB Algorithm to Get Phase Angle Shift from 0° to 180°

Fig. 4. Graphical abstract showing the algorithm's success in addressing phase delays in digital filters.

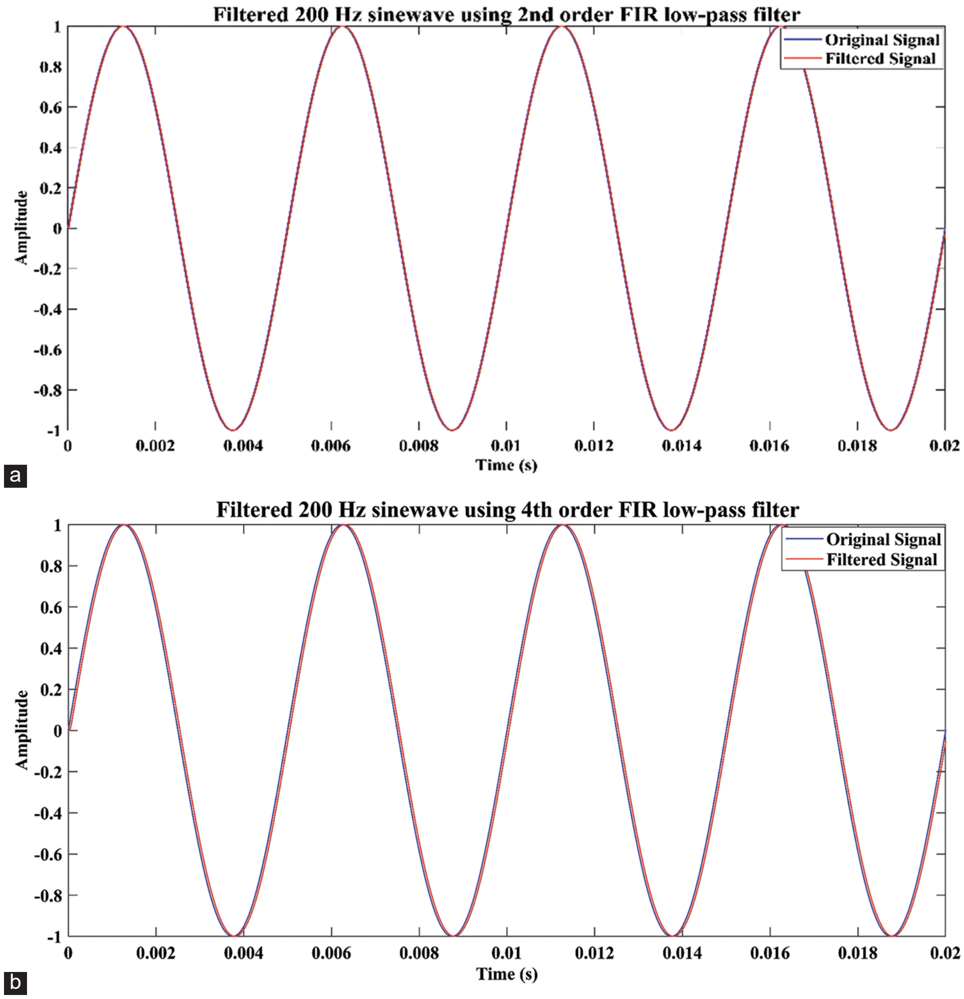

Fig. 5. Original and filtered signals for (a) 2nd order finite impulse response (FIR) low-pass filter and (b) 4th order FIR low-pass filter.

from MATLAB software are transferred to the Arduino IDE and the signals are filtered using both FIR and IIR filters implemented with the proposed algorithm. The algorithm ensures the availability of the necessary number of the

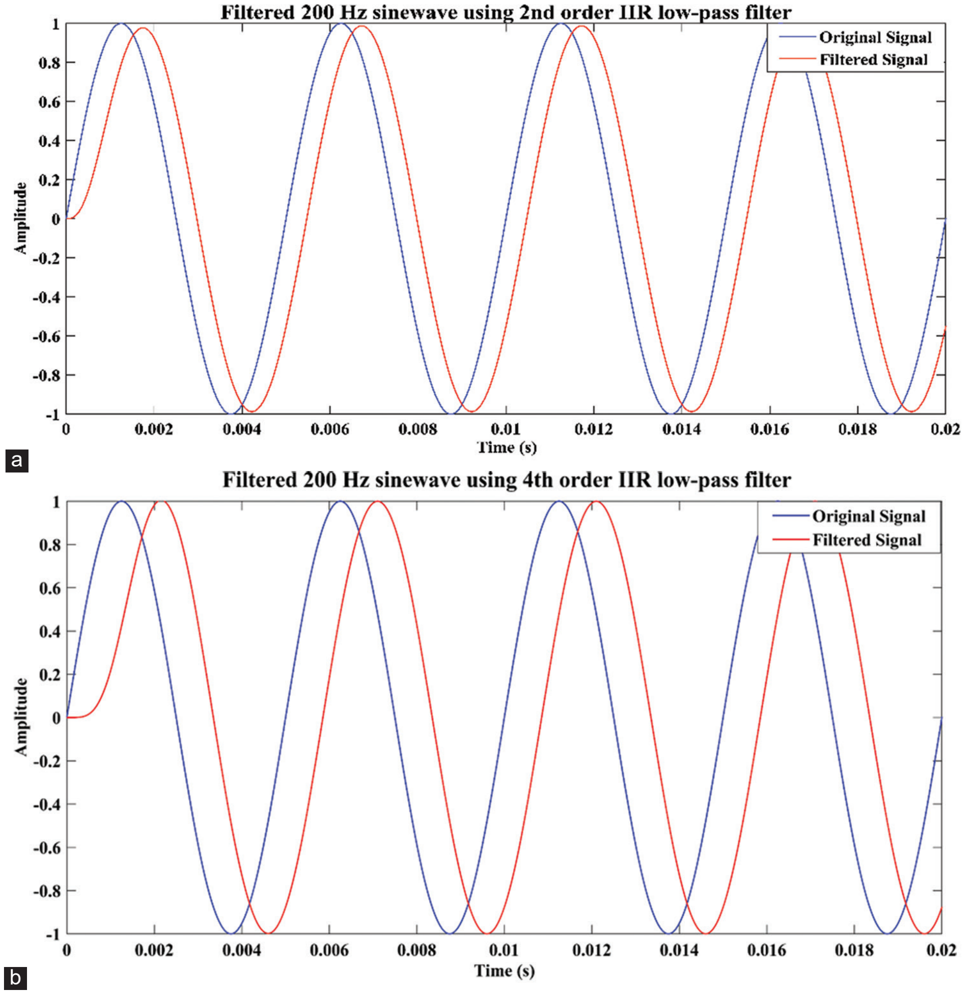

Fig. 6. Original and filtered signals for (a) 2nd order infinite impulse response (IIR) low-pass filter and (b) 4th order IIR low-pass filter.

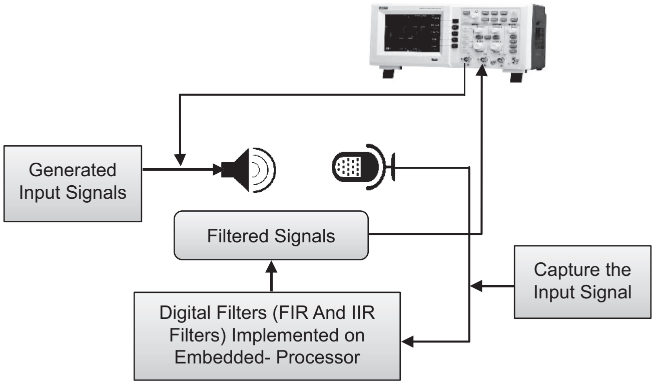
Generated Input Signals
Filtered Signals
Digital Filters (FIR And IIR Filters) Implemented on Embedded- Processor
Capture the Input Signal

Fig. 7. Connection setup for filtration of various signals.

most recent samples for computation and by adjusting a potentiometer, the phase shift between the two signals is controlled and the refinements could be seen on channels A and B of the oscilloscope. The correlation coefficients are then obtained and monitored through the Arduino serial monitor.

Fig. 8 shows the output of the second order FIR low-pass filter. The original signal is presented by the blue color and the filtered signal is represented by the yellow color. The second-order FIR filters exhibit minimal group delay, typically just one sample, resulting in negligible impact on the filter's output. While the difference between the original and filtered signals may not be readily apparent Fig. 8a, the introduction of a circular buffer allows for the adjustment of the phase shift between the input signal and the filtered signal, ranging from 0° to 180°, as illustrated in Fig. 8b and c.

Fig. 9 illustrates the results obtained from applying a fourth order FIR low-pass filter. The filter's output is minimally affected, since FIR filter has linear phase response as shown in Fig. 9a, where the difference between the original and filtered signals is not visibly distinct. However, by implementing a circular buffer, it is possible to adjust the phase shift between the input signal and the filtered signal. In Fig. 9b and c, the circular buffer is utilized to achieve phase shifts from 0° to 180°.

Fig. 10 shows the second order IIR low-pass filter for various phase angles. Due to non-linearities of IIR filters, there is a phase shift between the original and the filtered signals of approximately 40.3^{o}, as shown in Fig. 10a. To counteract by employing the proposed algorithm, the phase shift can be corrected, including higher ranges from 0° to

180°. Fig. 10b and c reveal the filtered signal after applying the proposed algorithm and adjusting the phase shift from 0° to 180°, as required.

Fig. 11 displays the fourth order IIR low-pass filter at different phase angles. Due to inherited non-linearity of IIR filters, a phase shift is introduced between the original and filtered signals. In this case, the phase shift is approximately 103.7°, as shown in Fig. 11a. To tackle this issue, the proposed algorithm is employed, providing the capability to dynamically adjust the phase shift within a range of 0–180°. In Fig. 11b and c, the filtered signals are presented after applying the proposed algorithm with phase shifts adjustment from 0° to 180°, as required. These figures demonstrate the effectiveness of the algorithm in manipulating the phase shift, resulting in altered phase relationships between the original and filtered signals.

A tabular approach is employed to compare the cross-correlation coefficients between FIR and IIR low-pass filters with various filtering orders and circular buffering algorithms. The cross-correlation coefficients are calculated in MATLAB and by the embedded processor. The results were nearly the same for all filter orders except the 4th order IIR low-pass filter. The proposed algorithm yields approximately a 0.99 correlation coefficient for both digital filters with different filter's orders using a 200 Hz sine wave as an input frequency, as shown in Table II.

Fig. 12 shows that the proposed algorithm results in achieving good performances values. Its versatility lies in the

TABLE II
Comparison of Cross-Correlation Coefficients

Filter order	Correlation coefficients from MATLAB	Correlation coefficients from Arduino	Correlation coefficients using proposed algorithm to align input and output signals	Correlation coefficients using proposed algorithm to make 180° phase shift between input and output signals
FIR 2nd order	0.99959	1	1	−0.99
FIR 4th order	0.99838	0.99	0.99	−0.98
IIR 2nd order	0.82969	0.74	0.99	−0.98
IIR 4th order	0.47993	−0.37	0.99	−0.99

FIR: Finite impulse response, IIR: Infinite impulse response

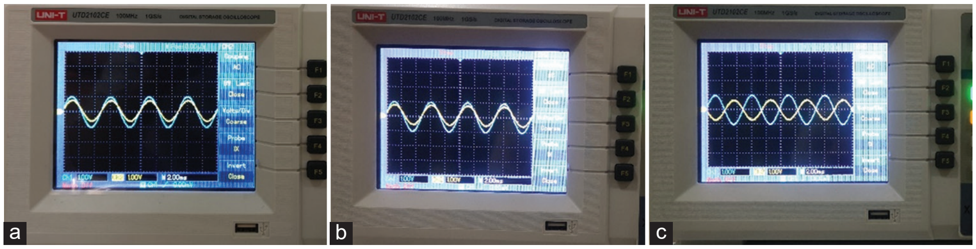

Fig. 8. Second order finite impulse response low-pass filter (a) output of the filter, (b) phase angle between the input and output of the filter is 0°, and (c) phase angle between the input and output of the filter is 180°.

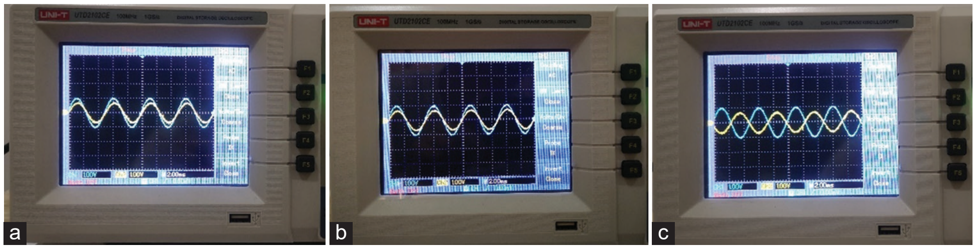

Fig. 9. Fourth order finite impulse response low-pass filter (a) output of the filter, (b) phase angle between the input and output of the filter is 0°, and (c) phase angle between the input and output of the filter is 180°.

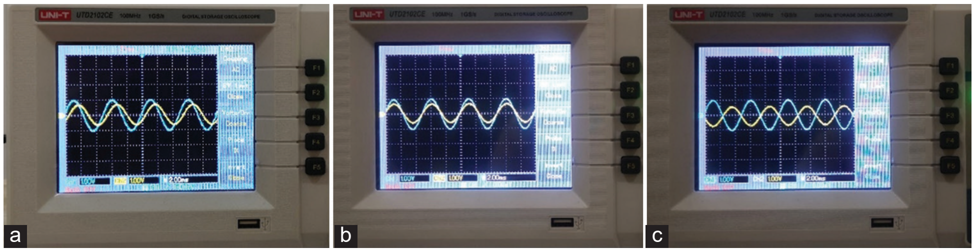

Fig. 10. Second order infinite impulse response low-pass filter (a) output of the filter, (b) phase angle between the input and output of the filter is 0°, and (c) phase angle between the input and output of the filter is 180°.

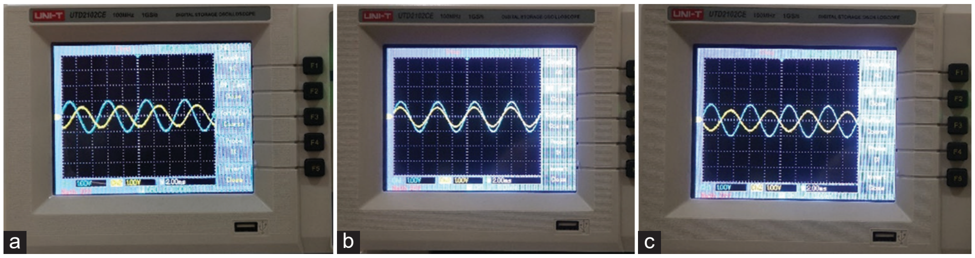

Fig. 11. Fourth order infinite impulse response low-pass filter (a) output of the filter, (b) phase angle between the input and output of the filter is 0°, and (c) phase angle between the input and output of the filter is 180°.

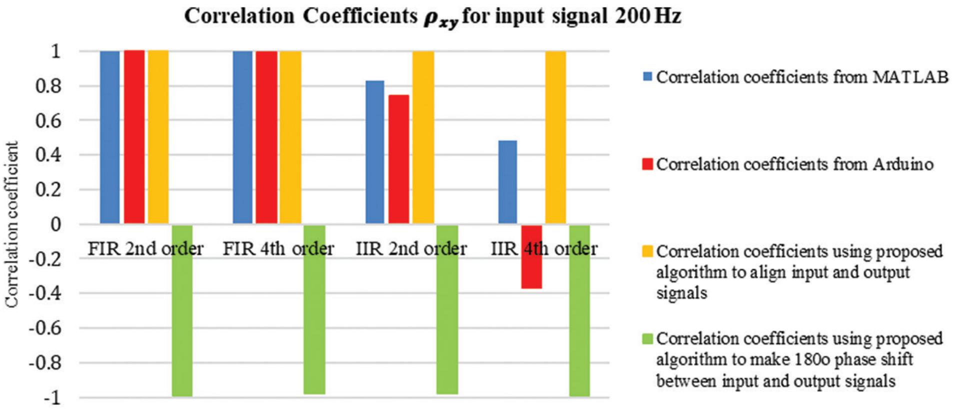

Fig. 12. Cross-correlation coefficients for finite impulse response low-pass, infinite impulse response low-pass filters with and without using circular buffering algorithm.

TABLE III
Comparison between Proposed Algorithm and Previous Works

References	Name of the Algorithm	Achievements
Lai and Lin, 2016	Two iterative reweighted minimax phase-error algorithms were used to develop linear phase IIR filters.	Filter’s group delay response achieves nearly equi-ripple characteristics across the entire passband.
Tan and Burrus, 2019	Gauss-Newton method was employed to create a linear phase IIR filter.	Comparisons with an FIR filter revealed that the designed IIR filter achieves lower magnitude error, a smaller order, and reduced group delay. It maintains a consistent phase response with a linearity of R^2 ≥0.99
Agrawal, Kumar, and Bajaj, 2020	An innovative technique is presented to design IIR filters with an almost linear phase response that uses fractional derivative constraints.	The proposed technique improves the passband response and provides a better transition width, however, there is a slight reduction in stopband attenuation in comparison to non-fractional design methods
Current work	New algorithm named circular buffering algorithm was used to digitally change the phase angle of both FIR and IIR digital filters.	The results show that by applying this novel buffering method, both the phase shift and the correlation between the original and filtered signals are effectively corrected. Furthermore, the proposed technique allows users to adjust values of phase shifts and obtained correlations in a flexible manner, ranging from 0° to 180° and−1 to+1, respectively.

FIR: Finite impulse response, IIR: Infinite impulse response

ability to adjust correlation coefficients according to specific application’s requirement, making it highly adaptable. Consequently, the algorithm enables the alignment of a signal with another desired signal or creates any required phase shift between them. These customizable adjustments offer enhanced flexibility and optimize the algorithm’s performance for various practical scenarios.

The primary objective of the current work is to enhance the efficiency of delayed signals processing in various applications. Table III presents a comparative analysis between the algorithm proposed in this work and prior research ones by precisely examining and contrasting it against them.

VI. Conclusion

This paper introduces an innovative circular buffering technique applicable to both FIR and IIR filters. In digital filters, the non-linear phase response and the large group delay create a phase shift between the input and output signals, affecting the correlation between them. To address this issue, the proposed approach involves designing digital filters using MATLAB and implementing them on an embedded processor to be applicable for real-worlds applications. To assess the effectiveness of the proposed algorithm, correlation coefficients are computed through simulations and practical implementation. The outcomes show that by applying this novel buffering method, both the phase shift and the correlation between the original and filtered signals are effectively corrected. Furthermore, the proposed technique allows users to adjust values of phase shifts and obtained correlations in a flexible manner, ranging from 0° to 180° and -1 to +1, respectively.

In conclusion, this innovative buffering technique offers a promising solution to reduce phase shifts and enhance correlations in the DSP. It also opens possibilities for improved signal manipulation and analysis in a variety of real-world applications.

VII. Limitation and Future Work

Arduino based digital filtration and circular buffering is used as a platform in this work. It is important to acknowledge the limitation inherited by this choice, as the filter order and circular buffer length increases, the computational complexity will be a restricting factor. To address this limitation and potentially enhance the computational efficiency, future implementation of this research could explore the utilization of Field-Programmable Gate Array devices. Their parallel processing capabilities may offer a more scalable and efficient solution to handle the increased complexity associated with higher filter's orders and longer circular buffers. Thus, ensuring that the circular buffering algorithm will be capable of addressing phase delays across wide ranges of frequencies and make it frequency-independent, to effectively correct delays in complex signals with various frequency components.

References

Agrawal, N., Kumar, A., and Bajaj, V., 2020. A new design approach for nearly linear phase stable IIR filter using fractional derivative. *IEEE/CAA Journal of Automatica Sinica*, 7(2), pp.527-538.

Dallalbashi, Z.E., 2020. MatLab based design and implemetation of digita filter. *IJCSNS International Journal of Computer Science and Network Security*, 20(1), p.95.

Hannah, A.A.S., and Agordzo, G.K., 2020. A design of a low-pass FIR filter using hamming window functions in Matlab. *Computer Engineering and Intelligent Systems*, 11(2), pp.24-30.

Kockanat, S., and Karaboga, N., 2015. The design approaches of two-dimensional digital filters based on metaheuristic optimization algorithms: A review of the literature. *Artificial Intelligence Review*, 44, pp.265-287.

Kohn, A.F., 2006. Autocorrelation and Cross-Correlation Methods. In: *Wiley Encyclopedia of Biomedical Engineering*. Wiley, United States.

Kuo, S.M., and Lee, B.H., 2001. *Real-Time Digital Signal Processing: Implementaions, Applications and Experiements with the TMS320C55X*. Wiley, Hoboken, NJ, USA.

Lai, X., 2009. Optimal design of nonlinear-phase FIR filters with prescribed phase error. *IEEE Transactions on Signal Processing*, 57(9), pp.3399-3410.

Lai, X., and Lin, Z., 2016. Iterative reweighted minimax phase error designs of IIR digital filters with nearly linear phases. *IEEE Transactions on Signal Processing*, 64(9), pp.2416-2428.

Li, J., 2022. A Comparative Study of Different Filters for Speech Signals. In: *International Conference on Intelligent Systems, Communications, and Computer Networks, China*, pp.394-403.

Nor, N.M., Hashim, N.M.Z., Rosli, S.N., and Aziz, K.A.A., n.d. A *Study on Application Digital Filtering for Audio Signal Processing in Digital Signal Processing*. IEEE, United States.

Ozkan, I.A., and Saday, A., 2018. *Design and Implementation of FIR Filter based on FPGA*. Marketlinks Logo, Bulgaria.

Pal, R., 2017. *Comparison of the Design of FIR and IIR Filters for a Given Specification and Removal of Phase Distortion from IIR Filters*. IEEE, United States, pp.1-3.

Pandey, A., and Pratibha, E., 2022. A review of techniques for optimization and implementation of digital filters on FPGA. *International Journal for Research in Applied Science and Engineering Technology* (*IJRASET*), 10(6), pp.133-139.

Smith, S.K., 2013. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Elsevier, Netherlands.

Tan, J., and Burrus, C.S., 2019. *Near-Linear-Phase IIR Filters Using Gauss-Newton Optimization*. IEEE, United States, pp.876-879.

Tan, L., and Jiang, J., 2018. *Digital Signal Processing: Fundamentals and Applications*. Academic Press, United States.

The MathWorks, Inc., 2023. *MathWorks*. Available from: https://www.mathworks.com/help/signal/ref/fir1.html [Last access on 2023 Aug 15].

Vijay, V., Rao, V., Chaitanya, K.R., Venkateshwarlu, S.C., Pittala, S., and Vallabhuni, R.R., 2022. *High-Performance IIR Filter Implementation Using FPGA*. IEEE, United States, pp.354-358.

Wang, Z., 2022. Different methods of linear phase IIR filer realization. *Highlights in Science, Engineering and Technology*, 27, pp.167-171.

Xiao, C., Olivier, J.C., and Agathoklis, P., 2001. *Design of linear phase IIR filters via Weighted Least-Squares Approximation. Proceedings* (*Cat. No. 01CH37221*). IEEE, United States, pp.3817-3820.

Zhao, H., 2022. Analysis of FIR Filter and IIR Filter for Human Speech Signal different Effects. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, China*, pp.272-280.