الجمهوريـــة الجـــزائريـة الديمـــــقراطيـة الشـــــــعبيـة

Republique Algerienne Democratique et Populaire

وزارة التـــعـليـم العـــالي والبـــحث العــــلمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــامعـــة الشهيـــد الشيــخ العربـــي التبســـي – تـبســـــة

Université Echahid Cheikh Larbi Tébessi – Tébessa

Faculté des Sciences et de la Technologie

Département de d'Électronique et Télécommunications

MEMOIRE

Présenté pour l'obtention du **diplôme** de **Master Académique**

Filière: Électronique

Spécialité: Instrumentation

Par: TIDJANI Hichem

DOUBIA Abd Ennour

THEME

Implementation of FIR filter using an innovative windows for speech signal

Présenté et évalué, le : 12/06/2024, par le jury composé de :

		Grade	Qualité
Dr.	M. MAAMRI	MCA	Président
Dr.	H. NEZZARI	MCA	Rapporteur
Dr.	A. AOUICHE	MCA	Examinateur

Promotion : 2023/2024

بسم الله الرحمن الرحيم

Dedicates

I warmly thank my family and friends, who contributed to raising my morale while I was working on the dissertation, and have always supported and encouraged me throughout this time of research. T.H

Today, as I stand on the brink of a new chapter in my life, I am overwhelmed with gratitude and joy. This momentous occasion would not have been possible without the incredible support of my family, friends, and faculty.

I am deeply thankful to each and every one of you. D.A.E

Acknowledgement

We sincerely thank everyone who contributed to the realization of this project, more particularly:

we wish to express gratitude towards our supervisor Dr. Hassen Nezzari for his unwavering encouragement, for his patience and support during the research and writing of this dissertation. Without his superb mentorship and willingness to help, this work would not have been possible.

Our sincere thanks also go to the members of the jury represented by both Dr. M. Maamri and Dr. A. Aouiche for the interest they have shown in our research by agreeing to examine our work and enrich it with their proposals. I would like also to thank our families (the Tidjani and Doubia families) for the inspiration, help and time they were willing to devote to us.

Table of Contents

List of Figure

List of Tables

ABSTRACT

FIR Filters are used, nowadays, in different applications because of their phase linearity as well as their intrinsic stability. This research introduces an innovative window-based near optimal FIR Filters design, especially for high order FIR Filters (complex mathematical calculations). And showcasing its effectiveness in enhancing speech signal quality by minimizing noise. In this research we try to rely on an innovative window called the Fractional order window whose characteristics depend on a parameter α. The fractional window function can adjust the spectral characteristics of FIR Filters, i.e., Ripple ratio, Main Lobe Width, and Side Lobes Ripple ratio by using two parameters: order 'N' and coefficient 'α'.

Order of window N! Yes, A characteristic of this window is that its main lobe and the amplitude of its side lobes also depend on the number of samples N, the Ripple ratio can be reduced to the lowest levels, something which does not exist for conventional windows. To performance appraisal of Fractional window in speech signals de-noising, FIR filter has been designed using Fractional window and compared with filter designed using Kaiser window. It has been shown that proposed window gives far better performance than Kaiser window function. The enhanced performance and computational efficiency of FIR filters with these innovate window make them particularly suitable for real-time applications. This research provides valuable insights into the optimization of FIR filters, laying the groundwork for future advancements in adjustable and hybrid filtering methods that promise to further refine speech signal processing and improve audio technologies.

ملخص

تُستخدم مرشحات FIR، في الوقت الحاضر، في تطبيقات مختلفة بسبب خطيتها الطورية بالإضافة إلى ثباتها الجوهري. يقدم هذا البحث تصميمًا مبتكرًا قائمًا على النوافذ بالقرب من التصميم الأمثل لمرشحات FIR، خاصة لمرشحات FIR عالية الرتبة (الحسابات الرياضية المعقدة). ويعرض فعاليته في تحسين جودة إشارة الكلام عن طريق تقليل الضوضاء. نحاول في هذا البحث الاعتماد على نافذة مبتكرة تسمى نافذة الترتيب الجزئي والتي تعتمد خصائصها على المعامل α. يمكن لوظيفة النافذة الكسرية ضبط الخصائص الطيفية لمرشحات FIR، أي نسبة التموج وعرض الفص الرئيسي ونسبة تموج الفصوص الجانبية باستخدام معلمتين: الترتيب "N" والمعامل "α".

ترتيب النافذة N! نعم، من مميزات هذه النافذة أن فصها الرئيسي وسعة فصوصها الجانبية تعتمد أيضًا على عدد العينات N، ويمكن تقليل نسبة التموج إلى أدنى المستويات، وهو أمر غير موجود في النوافذ التقليدية. لتقييم أداء النافذة الجزئية في إزالة ضوضاء إشارات الكلام، تم تصميم مرشح FIR باستخدام النافذة الجزئية ومقارنته بالمرشح المصمم باستخدام نافذة كايزر. لقد تبين أن النافذة المقترحة تعطي أداء أفضل بكثير من وظيفة نافذة كايزر. إن الأداء المحسن والكفاءة الحسابية لمرشحات FIR مع هذه النوافذ المبتكرة تجعلها مناسبة بشكل خاص للتطبيقات في الوقت الفعلي. يوفر هذا البحث رؤى قيمة حول تحسين مرشحات FIR، مما يضع الأساس للتطورات المستقبلية في طرق الترشيح القابلة للتعديل والهجينة التي تعد بمزيد من تحسين معالجة إشارات الكلام وتحسين التقنيات الصوتية.

INTRODUCTION

Signal processing is a vast and diverse field that emerged in the 20^{th} century. It's also a field that did not exist 75 years ago. Signal processing is not the transmission of signals, as through telephone wires or by radio waves, but the changes made to signals so as to improve transmission or use of the signals. Based on theories like those of Shannon and Fourier, it allows for precise manipulation of signals. Among the processes studied and designed by signal processing engineers are filtering, coding, estimation, detection, analysis, recognition, synthesis, recording, and reproduction [1].

The advancement of computers in the 1960s and 1970s had a positive impact on this discipline and led to the birth of digital signal processing (DSP). With the advent of more powerful computers and microprocessors, it became possible to process signals digitally rather than analogically, offering increased precision and flexibility. The first specialized DSP processors were developed in the 1980s [2], optimizing the mathematical operations necessary for signal processing. The implementation of digital algorithms like the Fast Fourier Transform (FFT) enabled rapid and efficient frequency analyses. Consequently, signal processing has become a vital technology in numerous fields: communications, information processing, consumer electronics, control systems, radar and sonar, medical diagnostics, seismology, and scientific instrumentation, among many others, laying the groundwork for even more advanced future developments.

Digital filtering is one of the fruits of this progression. Digital filtering is not simply about converting analog filters to digital filters; it is a fundamentally different way of thinking about signal processing, and many ideas and limitations of the analog method do not have equivalents in digital form [3].

The subject of digital filters is the natural introduction to the vast and fundamental field of signal processing. The basic power and simplicity of digital signal processing compared to the old analog methods are so significant that, where possible, we convert analog systems into equivalent digital forms. More importantly, digital signaling makes it easy to do fundamentally new things. The availability of modern integrated circuit chips, as well as micro-computers, has greatly expanded the application of digital filters.

Digital signals occur in many places. Telephone companies are rapidly converting to the use of digital signals to represent the human voice. Even radio, television, and sound systems are moving toward the all-digital methods since they provide such superior fidelity and freedom from noise, as well as much more flexible signal processing. Space imaging uses digital signaling to transmit information from planets to Earth, including extremely detailed images. Laboratory experiment recordings are now stored in digital form, from isolated measurements using a digital voltmeter to the automatic recording of entire sets of functions via a digital computer. Thus, these signals are immediately ready for digital signal processing to extract the message that the experiment was designed to reveal. Economic data are presented only in digital form. Digital filtering includes processes such as smoothing, prediction, differentiation, integration, signal separation, and noise suppression from a signal.

Digital filtering is ubiquitous in many modern audio applications, ranging from simple equalizers used in multimedia players to sophisticated signal processing systems in professional recording studios and communication devices. Digital filters can isolate and eliminate unwanted noise from an audio signal. Noise reduction algorithms are commonly used in communication systems and recording devices. They also allow the adjustment of the frequency characteristics of an audio signal to improve its perceived quality. For example, parametric equalizers can be used to boost or attenuate certain frequency bands to achieve a more balanced sound. The progress of digital signal processors (DSP) has allowed the implementation of digital filters in real time, enabling interactive applications such as electronic musical instruments and sound systems.

Digital filters can be classified into two major categories: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. In the literature, there are several methods for the synthesis and design of digital filters. This work focuses on the design of FIR digital filters synthesized by the window technique, intended for sound processing in general and speech as a particular case. This technique is based on the principle of limiting the data to be processed in the time domain with a weighting window, which leads to both positive and undesirable effects. These effects are mainly related to the type of used window. A good window is primarily characterized by the reduced width of the main lobe of its spectrum and the attenuation of its side lobes. The trade-off between these two contradictory characteristics, for specific applications, has led to the development of other innovative windows [4-11]. In this work, we will attempt to use one of these windows, which has not yet been explored or exploited, to filter audio signals using FIR digital filters [12].

The remainder of this thesis is structured as follows:

First chapter provides a general overview of digital signals and systems, explore the concept of data sampling and reconstruction process. Then, a brief presentation of digital filters and their design methods are presented.

Second chapter, briefly presents the principles of finite impulse response (FIR) filter design, studies the window method using different windows and shows the impact of their characteristics on the signal spectrum.

In the third chapter, a brief introduction to the speech signal is presented, as well as notions on the processing of this type of signals such as collection, sound effects and de-noising of the signal. Then, window function that we will use to synthesize an FIR filter, its origin, its spectral characteristics will be presented; following the same evaluation criteria as in the previous chapter. Low-pass FIR filters will be synthesized using this window in comparison with the Kaiser window. Finally, examples of speech sequence filtering and quantitative evaluations will be made.

Finally, a conclusion summarizes the findings and offers insights for future research directions.

CHAPTER I:
Fundamentals of Digital Signals and Systems

I.1. Introduction:

Digital signal processing has become a common tool for many disciplines. The topic below includes the methods of dealing with digital signals and digital systems. The techniques are useful for all the branches, which involve data acquisition, analysis and management. Before the digital era, signal-processing devices were dominated by analogue type. The major reason for DSP advancement and shift from analogue is the extraordinary growth and popularization of digital micro-electronics and computing technology [13].

Unlike continuous analog signals, digital signals are represented as discrete values, making them suitable for processing by computers. DSP enables efficient processing and offering greater flexibility and precision compared to traditional analog methods.

Digital signal processing (DSP) technology and its advancements have dramatically impacted our modern society everywhere. Without DSP, we would live in many less efficient ways, since we would not be equipped with voice recognition systems, speech synthesis systems, and image and video editing systems. Without DSP, scientists, engineers, and technologists would have no powerful tools to analyze and visualize data and perform their design, and so on [14].

In this chapter, we'll explore the basics of digital signals and systems in simple terms. We'll cover how digital signals are created through a process called sampling, and how systems manipulate these signals to perform various tasks. Understanding these fundamentals is essential for anyone interested in fields like digital signal processing, telecommunications, or data analysis.

I.2 Basic Concepts of Digital Signal Processing:

Digital Signal Processing (DSP) is a fundamental aspect of modern technology, shaping everything from audio processing to telecommunications and beyond. At its core, DSP revolves around manipulating digital signals to extract, enhance, or analyze information.

Key concepts in DSP include sampling, quantization, filtering, and transformation techniques like Fourier analysis. Understanding these basics lays the foundation for exploring the vast applications and advancements in digital signal processing.

I.2.1 Digital signals:

In our daily lives, analog signals appear as speech, audio, seismic, biomedical, and communications signals. To process a digital signal, the analog signal must be converted into a digital signal, that means analog-to-digital conversion (ADC) must take place, and then the digital signal is processed across many ways.

A typical digital signal x(n) is shown in Figure (1.1), where both the time and the amplitude of the digital signal are discrete. Notice that the amplitudes of digital signal samples are given and sketched only at their corresponding time indices, where x(n) represents the amplitude of the n'th sample and n is the time index or sample number [14]. We know that:

x (0): zero-th sample amplitude at the sample number n = 0
x (1): first sample amplitude at the sample number n = 1
x (2): second sample amplitude at the sample number n = 2
x (3): third sample amplitude at the sample number n = 3, and so on

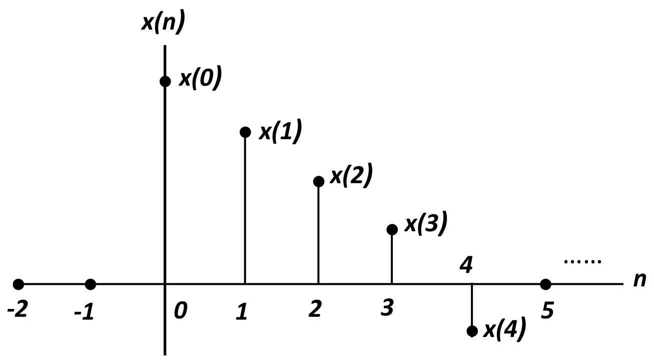


Figure 1.1: Plot of the digital signal samples.

In other concept, a discrete signal is a signal whose independent variable (time) is a discrete variable (instants are multiple integers of a duration Ts).

A digital signal is a discrete signal whose amplitude is discrete each sample is quantified (amplitudes are multiple integers of a quantity q).

I.2.2 Common Digital Signal:

Among the infinity of sequences that can be imagined, there are some that are widely used in signal and system analysis. We define and plot each of them as follows [14]:

a) Unit-impulse sequence defined by:

$$\&[\text{n}] = \begin{cases} 1 & if \ \text{n} = 0 \\ 0 & if \ \text{n} \neq 0 \end{cases}$$

Any sequence can be considered as a sum of shifted pulses [k-n] and amplitude x[n]. The sequence x[k] can therefore be described by the following expression:

$$\text{x}[\text{n}] = \sum_{k=-\infty}^{+\infty} \text{x}[\text{k}].\&[\text{n} - \text{k}] \ \ldots\ldots\ldots \ (\text{I}.1)$$

b) Unit-step sequence defined by:

$$u[n] = \begin{cases} 1 & if \ \ n \geq 0 \\ 0 & if \ \ n < 0 \end{cases} \quad \text{Ou} \quad \text{u}[\text{n}] = \sum_{l=-\infty}^{n} \&[\text{n} - \text{l}] \ \ldots\ldots\ldots \ (\text{I}.2)$$

c) Exponential waveform:

$$x[n] = a^k$$

d) Sinusoidal waveform:

$$\text{x}[\text{n}] = \text{A}.\sin\left[\frac{2\pi}{N}(n - n_0)\right] \ \ldots\ldots\ldots \ (\text{I}.3)$$

With N is signal period (integer)

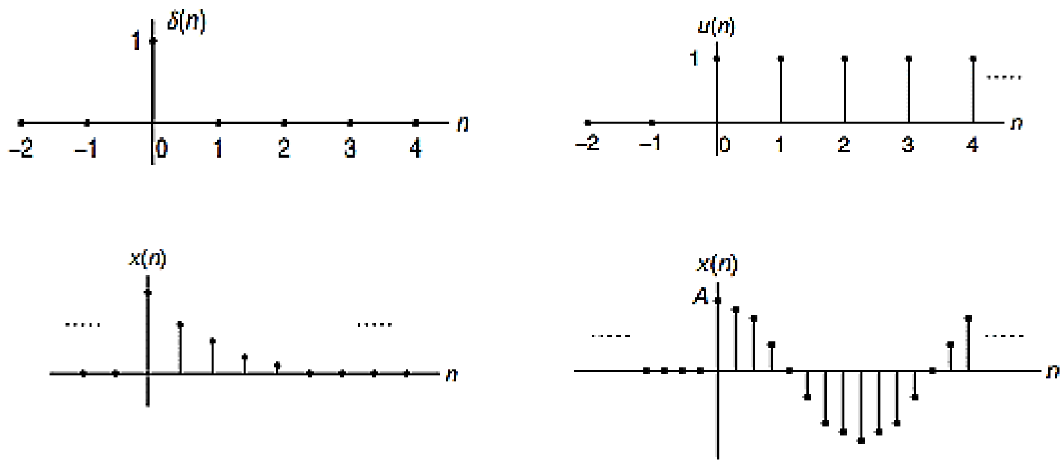


Figure 1.2: respectively, Plots of samples of digital sequences above

I.3 Digital system:

A digital system is predefined function that takes in a digital signal (the input) and gives out another digital signal (the output) or response of the system.

Such a system is defined mathematically as an operator or transformation that modifies an input sequence x[k] into an output sequence y[k]. We can represent this transformation by an operator F such that y[k] = F{x[k]} and give the equation or its functional diagram [15].

I.3.1 Systems properties:

In this section, we focus on properties such as linearity, time invariance, and causality.

I.3.1.1 linearity:

Suppose a linear system where $y_1(n)$ is the system output using an input x1(n), and y2(n) is the system output using an input $x_2(n)$. We find that the system output due to the weighted sum inputs $a.x_1(n) + b.x_2(n)$ is equal to the same weighted sum of the individual outputs obtained from their corresponding inputs, that is:

$$y(n) = ay_1(n) + by_2(n) \ldots\ldots\ldots \text{(I.4)}$$

Where a and b are constants [14].

I.3.1.2 Stability:

A system is stable if, whatever the finite amplitude sequence applied to the input, its output does not become infinitely large [15].

I.3.1.3 Static systems:

A static or memoryless system is a type of system where the output, y[k], only depends on the input signal at a particular time, k. This means that the output at any given moment doesn't rely on any past or future inputs, only on the input signal at that specific time, such as if you have a function that doubles the input value, it doesn't matter when the input occurred, only its value at the current time affects the output.

I.3.1.4 Dynamic systems:

A dynamic or memory system is one that considers not only the current input but also takes into account past inputs to determine the output. This means that the output depends on the input signal at that time as well as previous input signals or a sequence of inputs [14].

In a dynamic system like a digital filter, the output at a certain time depends on the current input as well as previous inputs. This is because the filter operates by processing a sequence of inputs over time, applying a set of rules or algorithms that involve memory of past inputs to produce the output.

I.3.1.5 Time Invariance:

Suppose a time-invariant system where $y_1(n)$ is the system output for the input $x_1(n)$.

Let $x_2(n) = x_1(n - n_0)$be the shifted version of $x_1(n)$ by n_0 samples. The output $y_2(n)$ obtained with the shifted input $x_2(n) = x_1(n - n_0)$ is equivalent to the output $y_2(n)$ acquired by shifting $y_1(n)$ by n_0 samples, $y_2(n) = y_1(n - n_0)$ [14].

This can simply be viewed as the following:

- If the system is time invariant and $y_1(n)$ is the system output due to the input $x_1(n)$, then the shifted system input $x_1(n - n_0)$ will produce a shifted system output $y_1(n - n_0)$ by the same amount of time n_0

I.3.1.6 Causality:

A causal system is one in which the output $y(n)$ at time n depends only on the current input $x(n)$ at time n, its past input sample values such as $x(n - 1), x(n - 2), \ldots$

- Otherwise, if a system output depends on the future input values, such as $x(n + 1)$, $x(n + 2), \ldots$, the system is noncausal. The noncausal system cannot be realized in real time [14],

I.4. Digital signals collection:

I.4.1. Sampling of Continuous Signal:

Figure (1.3) describes a simplified block diagram of a digital signal processing (DSP) system. The analog filter processes the analog input to obtain the band-limited signal, which is sent to the analog-to-digital conversion (ADC) unit. The ADC unit samples the analog signal, quantizes the sampled signal, and encodes the quantized signal levels to the digital signal [14].

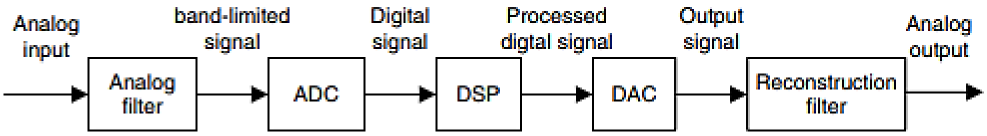


Figure 1.3: A digital signal processing schema

We start by discussing the concept of sampling processing in the time domain. Figure (1.4) illustrates an analog (continuous-time) signal defined at every point along the time and amplitude axes. This means the analog signal comprises an infinite number of points, making it impossible to digitize them all. Additionally, processing an infinite number of points is not feasible for digital signal (DS) processing. Sampling addresses this issue by taking samples at fixed time intervals, as depicted in Figures (1.4), where the time T represents the sampling interval or sampling period.

Each sample retains its voltage level throughout the sampling interval T to allow the ADC sufficient time for conversion. This technique is known as sample and hold. With one amplitude level per sampling interval, we can plot each sample's amplitude at its respective sampling time instant, as shown below [14].

For a given sampling interval T, which is the time duration between two sample points, the sampling rate can be calculated as: $fs = \frac{1}{T}$

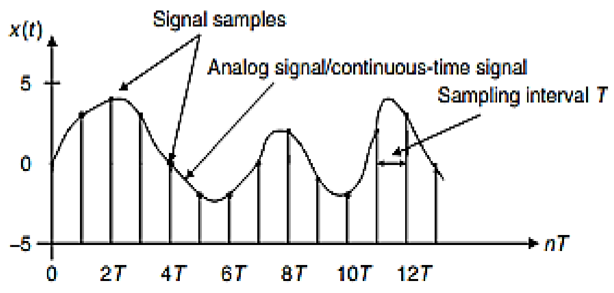


Figure 1.4: representation of the analog signal and digital samples

I.4.2 Expression of a sampled signal:

We have a continuous signal $x(t)$ that we will sample at specific times (kTs). The sampled signal denoted as $\breve{x}(\mathrm{t})$, can be represented as a sum of narrow rectangular pulses, each with a width of (Te), centered at the times (kTs), and an amplitude of $x(kTs)$. These pulses can be represented mathematically as unit-impulse, also centered at (kTs), with amplitudes equal to the area of the rectangular pulses. This leads us to the expression [16]:

$$\breve{x}(\mathrm{t}) = \sum_{k=-\infty}^{+\infty} \mathrm{Ts.\,x(k.\,Ts)\&[t-k.\,Ts]}$$

$$= \mathrm{x(t).\,Ts.} \sum_{k=-\infty}^{+\infty} \mathrm{\&[t-k.\,Ts]} = \sum_{k=-\infty}^{+\infty} \mathrm{x(t).\,Ts.\,U}(t) \ldots\ldots\ldots.(\mathrm{I.5})$$

U(t) as Unit-impulse.

I.4.3 Spectrum of the sampled signal:

The spectrum of the sampled signal is obtained by convolving the spectrum of the continuous signal with the unit-impulse comb spectrum. This results in:

$$\breve{x}(f) = Ts\,X(f) \otimes U(f) = X(f) \otimes \sum_{n=-\infty}^{+\infty} \&(f-nfs) = \sum_{k=-\infty}^{+\infty} X(f) \otimes \&(f-nfs)$$

$$X(f) = \sum_{n=-\infty}^{+\infty} X(f-nfs) \ldots\ldots\ldots.(\mathrm{I.6})$$

I.4.4 Sampling theorem:

We can retrieve the information of the original signal in the sampled signal, but it is necessary to translate this somewhat more formally. Sampling the signal therefore does not produce any loss of information if the sampling frequency f_s of the converter is greater than twice the highest frequency, which leads to “Shannon's Theorem”. Shannon's theorem, also known as the sampling theorem, is one of the foundations of digital signal processing. The principle of this theorem is: "Any function (t) whose spectrum has bounded support F(f)=0 for f>fm is completely defined by its samples f(N/fs) if fs>2fm. Sampling at a frequency fs of a function f(t) that violates Shannon's theorem does not produce spectral overlap."

If an analog signal is not appropriately sampled, aliasing will occur, which causes unwanted signals in the desired frequency band. The sampling theorem guarantees that an analog signal can be in theory perfectly recovered [17].

I.4.5 Sampling resolution:

Sampling resolution is the value of sampling which is a parameter use to measure sound wave change. It is a binary bit when the sound collecting, playing sounds file.

Sampling resolution and sampling frequency are two important indexes to the audio interface which is also a standard to choose an audio interface. No matter how sampling frequency is, theoretically the sampling resolution decides the range of sound intensity. Increase 1 sampling resolution amounts to range increase 6dB. The more resolution has the more accurate the signal collect is [18].

I.5.6 Signal Reconstruction:

Reconstructing the continuous-time signal from its samples involves adding the detected reproductions of the Fourier transform. However, this operation isn't always reversible. For example, if the continuous-time signal Cs(t) is real and has a non-zero component at frequencies ±fs/2, we'll encounter X(−fs/2) =X(fs/2), during sampling. When we add the shifted replicas of these real components in Y(−fs/2) and Y(fs/2), the information regarding the imaginary part of the continuous signal before sampling adheres to certain constraints.

It's important that a single-frequency component of the sampled signal doesn't originate from several components of the continuous-time signal due to the addition of the replicas of the Fourier transform of the time signal.

Furthermore, for each component of the sampled signal, it's crucial to know the frequency band from which it originated in the continuous-time signal.

Continuous-time signals here occupy a limited but different frequency band. The first continuous-signal Cs(t) satisfies Shannon's conditions; it doesn't exhibit spectral folding during sampling, and reconstruction is seamless. The second continuous-signal y(t) has a bandwidth wider than half the sampling frequency: reconstructing it through low-pass filtering of sampled signal doesn't yield the original signal.

The most natural constraint we can impose on the continuous-time signal to meet these conditions is to limit its frequency with linear filtering; ensuring its Fourier transform is zero outside the frequency band $-f$s/2 to $+f$s/2. In this case, the replicas of Y(w) don't overlap, and we know the initial frequency band of the continuous-signal Cs (t), which calculates the signal between samples, this is known as Shannon interpolation, and it's exact if the previous condition is met, therefore, if Shannon's condition is satisfied, sampling becomes a reversible operation [17].

I.5 Frequency representation:

Studying the frequency content of a signal involves representing it as a combination of sinusoids To assess the importance of the cosine component $(2\pi f_0 k)$ at frequency f_0 in the $x(k)$ signal, one common approach is to calculate the scalar product between $x(k)$ and cos $(2\pi f_0 k)$. This is done through the discrete-time Fourier transform, similar to how the Fourier transform works for continuous-time functions [19].

In practice, when dealing with digital computers, we face the challenge of evaluating the discrete-time Fourier transform. However, this is not straightforward because, in its definition, frequency is continuous within the interval $(0\ to\ 1)$[20].

I.5.1 Discrete Time Fourier Transform (DTFT):

By definition, the discrete-time Fourier transform of a sequence $\{x(k)\}$ is the function of f, periodic with period 1[21]:

$$X(f) = \sum_{k=-\infty}^{+\infty} x(k) e^{(-2j\pi f)} \quad \dots\dots\dots (\text{I}.7)$$

It is a continuous frequency function. It is customary to represent it on intervals (-1/2, +1/2) or$(0, 1)$, because of its periodicity.

The inverse formula is deduced:

$$x(k) = \int_{-\frac{1}{2}}^{\frac{1}{2}} X(f)e^{2j\pi f}\, df \dots\dots\dots (I.8)$$

I.5.2 Discrete Fourier Transform (DFT):

One of the disadvantages of the Fourier transfer account at a separate time (DTFT) for the value of the $x(k)$ sample is that can be extremely computationally intensive because it involves an infinite number of terms and the frequency f changes constantly over the period (0 to 1). This is why the concept of separate Fourier conversion was introduced [21]. It limits the calculation to a specific number of k values and f values. The practical value of this method often comes from the discovery of efficient calculation methods or when fast Fourier or FRT conversion methods are not available [22].

DFT is defined by the following relationship.

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{\left(-\frac{2j\pi kn}{N}\right)} \qquad n \in \{0, \dots, N-1\} \dots\dots\dots (I.9)$$

The reverse formula is:

$$x(k) = \frac{1}{N}\sum_{n=0}^{N-1} X(n)e^{(2j\pi kn/N)} \qquad k \in \{0, \dots, N-1\} \dots\dots\dots (I.10)$$

- Fourier's rapid transformation is an algorithm allowing the passage between time and frequency domains and that the FFT samples in time and frequency [15]. The equations that connect these two spaces are

$$\Delta f = {}^{1}/_{T} \qquad fmax \equiv fs = 1/\Delta t \equiv 1/Ts$$

I.6 Digital Filtering:

Figure (1.5) depicts a digitized noisy signal resulting from digitizing a random analog signal containing a useful low-frequency component and noise spanning the entire frequency range. Following sampling, the noisy signal (n) can be improved using digital filtering. Given that the useful signal contains a low-frequency component, any high-frequency components above that of the useful signal are considered noise, which can be eliminated using a digital low-pass filter. The diagram below shows a simplified digital signal processing (DSP) system operating as a basic digital low-pass filter. This filter processes the digitized noisy signal [14].

Figure 1.5: Digital filtering block

In Figure (1.6), the clean digital signal y(n) is obtained by applying the digital low-pass filter to the digitized noisy signal x(n). Common applications of noise filtering include obtaining clean digital audio [14].

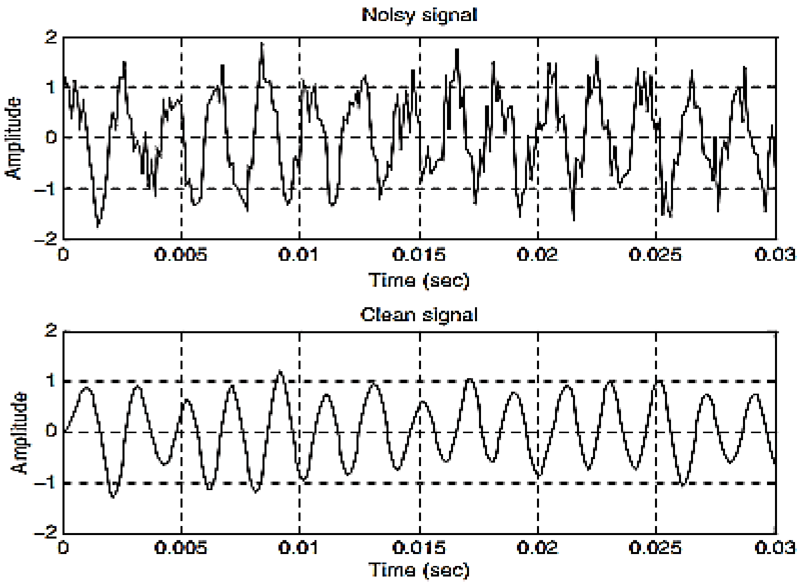


Figure 1.6: Respectively, Digitized noisy signal and clean digital signal

I.6.1 The Difference Equation and Digital Filtering:

We begin with developing the filtering concept of digital signal processing (DSP) systems. With the knowledge acquired to describe and analyze linear time-invariant systems. A DSP system (digital filter) is described in diagram below.

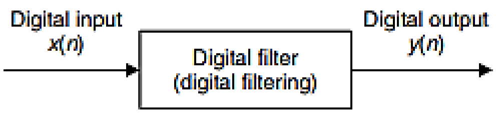


Figure 1.7: Digital filter representation

Let x(n) and y(n) be a DSP system's input and output, respectively, we can express the relationship between the input and the output of a DSP system by the following difference equation [14].

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) - a_1 y(n-1) - \cdots - a_N y(n-N) \ldots\ldots (I.11)$$

With: $b_i, 0 \leq i \leq M \; and \; a_j, 1 \leq j \leq N$

Equation can also be written as:

$$y(n) = \sum_{i=0}^{M} b_i x(n-i) - \sum_{j=1}^{N} a_j y(n-j) \dots\dots\dots (\text{I}.12)$$

We observe that output system is the weighted summation of the current input value x(n) and its past values: x (n -1), ..., x (n- M), and past output sequence: y(n-1),..., y(n-N).

The system can be verified as linear, time invariant, and causal. If the initial conditions are specified, we can compute system output (time response) y(n) recursively. This process is referred to as digital filtering [14].

I.6.2 Filtering Field:

I.6.2.1 Temporal filtering:

We define temporal filtering as the operation of interrupting or attenuating a signal. We assume a time signal $e(t)$ and a time filter $h(t)$ which cut the signal in the interval $[t_0 - T \quad t_0 + T]$, so the filtering result will be as follows:

$$S\,(t) = h(t).e\,(t) \dots\dots\dots (\text{I}.13)$$

Filtering temporally $e(t)$ by $h(t)$ produces the product: $e(t).h(t)$.

This operation generally has the effect of selecting a part of signal $e(t)$ by completely cutting or strongly attenuating the signal outside the selected duration.

This temporal filtering is generally called "weighting" where $h(t)$ is "the weighting function". The modification which causes this temporal filtering at the level of the frequency spectrum of $e(t)$ is given by applying Plancherel's theorem [17]:

$$S(t) = e\,(t).h(t) \xLeftrightarrow{TF} E(f) * H(f) \dots\dots\dots (\text{I}.14)$$

Consider the filter $h(t)$ (gate filter), which only lets the signal pass in interval between

$[t_0 - T \; t_0 + T]$, The Fourier transform:

$$H(f) = 2T \frac{sin2\pi fT}{2\pi fT}[cos2\pi t_0 f - jsin2\pi t_0 f] \dots\dots\dots (\text{I}.15)$$

If T is large enough, $H(f)$ can be compared to a Dirac pulse and the filtering effect will not be significant, but it nevertheless exists.

I.6.2.2 Frequency filtering:

The filtering operation that we have just applied to the temporal representation can also be applied to its frequency representation. We will speak of frequency filtering as the operation consisting of detecting, interrupting, attenuating all or part of the frequency components of a signal [19].

The spectrum $S(f)$ of the output signal is the product of the spectrum $E(f)$ of the input signal $e(t)$ and the frequency function of the filter $H(f)$:

$$S(f) = E(f).H(f) \ \dots\dots\dots (I.16)$$

The frequency representation of $e(t)$ is $E(f)$. $E(f)$ can be subjected to band-pass filtering (only frequencies included in the interval $[fo - Af \ fo + Af]$ are transmitted).

By virtue of Plancherel's theorem, we can replace a product in the frequency domain by a convolution product in the time domain

$$S(f) = E(f).H(f) \xLeftrightarrow{TF} s(t) = e(t) * h(t) \dots\dots\dots (I.17)$$

I.6.2.3 Convolution:

As we discussed before, the input and output relationship of a linear system in frequency domain is shown as

$$S(f) = E(f).H(f) \ or \ \dots\dots\dots (I.18)$$

The system output is the product of the spectrum input and transfer function [17].
But in time domain, the system output is given as

$$y(t) = h(t) * x(t) \ \dots\dots\dots (I.19)$$

where « * » denotes linear convolution of the system impulse response $h(t)$ and the system input $x(t)$. The linear convolution is further expressed as [14].

$$s(t) = e(t) * h(t) = \int_{-\infty}^{+\infty} e(t)h(t - t\tau)d\tau \dots\dots\dots (I.20)$$

This means that the convolution output will cover all possible positions where the two sequences overlap, including those where one sequence overflows the other.
It is important to note that the common filtering operations that we carry out are, in fact, convolution operations since we operate on signals in their temporal representation and not in their frequency representation [17].

I.6.3 Filter template (Filter model):

A filter template refers to a model or scheme used to provide a guide for designing and manufacturing a filter that meets the specifications required for a particular application.

- **Band-pass:** frequency range where gain takes values between $[1-\delta p\ , 1+\ \delta p]$where δp is the ripple rate.
- **Transition band:** frequency range where gain fades in A ratio.
- **Stop-band:** frequency range where gain takes values below δ.

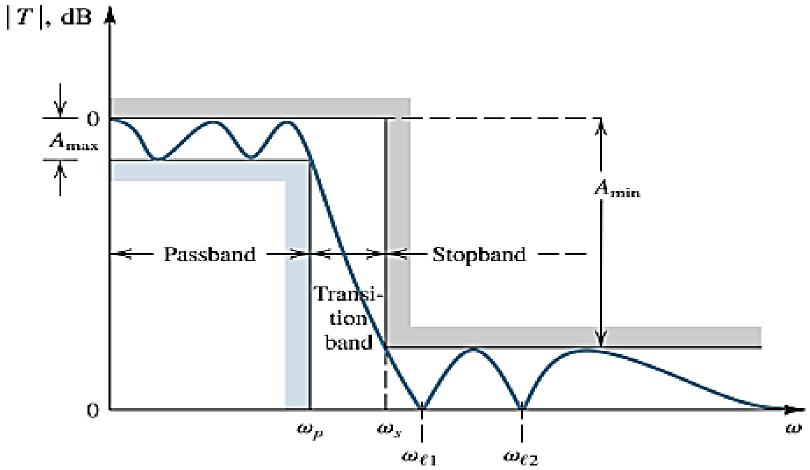


Figure 1.8: Filter model

I.6.4 Filter classification:

Filters can be classified into two broad categories depending on the duration of the filter's impulse response. These two categories are as follows [23].

I.6.4.1 Finite impulse response filter or FIR filter:

These filters are affected by the impulsive responses from the end. These casings are non-null in a new period of time.

The FIR filter is replaced by an adaptation due to differences such as:

$$\sum_{i=0}^{n} a_i y(k-i) \ = \sum_{i=0}^{m} b_i x(k-i) \ldots\ldots\ldots (\text{I}.21)$$

With $n=0$ and $m=N\ -1$

$$a_0 y(k) = \sum_{i=0}^{N-1} b_i x(k-i) \ldots\ldots\ldots (\text{I}.22)$$

Then become:

$$a \neq 0 \Rightarrow y(k) = \frac{1}{a_0}\left[\sum_{i=0}^{N-1} b_i x(k-i)\right] \ldots\ldots\ldots (\text{I}.23)$$

The transfer function is given by:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{a_0}\left[\sum_{i=0}^{N-1} b_i z^{-1}\right] \ldots\ldots\ldots (\text{I}.24)$$

I.6.4.2 Infinite impulse response filter or IIR filter:

These filters are characterized by impulse responses of infinite duration. In other words, the samples h(k) is non-zero over an infinite interval.

$$\sum_{i=0}^{n} a_i y(k-i) = \sum_{i=0}^{m} b_i x(k-i) \ldots\ldots\ldots (\text{I}.25)$$

With $\boldsymbol{n \neq 0}$

The transfer function is given by:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{m} b_i z^{-i}}{\sum_{i=0}^{n} a_i z^{-i}} \ldots\ldots\ldots (\text{I}.26)$$

- We can also classify it based on distinguishing between the various means of implementing a filter to carry out the filtering operation. We can distinguish two classes**:**

Non-recursive discrete filters: This means filters without feedback loop.

Figure 1.9: non-recursive discrete FIR Filter

Recursive discrete filters: This means filters with at least one feedback loop.

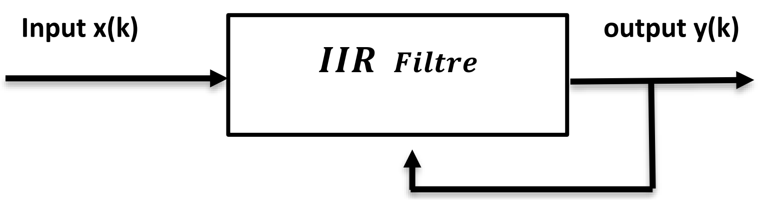


Figure 1.10: Recursive discrete IIR Filter

I.6.4.3 Comparison of filters:

The complexity of an IIR filter is less than that of a FIR filter of the same order. This property can be useful on platforms with limited computing power.

- Generally, FIR filters are less susceptible to quantification errors than IIR filters. The absence of recursion prevents cumulative errors.
- A FIR filter is less selective than an IIR filter of the same order. That is, the transition between bandwidth and rejected bandwidth is slower than in the case of the IIR filter.
- Unlike an IIR, a FIR filter can have a symmetrical impulse response and introduce a delay on the signal but no phase shift.

Overall, the choice between IIR and FIR filters depends on the specific requirements of the application. FIR filters are simpler, less prone to quantization errors, but less selective. IIR filters offer sharper cutoffs and can have symmetrical impulse responses but may be more computationally demanding and susceptible to quantization errors.

CHAPTER II:
Synthesis of Finite Impulse Response (FIR) Filters Using the Window Method

II.1 Introduction:

The exploration of signals and systems stands as a cornerstone in the curriculum of engineering institutions worldwide. Rooted in the pioneering contributions of luminaries like Fourier and Laplace, this field encapsulates a rich tapestry of mathematical principles. Over the past seven decades, the theory of signals and systems has evolved into a sophisticated framework, wielding profound influence across diverse domains of science and engineering. From power systems to communication networks, from automatic control to circuit design, and from filtering techniques to signal processing methodologies, its impact has been transformative, ushering in an era of remarkable technological advancement that permeates every aspect of our lives [14].

Digital filters are crucial tools in digital signal processing (DSP), and they are widely used in various applications including audio processing, telecommunications, and image processing. They alter discrete-time signals to either amplify or reduce specific components of the signal. A digital filter is a mathematical method that modifies a digital signal's properties. Unlike analogue filters, which process continuous signals, digital filters work with discrete signals, which are often represented as sequences of numbers.

Finite Impulse Response (FIR) filters are so Importance, one of the primary advantages of FIR filters is their inherent stability. An FIR filter's impulse response settles to zero in a fixed number of steps; hence there are no feedback loops that might cause instability. This predictable behavior assures consistent performance, which is critical in many applications.

FIR filters can also be built with a linear phase response, which means that all frequency components in the input signal are delayed by the same amount of time. This feature is especially significant in applications such as audio processing and data networking, where phase distortion can degrade signal quality. Designers can customize the filter characteristics to satisfy specific needs, such as low-pass, high-pass, band-pass, or band-stop filtering. This versatility makes FIR filters appropriate for a wide range of applications. The filter coefficients are straightforward to compute and apply to the input signal using convolution procedures. This simplicity translates into ease of use and decreased computing complexity in a variety of applications.

FIR filter design is a fundamental aspect of digital signal processing, involving key concepts such as impulse response, linear phase response, and filter coefficients. By understanding and applying these concepts, designers can create effective FIR filters for a wide range of applications, ensuring stability, precision, and desired frequency characteristics.

This chapter introduces principles of the finite impulse response (FIR) filter design and investigates design methods such as the window method, frequency sampling method and optimization method. Then the chapter illustrates in particular how to apply the designed Finite Impulse Response filters using different windows and shows the Impact of their characteristics on signal spectral.

II.2 Methods of FIR Filter Synthesis:

There are some design criteria to customize the performance of FIR. Thus, there must be some specific methods of FIR design corresponding to these criteria. The filter design methods are divided into three major categories: the frequency sampling method, the optimization-based method and the windowing-based method.

II.2.1 The frequency sampling method:

The frequency sampling provides another approach. Its notable characteristic lies in the ability to calculate filter coefficients based on specified magnitudes of the desired filter frequency response uniformly across the frequency range, offering design flexibility.

The development process begins by defining $h(n)$ for n=0, 1, ..., $N-1$ as the causal impulse response (FIR filter coefficients) approximating the FIR filter. Correspondingly, (k) for k=0, 1, ..., $N-1$ represents the discrete Fourier transform (DFT) coefficients. (k) is obtained by sampling the desired frequency filter response $(k)=H\ (e^{j\omega})$ at evenly spaced intervals in the frequency domain. Subsequently, adhering to the definition of the inverse discrete Fourier transform (IDFT), we can compute the FIR coefficients [14].

II.2.2 Optimization method:

In case it is necessary to create a filter without using any continuous filter as a starting point. It is possible to base it on an optimization procedure carried out on a computer. The starting point here is the definition of the tolerated errors between the desired frequency response $H_d(e^{j\theta})$ and the approximate version H$(e^{j\theta})$, which is characterized by the transfer function $H(Z)$.

$$H(Z) = \frac{\sum_{i=1}^{N} b_i Z^{-1}}{\sum_{i=1}^{M} a_i Z^{-1}} \dots\dots\dots (\text{II. }1)$$

From roughly estimated values of a_i and b_i and we can determine an error ε by comparing the desired and approximate functions at particular frequencies θi. (These frequencies can be chosen arbitrarily, they do not need to be evenly spaced, for example). The error criterion generally takes the form:

$$\varepsilon = \sum_{i=1}^{K} W(e^{j\theta i}) \left(|\mathrm{H}(e^{j\theta i})| - |H_d(e^{j\theta i})| \right)^{2p} \ldots\ldots\ldots (\mathrm{II}.2)$$

Where $W(e^{j\theta i})$ is a frequency-independent weighting function, which can be used to weight the error more heavily at some frequencies than at others, p is a constant whose value can be chosen [17].

II.2.3 Windowing based Method:

The theoretical definition of the inverse Fourier transform TFDI shows that an infinite number of terms is necessary to transform H(w) into h(n). So, if we were able to generate and use an infinite number of terms with TFDI, we could achieve the perfect filter function.

In practice, of course, only a finite number of terms of H(n) are actually used. That is, h(n) is obtained by truncating the original infinite sequence. This truncation obviously has harmful effects on the practical frequency response: it introduces in particular a distortion, it then proves necessary to apply compensation to the h(n) resulting from the truncation, this operation is called "windowing", it is sometimes associated with a "pre-deformation" of h (n). Generally, this "pre-deformation" is carried out by "passing" H(n) through a natural (Rectangular) window [17].

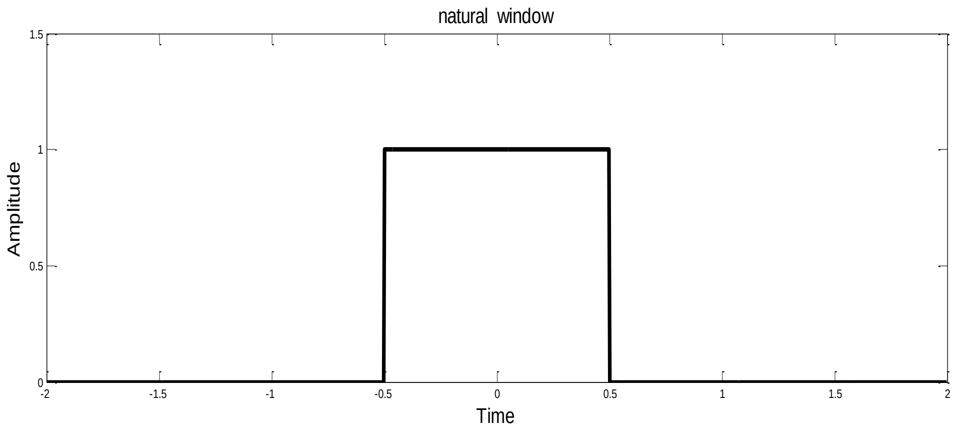


Figure 2.1: Natural (Rectangular) window

On the other hand, if it is necessary to operate the filter in real time, it can respect causality. For this it is necessary to delay the impulsional response which translates the frequency into a linear phase shift as a function of the frequency. The principle of the method can be summarized in the following points:

- Selection of an ideal template.
- Periodization of this template.
- Employing Fourier series decomposition to derive coefficients.
- Truncating to confine the sampled points.
- Adjusting the response to ensure causality, resulting in a causal filter.

II.3 Introduction to the Window Method:

II.3.1 Concept of Windowing:

Whether in the time domain or in the frequency domain, a window function is created each time there is a voluntary or involuntary limitation of the duration of observation, recording or use of a signal. This process is known as windowing. In digital systems, it's impractical to operate on an infinite number of data points. Thus, it becomes necessary to truncate the duration of observation or recording [14].

In the time domain, digital signals are acquired within finite acquisition times, meaning that the portion of the signal acquired and processed is defined over a finite duration. This limited knowledge of the signal can be conceptualized as multiplying it by a time window function called a rectangular window (natural window) represented by:

$$w_{Rect}(k) = \begin{cases} 1 \; ; \; 0 < k < N-1 \\ 0 \; ; \quad otherwise \end{cases}$$

The function $W_{Rect}(f)$, is the Fourier transform of the rectangular window and called spectral window

$$W_{Rect}(f) = \sum_{k=0}^{N-1} e^{-j2\pi fk} = e^{-j\pi f(N-1)} \frac{\sin \pi fN}{\sin \pi f} \dots \dots \dots (\text{II}.3)$$

This truncation of infinite signals to obtain signals of finite duration leads to undesirable effects. Essentially, a truncated signal emerges from multiplying a signal of infinite duration by a rectangular window:

$$x_N(k) = w_{Rect}(k)x(k) = \begin{cases} x(k) \; ; 0 < k < N-1 \\ 0 \quad ; \quad otherwise \end{cases}$$

This temporal multiplication results in the frequency domain by a convolution between the spectrum of the signal x(k) and that of the rectangular window. So:

$$X_N(f) = X(f) * W_{Rect}(f) = \int X(\tau) W_{Rect}(f-\tau) d\tau \dots \dots \dots (\text{II}.4)$$

Therefore, the spectrum of $x_N(k)$ will be different from the spectrum of the signal $x(k)$ in one hand. On the other hand, windowing involves applying a mathematical function to the signal data, which tapers the signal towards zero at its endpoints. This tapering helps to mitigate spectral leakage and other artifacts that may arise due to finite observation durations or discontinuities at the edges of the data. Common window functions include the Rectangular, Hamming, Hanning and Blackman windows, among others. Each window function has its own characteristics and trade-offs in terms of main lobe width, sidelobe suppression, and spectral resolution, allowing for tailored adjustments depending on the specific requirements of the signal processing task at hand [19].

II.3.2 Weighting windows:

Extensive research has been conducted to develop weighting windows that minimize spectral distortion, aiming for spectra closely aligned with desired characteristics. When comparing to the natural window, typical windows exhibit a widened central lobe but reduced oscillations. It appears that tightening the central lobe of a window leads to energy leaking into oscillating secondary lobes, known as spectral leakage [14]. Despite this, the compactness of frequency representation notably improves. This issue can be addressed by adjusting the window function's shape while aintaining the same duration observation N.

Here we represent the main windows [14,17,24], with their expressions w(k) and comparisons between them based on their characteristics (Table 2.1).

II.3.2.1 Rectangular Window (natural window):

In addition to what we mentioned earlier, a rectangular window is defined by its duration and its cyclical ratio. It can also be written in the form:

$$w_{Rect}(k) = rect\left(k + \frac{M}{2}\right) \text{ where k} \in [-\frac{M}{2}, \frac{M}{2} - 1] \ldots\ldots\ldots. (\text{II}.5)$$

In the case of a periodic signal, the spectrum is modified by a discrete Fourier transform $(\text{II}.3)$, and there is an array of functions $sinc(Nf)$ centers on the frequences that compose the initial theoric signal. This quality of the results may be recommended for the purpose of the spectrum, especially when it is composed of a large number of heads, including those from outside.

Truncation introduces oscillations in the spectrum of $x_n(k)$ Even with an increase in the number of truncation points N, these oscillations persist, with only their frequency changing, not their amplitudes. To mitigate the impact of the rectangular window, signals are often multiplied by alternative windows.

These windows should have a narrow central peak in their spectrum and significantly reduced side-lobes. In many scenarios, the rectangular window is considered too steep, leading to a preference for softer windows that produce fewer unwanted frequency effects.

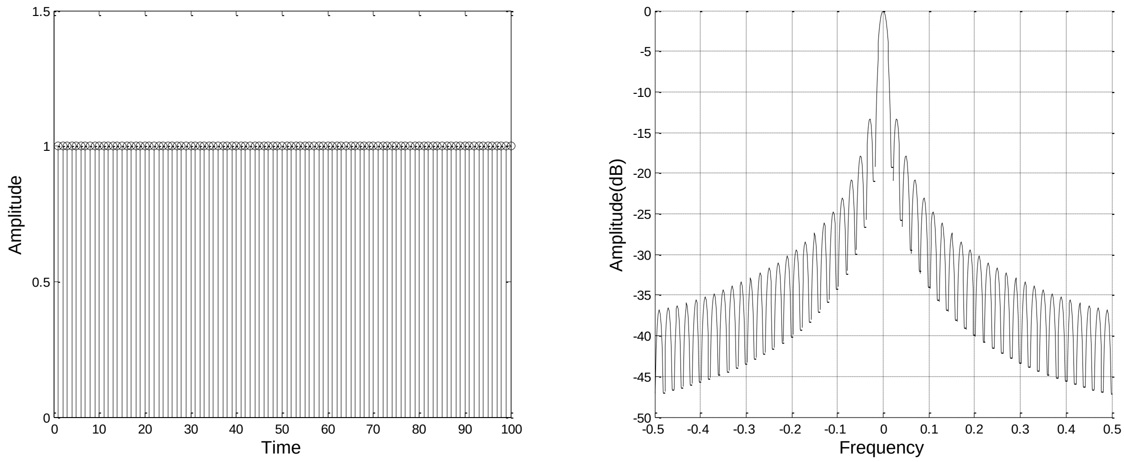


Figure 2.2: Representation of rectangular window in time and frequency domain.

II.3.2.2 Hanning Windows:

The Hanning window, also known as the Hann window or raised cosine window, is indeed a commonly used function in signal processing for various purposes including spectral analysis and noise reduction. It's particularly favored because it offers a good compromise between main lobe width and side lobe suppression. The mathematical expression for the Hanning window $w(k)$ of length N is given by:

$$w_{han}(k) = 0.5 * \ (1 + cos\ (\frac{2\pi k}{N-1}))\ \ where\ k = 0, 1 \dots, N-1 \dots \dots \dots . (\text{II}.6)$$

As before, $w_{han}(k)$ is identically zero outside the interval $[0, N-1]$. This window function is known as Hanning.We can generalize the window $w_{han}(k)$ with a parameter α as follows:

$$w_{han}(k) = \alpha - (1-\alpha\ )cos\ (\frac{2\pi k}{N})\ \ where\ k = 0, 1 \dots, N-1 \dots \dots \dots . (\text{II}.7)$$

The general form (II.7) is called the generalized Hamming window function.

- For α=1/2, we obtain the Hanning window.
- If α=0.54, the window obtained is called the Hamming window.
- The value α=1 corresponds to the rectangular window function.

Performing the convolution product in the frequency domain yields the result:

$$W_{han}(f) = \alpha\, W_{rect}(f) + (\frac{1-\alpha}{2})W_{rect}\left(\frac{f-1}{N}\right) + \left(\frac{1-\alpha}{2}\right)W_{rect}\left(\frac{f+1}{N}\right) \dots \dots \dots . (\text{II}.8)$$

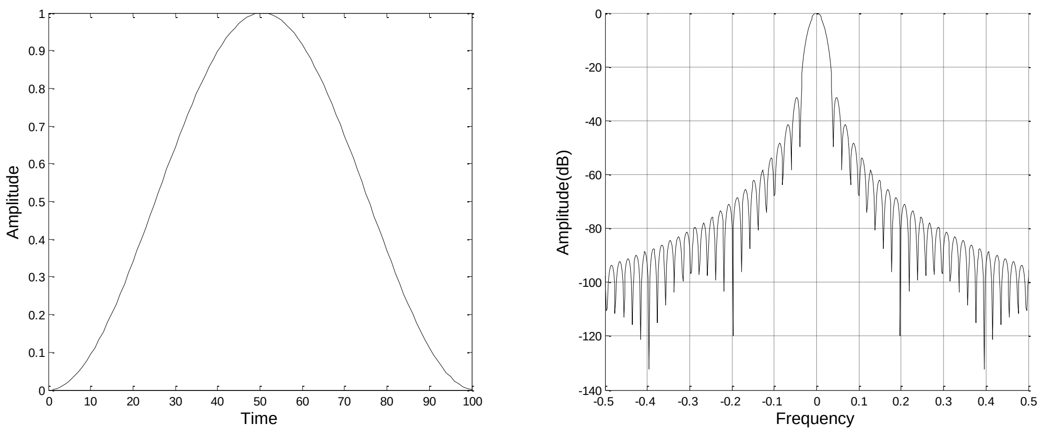


Figure 2.3: Representation of Hanning window in time and frequency Domain.

The Hanning window and its Fourier transform appear in Figure (2.3), it can be seen as one period of a cosine, so that its negative peaks just touch zero (hence the alternate name "raised cosine"). Since it reaches zero at its endpoints with zero slope. As a result, the side lobes roll off approximately 18 dB, the first side lobe has dropped from -13 dB (rectangular-window case) down to -31.5 dB, with a slightly stronger attenuation. The main-lobe width is of course double that of the rectangular window [16].

II.3.2.3 Hamming Window:

Hamming window is similar to the hanning window but with different weights, and it is widely used in speech processing, it is defined by:

$$w_{ham}(k) = \begin{cases} 0.54 - 0.46\cos\dfrac{2\pi k}{N-1} & where\ \ \mathrm{k} \in 0, 1 \ldots, N-1 \\ 0 & otherwise \end{cases} \quad \ldots\ldots\ldots.(\mathrm{II.9})$$

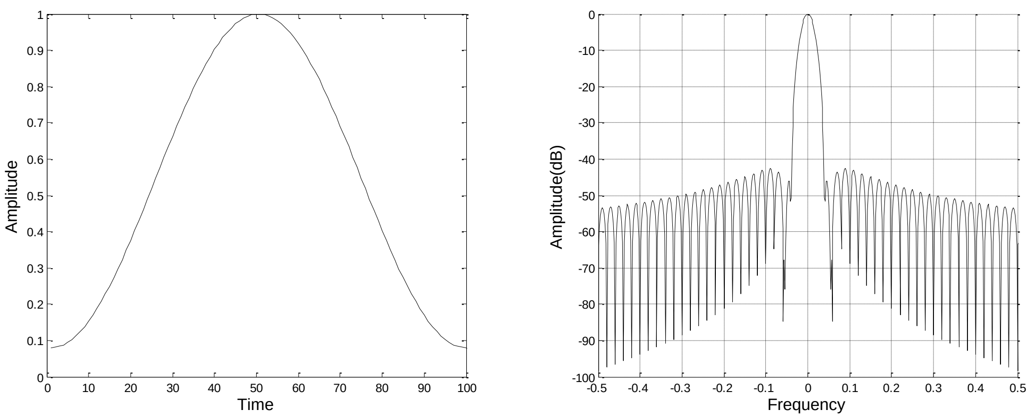


Figure 2.4: Representation of Hamming window in time and frequency Domain.

II.3.2.4 Blackman Window:

The Blackman-Harris window is similar to Hamming and Hanning windows. The resulting spectrum has a wide peak, but good side lobe compression. There are two main types of this window. The Blackman-Harris is a good general-purpose window, having side lobe rejection in the high 90 dB and a moderately wide main lobe. It also has all the dynamic range needed, but it comes with a wide main lobe. [24]. Blackman window $w_B(k)$ of length N is given by:

$$w_B(k) = \begin{cases} a_0 + 2\sum_{l=1}^{L} a_l \cos\frac{2\pi kl}{N} & k = 0, 1 \dots, N-1 \\ 0 & otherwise \end{cases} \dots\dots\dots (\text{II}.10)$$

where l = (N-1)/2. The coefficients a_l must satisfy the condition

$$a_0 + 2\sum_{l=1}^{L} a_l = 1$$

To avoid a multiplicative factor. The Fourier transform of the window $w_B(k)$ can be calculated as in the case of the $w_{han}(k)$window. We obtain:

$$w_B(f) = a_0 w_{Rect}(f) + \sum_{l=1}^{L} a_l w_{Rect}\left(f - \frac{1}{N}\right) + \sum_{l=1}^{L} a_l w_{Rect}\left(f + \frac{1}{N}\right) \dots\dots\dots . (\text{II}.11)$$

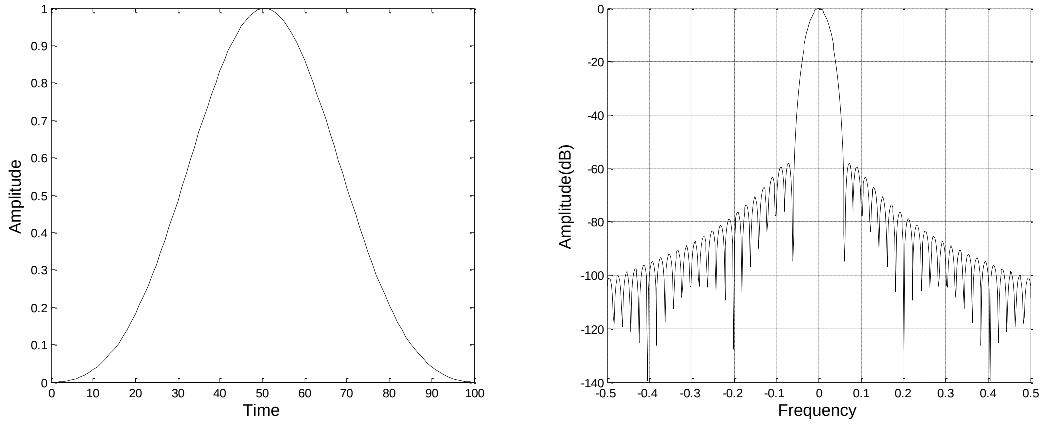


Figure2.5: Representation of Blackman window in time and frequency Domain.

II.3.2.5 Kaiser Window:

One popular parameterized window is the Kaiser window. This window makes it possible to specify in the frequency domain the compromise between the width of the central lobe and the amplitude of the secondary lobes through the values of the β parameter. The general form of this window is as follows:

$$w_k(k) = \begin{cases} \dfrac{I_0\left[\beta\sqrt{N^2 - 4(k - (N/2))^2}\right]}{I_0(\mathrm{Bn})} & where\ \mathrm{k} \in 0, 1 \ldots, N-1 \\ 0 & otherwise \end{cases} \ldots\ldots\ldots (\mathrm{II.}12)$$

The Kaiser window is specified differently from previous windows. Rather than the filter order, the amount of ripple and the width of the transition band are specified [15]. it is possible to calculate the values of α and β using the equations below.

Parameter α determines the tradeoff between main lobe width and side lobe levels of the spectral leakage pattern. The main lobe width is given by $2\sqrt{1 + \alpha^2}$.

$$\beta = \begin{cases} 0.1102(\alpha - 8.7) & \alpha > 50 \\ 0.5842(\alpha - 21)^{0.4} + 0.07886(\alpha - 21) & 50 \geq \alpha \geq 21 \\ 0 & \alpha < 21 \end{cases} \quad where\ \alpha = \frac{N-1}{2} \ldots\ldots\ldots (\mathrm{II.}13)$$

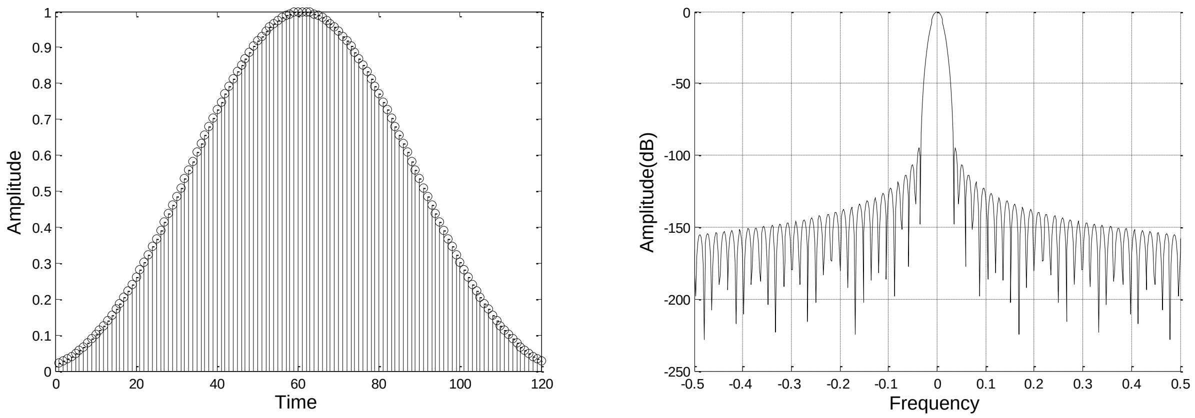
Once you have values for N and β, you can finally calculate the actual window weights. The Kaiser window equation makes use of another function I_0, where I_0 is the 0th-order modified Bessel function of the first kind.

$$I_0(x) = \sum_{k=0}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2 \ldots\ldots\ldots (\mathrm{II.}14)$$

A formula that predicts a reasonably accurate order estimate for a Kaiser windowed FIR is defined in terms of transition bandwidth (Δf), sample period (Ts), and passband error deviations δp and attenuation error deviations δa. The computed order estimate is given by

$$N = \frac{20\,log(\delta) - 7.95}{14.36\,\Delta f T_s}\ where\ \delta = \min(\delta p, \delta a) \ldots\ldots\ldots (\mathrm{II.}15)$$

A Kaiser-Bessel window strikes a balance among the various conflicting goals of amplitude accuracy, side lobe distance, and side lobe height. It compares roughly to the Blackman-Harris window functions, but for the same main lobe width, the near side lobes tend to be higher, but the further outside lobes are lower. Choosing this window often reveals signals close to the noise floor [25].



Figuer2.6: A Kaiser window where N=120, β=5.5981 in time and frequency Domain.

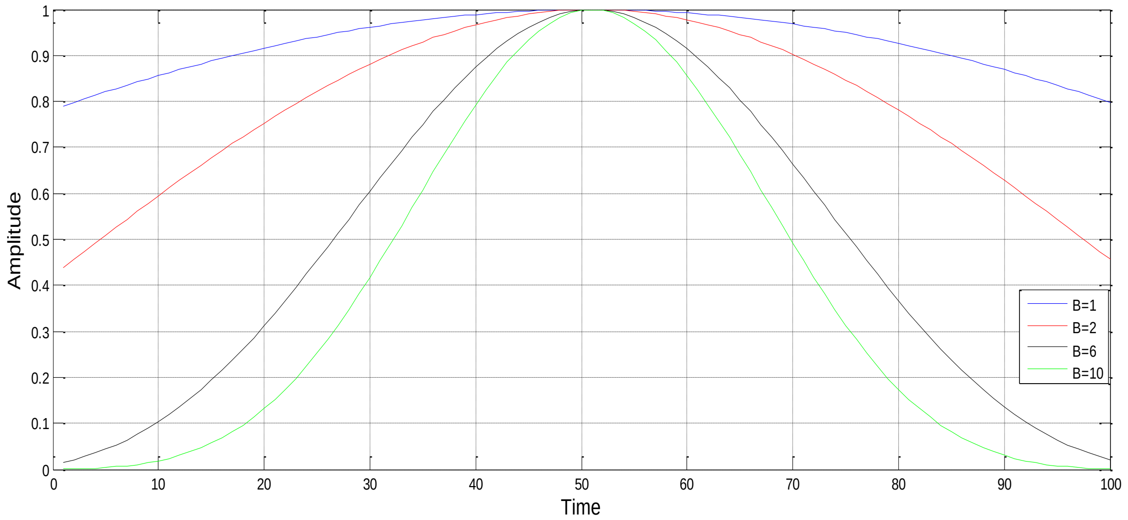


Figure 2.7: The Kaiser window N=100 for various values of parameter β.

II.3.3 Comparison of the weighting windows:

Different window functions can be compared based on their performance metrics such as main-lobe width, side-lobe attenuation, and overall filter shape. This comparative analysis helps in selecting the most appropriate window for a given application.

Despite having the sharpest transition band, the rectangular window has substantial spectral leakage and aliasing problems due to its inadequate side lobe attenuation. Better side lobe attenuation is achieved by Hamming and Hanning windows, but the main lobe becomes broader, which may impair frequency resolution. While the Blackman window is excellent at suppressing side lobes, it also has a much broader main lobe, which can lead to less abrupt transitions and possible signal distortion. Despite its great versatility and customization, the Kaiser window necessitates careful parameter β selection, which increases the computational load and complexity of the design process [17, 26].

Table2.1: Characteristics of the weighting windows.

window	Main lobe width	Second lobe amplitude	Sidelobe decay dB/oct
Rectangular	$^{2}/_{N}$	-13 dB	6 dB
Hanning	$^{4}/_{N}$	-32 dB	18 dB
Hamming	$^{4}/_{N}$	- 43 dB	6 dB
Blackman-Harris	$^{6}/_{N}$	-57 dB	18 dB
Kaiser-Bessel β=5.59816	$^{8}/_{N}$	-95 dB	6 dB
	(Depend on β)		

II.4 Effects of Window Functions on Signal Processing:

Window functions play a crucial role in digital signal processing, particularly in the design of Finite Impulse Response (FIR) filters and spectral analysis techniques. These functions are used to shape the frequency response of filters and to reduce spectral leakage in Fourier analysis. Understanding the effects of window functions is essential for achieving the desired trade-offs in signal processing applications [25].

II.4.1 Main Lobe Width and Side Lobes Level:

Analysis or weighting windows play a very important role in spectral observation. A window is characterized by:

- The width of its main lobe, it sets the resolution of the analysis, that is to say the ability to separate two frequencies close to each other.
- The amplitude of the secondary lobes, sets the dynamics of the analysis, ...the ability to measure the very different amplitudes of two frequency components relatively far from each other For a number of samples N fixed the rectangular window is the one which gives the best resolution; the width of the main lobe is equal to $2Fs/N$.

In the opposite, the rectangular window it does not allow good dynamics to be obtained because the secondary lobes decrease very slowly. In general, a window is characterized by the attenuation A (in dB) of the first secondary lobe. In the case of a rectangular window, A_{rect} is equal to -13 dB; this value is independent of the number of points N.

To obtain a better dynamic, there are other windows such as Hanning, Hamming and Blackman.

- Hanning, Hamming, and Blackman windows are known as trigonometric windows. We can see that all of these windows have a primary lobe broader than 2Fs/N (4Fs/N for Hanning and Hamming, and 6Fs/N for Blackman), therefore they have lesser resolution than the rectangular window. However, they have fewer secondary lobes than the rectangular window, resulting in improved dynamics.

The choice of a window depends on the frequency separation between the two components we wish to measure [15, 26].

II.4.2 Trade-offs Between Resolution and Leakage:

Understanding and managing the trade-offs between resolution and leakage is essential to achieving accurate and reliable spectral analysis in signal processing applications. By carefully considering these tradeoffs, systems that meet the desired performance criteria can be designed, while minimizing unwanted artifacts and distortions in the frequency domain.

Resolving Close Frequencies (frequency resolution):

To illustrate how each window affects the ability to discriminate very close frequencies, we have an example of separating two very close frequencies with a difference of ΔF=300 Hz. The amplitudes of the two components are equal (our example is two very close frequencies, f1=1000 Hz and f2=1300 Hz).

in MATLAB:

- Generate and display the composite signal.

- Calculate the frequency resolution of the FFT (Fs = 20 kHz et N = 200).

- Apply a rectangular window and then a Hamming window to the signal.

- Perform the FFT and check the resolution obtained [27].

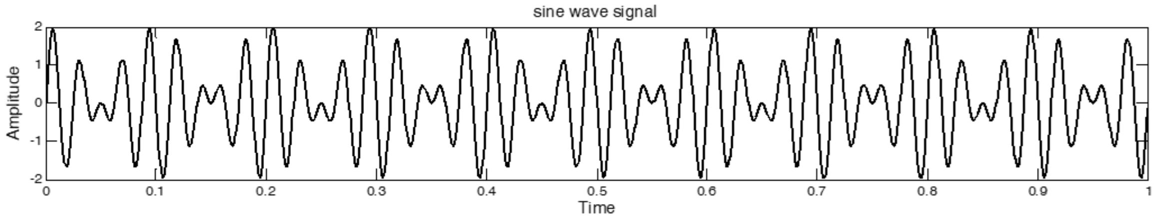


Figure 2.8: The composite signal.

The figures below illustrate the results obtained by MATLAB; they show the separation of two very close frequencies. Figure (2.9) shows the x(k) samples after passing through a rectangular window and also the x(k) samples after passing through a Hamming window.

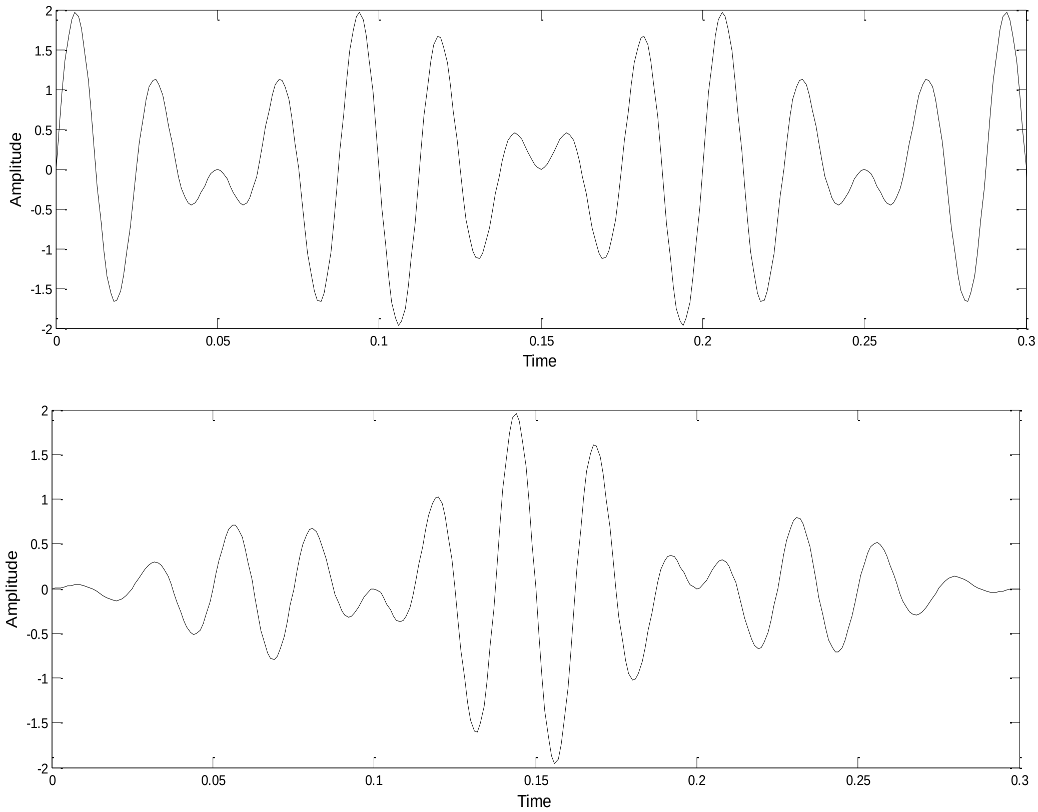


Figure 2.9: respectively, x(k) after passing through a rectangular and Hamming window.

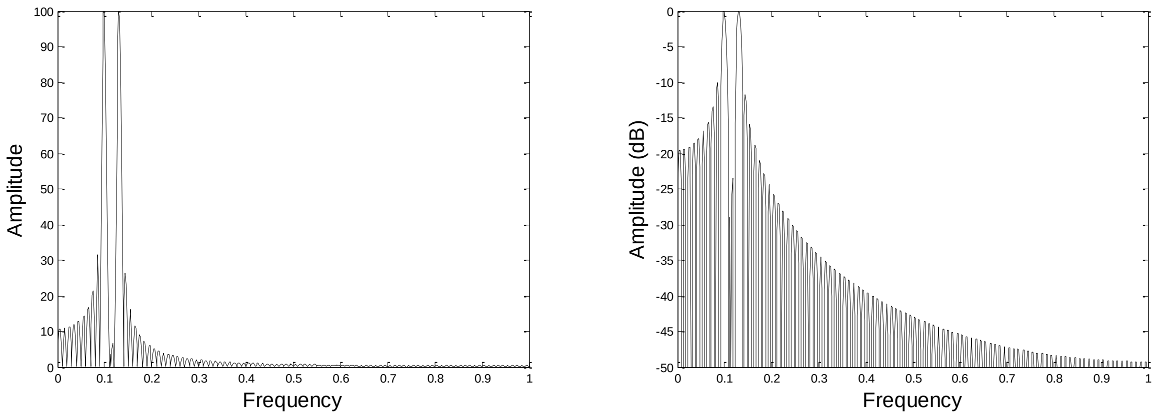


Figure 2.10: Rectangular window-weighted signal spectrum.

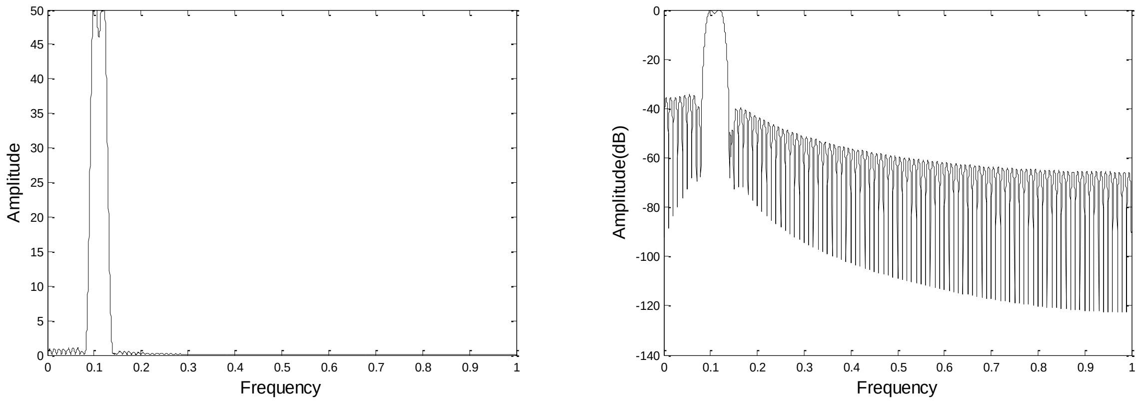


Figure 2.11: Hamming window-weighted signal spectrum.

The frequency resolution is given by the relation:

$$\Delta f = Fs/N \dots\dots\dots.(\text{II}.16)$$

The condition $\Delta F > Fs/N$ (ΔF: difference between the two close frequencies to be separated) is satisfied if ΔF is greater than twice the frequency resolution Δf in the case of rectangular window and four times in the case of Hamming window. Therefore, the width of the main lobe of the window function in the frequency domain determines the frequency resolution (The condition must be verified for the two frequencies to be separable).

From the figures (2.10) and (2.11) above, the signal spectrum shows the visualization of the frequencies. The two signal components weighted by the rectangular window are well observed because the relationship ΔF>2Fs/N (300 > 200) is verified (Fs = 20 kHz and N = 200).

In the case of the Hamming window, however, it is impossible to separate the two frequency components because ΔF<4Fs/N (300 < 400).

If ΔF is exactly equal to $2\Delta f$ or $4\Delta f$, the condition is not strictly verified, but we are at the limit of resolution. An increase in the number of N points may be necessary to better separate the frequencies [27].

- **Resolving Distant Frequencies (Dynamics Issue):**

Understanding the dynamics of distant frequencies is essential for designing effective signal processing tools and ensuring high-quality signal reproduction [28].

To illustrate how each window affects the ability to correctly represent distant frequencies with very different amplitudes, we have an example of separating Two distant frequencies, f1=1000 Hz and f2=6000 Hz, with very different amplitudes (ratio of 0.01). The windowing parameters must be appropriate. Sometimes adjusting these values can improve the results.

The figures below illustrate the results obtained by MATLAB; they show the separation of two distant frequencies (The ratio of the amplitudes of the two components is very large). Figure 2.12 shows the z(k) samples after passing through a rectangular window. Figure 2.13 shows the z(k) samples after passing through a Hamming window.

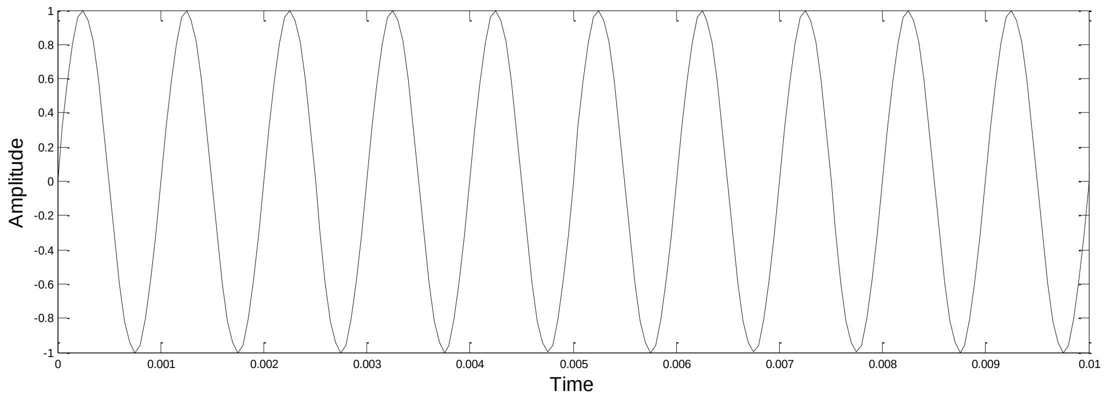


Figure 2.12: Rectangular window-weighted composite signal samples.

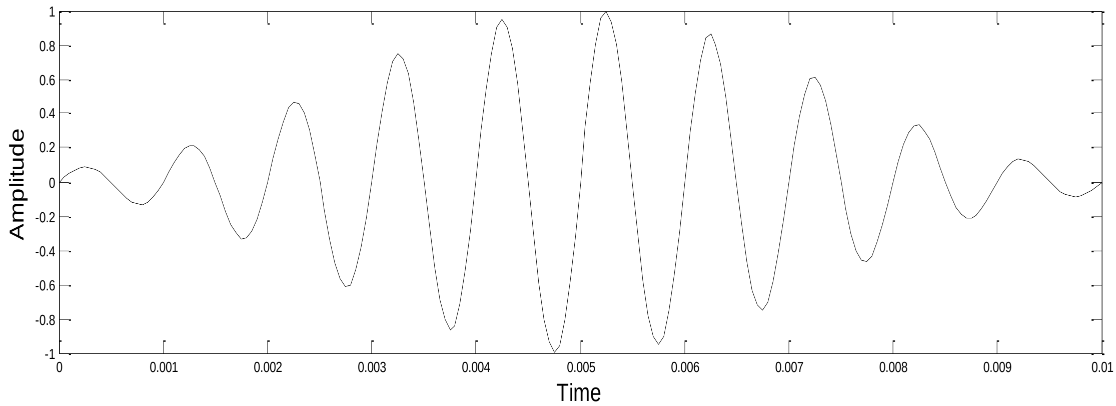


Figure 2.13: Hamming window-weighted composite signal samples.

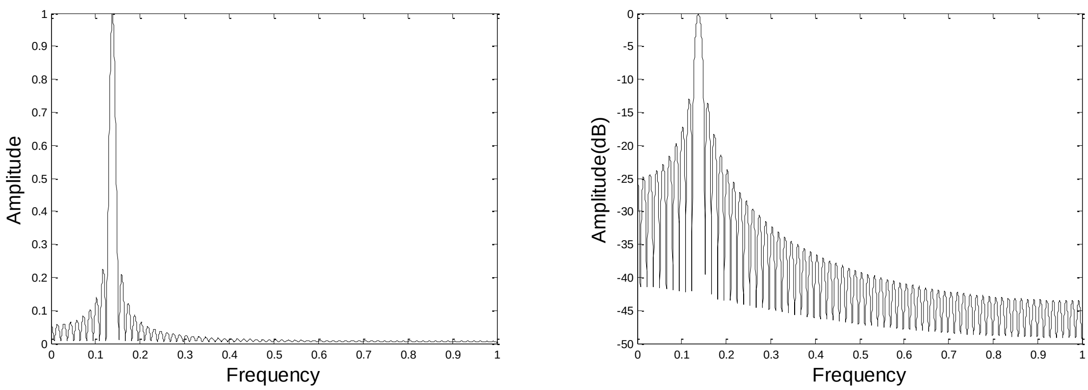


Figure 2.14: Rectangular window-weighted signal spectrum.

Figure 2.15 shows the spectrum of Z(n) weighted by the Hamming window. The use of a hamming window allows the components to be removed as clearly shown.

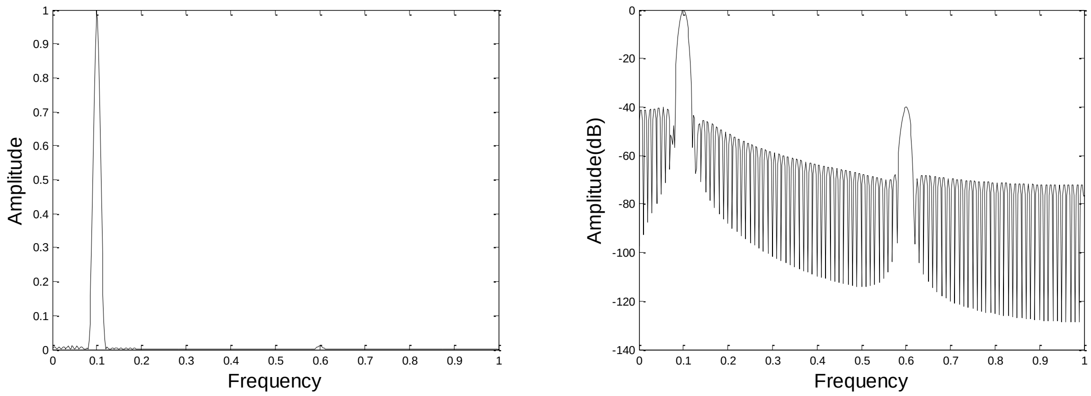


Figure 2.15: Hamming window-weighted signal spectrum.

Rectangular Window provides good resolution for close frequencies, but is less effective for frequencies with a large amplitude ratio due to high amplitude of its side lobes (The frequency components are not clear in figure (2.14).

Hamming Window has better attenuated side lobes, which is advantageous for distant frequencies with a large amplitude ratio, but may make it harder to distinguish very close frequencies.

- **Summary:**

When a window function is applied to another function or data sequence, the product is zero outside the interval, leaving only the portion where they overlap, effectively creating a "view through the window". In practice, the segment of data within the window is first isolated and then multiplied by the window function values. The primary purpose of window functions is to apply tapering to the data, rather than merely segmenting it [28].

The ability to distinguish between closely spaced frequency components of a signal is known as frequency resolution in signal processing, and it is essential for tasks like spectrum analysis. However, spectral leakage is a phenomenon in which energy from a signal at one frequency seeps onto adjacent frequencies during Fourier analysis, resulting in distortion can make it challenging to achieve greater frequency resolution. When the frequency of the signal differs from the frequency bins of the Fourier transform, this issue is more apparent. Thus, in situations where great frequency resolution is necessary. It is clear how the different window functions affect spectral leakage and frequency resolution, which will help to make an informed decision based on the needs and intended applications. The dynamic range of the window affects its ability to distinguish between weak and strong signals. A window function with a high dynamic range preserves the amplitude accuracy of weak signals in the presence of strong signals.

Different window functions provide varied widths of the main lobe in the frequency domain. A thinner main lobe improves frequency resolution, allowing for the separation of closely spaced spectral components. They also influence the amplitude of side lobes, which are tiny peaks located on each side of the main lobe. Lower side lobe levels limit spectral leakage, increasing the accuracy of spectral analysis.

The choice of window affects several aspects of signal analysis (Spectral):

-The main lobe width of the window function in the frequency domain determines the frequency resolution. Narrower main lobes result in better frequency resolution, side-lobes occur around the main lobe and can cause spectral leakage (Spectral leakage occurs when the frequency content of a signal extends beyond the frequency bin it occupies), which can distort the amplitude and frequency characteristics of neighboring signals.

-Lower side-lobe levels are desirable to minimize this effect. (Reducing the amplitude of signal components near the edges). In typical applications, the window functions used are non-negative, smooth, "bell-shaped" curves. Rectangular and other functions can also be used.

The effect of windowing in Fourier analysis is crucial for understanding how different types of windows impact the resolution and dynamics of the spectrum obtained by the Fourier transform [27].

II.5 Effects of Window Functions on FIR Filters:

As previously stated, the window clearly influences the spectrum analysis of data using the Fourier transform. In this section, we will look at how Window Functions affect FIR Filters and how to combine the Finite Impulse Response (FIR) filter synthesis approach with windows to generate a digital model of a low-pass filter.

II.5.1 Importance of Window Functions in FIR Filter Design:

Window functions are essential for the design and performance of Finite Impulse Response (FIR) filters in digital signal processing. They regulate the frequency response of FIR filters by balancing the trade-off between main-lobe width and side-lobe level, affecting the filter's capacity to differentiate between closely spaced frequencies while suppressing undesirable ones.

Furthermore, window functions diminish the Gibbs phenomenon, which causes oscillations at discontinuities when truncating an ideal filter's impulse response, by gently tapering the endpoints of filter coefficients. The window approach simplifies FIR filter design by making it simple to execute and comprehend. It includes multiplying the ideal impulse response by a window function, which is useful in practical situations where computing economy is crucial [29].

II.5.2 Real FIR filter design using window functions:

An ideal low pass filter is defined by [30]:

$$H_d(\omega) = \begin{cases} 1 & if\ |\omega| \leq \omega_c \\ 0 & if\ \omega_c < |\omega| < \pi \end{cases} \quad \dots\dots\dots (II.17)$$

The impulse response is given by:

$$h_d(k) = \frac{\omega_c}{\pi}\frac{\sin(\omega_c k)}{\omega_c k} \quad \dots\dots\dots (II.18)$$

To synthesize a symmetrical, low-pass, real and causal filter, simply apply a rectangular window in a symmetrical manner on the ideal response of the filter, then shift it to make it causal. The truncation is carried out using a rectangular window of the form:

$$w(k) = \begin{cases} 1\ ;\ 0 < k < N-1 \\ 0\ \ ;\ \ \ \ otherwise \end{cases} \quad \dots\dots\dots (II.19)$$

This temporal multiplication $h(k) = h_d(k).w_{Rect}(k)$, results in the frequency domain by a convolution as [30]:

$$H(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\tau).W(\omega-\tau)d\tau \dots\dots\dots (II.20)$$

Where, $W(\omega)$ is the Fourier transform of rectangular window for the causal case; where a linear phase term appears; given as:

$$W(\omega) = e^{-j\omega\left(\frac{N-1}{2}\right)}\frac{\sin(\omega.N/2)}{\sin(\omega/2)} \quad \dots\dots\dots (II.21)$$

The amplitude and phase responses can be written in the following form:

$$|W(\omega)| = \frac{|\sin(\omega.N/2)|}{|\sin(\omega/2)|}, \quad |\omega| < \pi \quad \dots\dots\dots (II.22)$$

$$\theta(\omega) = \begin{cases} -\omega\,(N-1)/2\ ; & when\ \ \sin(\omega N/2) \geq 0 \\ -\omega\,(N-1)/2 + \pi\ \ ; & when\ \ \sin(\omega N/2) < 0 \end{cases} \quad \dots\dots\dots (II.23)$$

The width of the main lobe of |W(w)| will increase the width of the frequency response transition band of the resulting low-pass filter h(n), while the amplitude of the side lobes will introduce ripples and create side lobes in |H(w)|, the level of ripples is maximum near the cutoff frequency, and this maximum level is the same (in linear units, not dB) in the pass-band and in the stop-band. Using a window w(n) with smaller side-lobes allows the ripples and side-lobes of |H(w)| to be reduced. In return, the main lobe of |W(w)| becomes wider, which will increase the width of the transition band.

II.5.3 Impact of Window Length (Filter Order) on Performance:

To demonstrate the impact of window length (filter order) on the performance of FIR filters using a Hamming window, we can compare the frequency responses for different filter orders, such as N = 51, 101 and 401. Figure (2.16) demonstrates how different window lengths affect the transition band and overall filter performance. By experimenting with different filter orders and observing their impact on the frequency response, one can gain a deeper understanding of the trade-offs involved.

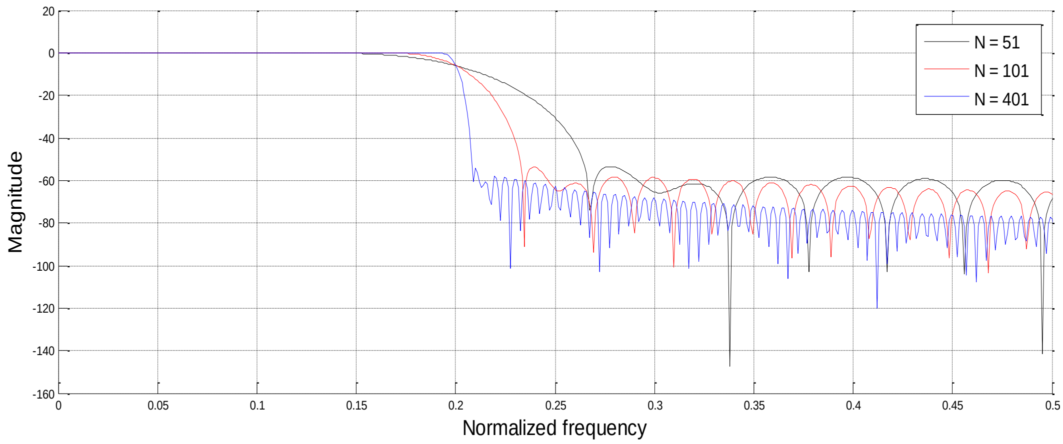


Figure 2.16: Frequency Response of FIR Filter for different orders.

By observation, it has been demonstrated that raising the filter order (window length) reduces the transition band in size and enhances FIR filter performance in terms of frequency resolution.

- **Summary:**

The length of the window, or equivalently the filter order, has a substantial impact on the performance of Finite Impulse Response (FIR) filters in a variety of ways. The filter order has a direct impact on the frequency resolution of the FIR filter, with a higher filter order (longer window) resulting in a narrower main lobe in the frequency response. This improves resolution of closely spaced frequencies and makes the filter more selective. The filter order also affects the transition band, which is the range of frequencies between the pass-band and stop-band. A higher filter order creates a smaller transition band, resulting in a sharper cutoff between the pass-band and stop-band. Furthermore, the side-lobe levels, which indicate the leaking of stop-band frequencies, are affected by both the filter order and window function. In general, raising the filter order decreases the relative amplitude of side lobes, resulting in improved stop-band attenuation.

II.5.4 Comparison of Low-pass FIR Filters made with Different Windows:

To compare FIR filters designed with different window functions. We'll focus on the following key aspects for each window function: main lobe width, side lobe attenuation, transition band sharpness, and overall impact on the filter's performance.

FIR low-pass filter was designed and tested using different windows. The filter's performance was evaluated by observation of frequency response of each window.

When designing the noise reduction filter, we specify a cut-off frequency of 150 Hz and a filter order of 50. We design a low-pass FIR filter using Rectangular, Hamming, and Blackman windows of length 51, which is known to reduce side lobes of frequency response and disparity in minimize noise leakage into the passband. Using a normalized cut-off frequency, we generate the filter coefficients to preserving signal components below this threshold. Applying this FIR filter to the noisy attenuate frequencies above 150 Hz while sinusoidal signal should significantly reduce noise and improve clarity. The effectiveness is confirmed by analyzing the filter's frequency response shown in Table (2.2), (2.3) and visualizing the original and filtered signals Figure (2.19). The filter was applied to the noisy signal.

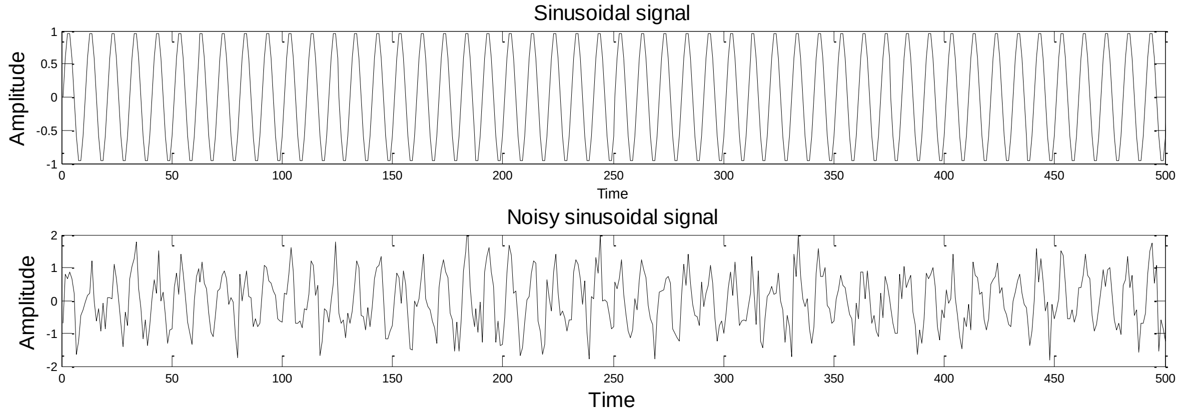


Figure 2.17: Both the noisy and the original signal.

The effect is very obvious when adding noise into the signal. This can be clearly seen in the time domain graph of figure (2.17); that it was nearly impossible to discern the original signal.

If the original signal contains specific frequency components that must be retained and other frequencies that must be removed, the cutoff frequency must be placed appropriately to separate these components. In this example, the original signal is a sinusoidal wave at 100 Hz, then selecting a different window each time, makes ensure that this sinusoidal wave is passed through the filter without being affected.

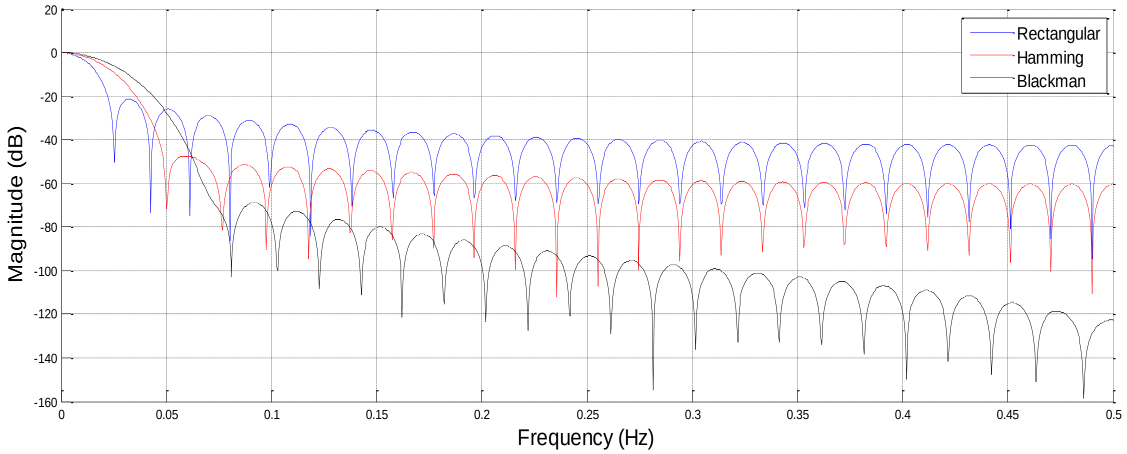


Figure 2.18: FIR filter's frequency response for different windows.

The windows found in Figure (2.18) are simply some of the more popular representatives from the class of fixed-coefficient windows. All of the windows, except the rectangular window, are seen to have sample values at or near zero at the "ends" of their support interval (k= N−1). This eliminates or reduces any potential jump discontinuities located at the ends of the window, since Gibbs phenomenon is attributed to such discontinuities, windows can suppress this type of distortion.

The key parameters of comparison include the realized transition bandwidth and pass-band and stop-band critical frequencies. the fact that the rectangular window has a locally weak stop-band attenuation. Other choices of windows, such as Hamming, Blackman offer better trade-offs between pass-band and stop-band gain. It can be seen that each window brings its unique set of performance parameters and produce different outcomes. Blackman window function provides higher side lobe attenuation compare to rectangular and Hamming windows however the main lobe width is slightly increased [26].

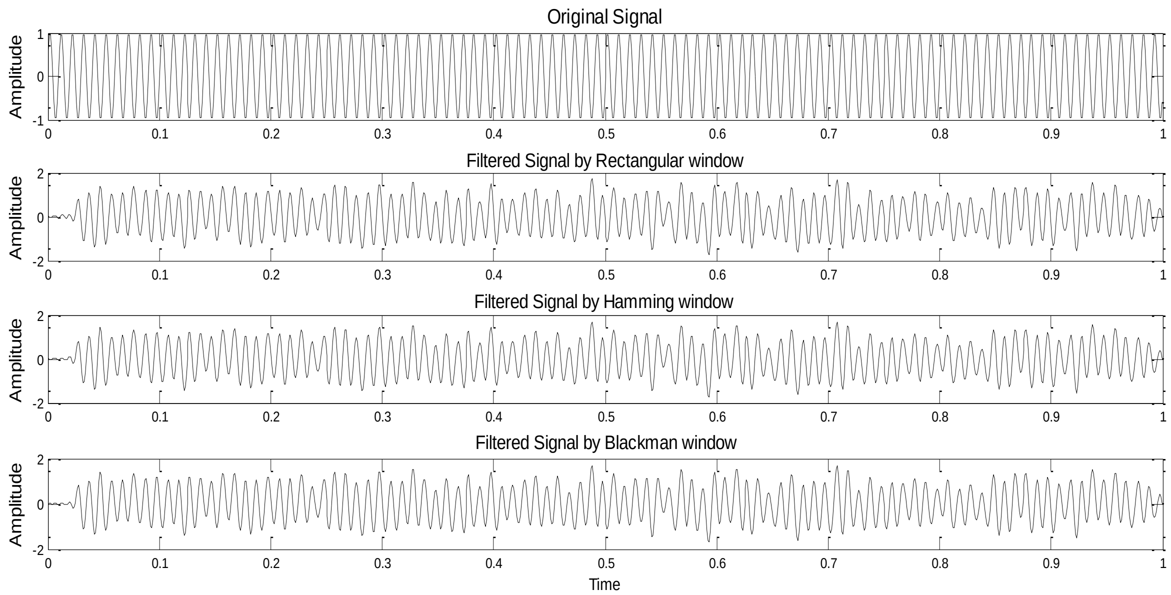


Figure 2.19: Respectively the original, and filtered signal for each windows.

After applying the filter and although the improvement after filtering was small, the filter was successfully efficient, while the frequency response of the filter showed efficient passing of frequencies below 150 Hz and attenuation of higher frequencies.

Choosing a cutoff frequency too close to the desired frequency might lead to distortions in the filtered signal. Therefore, a slightly higher cutoff frequency is selected to ensure the desired frequencies are passed without significant distortion.

The FIR low-pass filter was successfully designed and implemented using different windows. The filter demonstrated good capability in reducing noise from the noisy signal and improving signal quality. This process illustrates the impact of window characteristics on performance of digital Low-pass FIR filters can be used in signal processing to enhance signal quality.

Table 2.2: Evaluation of the frequency response for different windows.

Window	**Observation of FIR bands criteria**		**Observation of frequency response**
Rectangular	Transition Band	Sharpest transition band	-It shows the most ripple, but provides good cut-off stiffness despite some cut-band overshoot
	Passband	Steep roll off and significant passband ripples	
	Stopband	Poor stopband attenuation, with significant spectral leakage	
Hamming	Transition Band	Moderate transition band	-It shows the least ripple, but has very poor crossover stiffness because the frequency response does not reach the minimum attenuation required in the cut-off band
	Passband	Improved passband performance with reduced ripples	
	Stopband	Better stopband attenuation compared to the rectangular window	
Blackman	Transition Ban	Wider transition band	- It shows a slight continuous decrease in gain as a function of passband frequency, which can be annoying in one hand. On the other hand, the cut-off steepness obtained is deplorable, since it is with this window that the minimum cut-band attenuation is obtained
	Passband	Very low passband ripples	
	Stopband	Excellent stopband attenuation	

The key parameters of comparison include the realized transition bandwidth and pass-band and stop-band critical frequencies. the fact that the rectangular window has a locally weak stop-band attenuation. Other choices of windows, such as Hamming, Blackman offer better trade-offs between pass-band and stop-band gain. It can be seen that each window brings its unique set of performance parameters and produce different outcomes.

There are also several adjustable windows that augment the list of fixed coefficient windows presented in table (2.3). One popular adjustable window is the Kaiser window [24].

Table 2.3: Comparison of FIR's filter characteristics made with different Window.

Window Function	Main Lobe Width	Side Lobe Attenuation	Transition Band Sharpness	Side Lobe Roll-Off dB/dec.	Typical Applications
Rectangular	Narrowest	Poor (-21dB)	Sharpest	29	Rarely used due to poor stopband performance
Hamming	Moderate	Good (-41 dB)	Moderate	20	General-purpose filtering
Hanning	Moderate	Slightly better (-44 dB)	Moderate	60	General-purpose filtering with slightly better stopband attenuation
Blackman	Wider	Excellent (-57 dB)	Less sharp	60	High stopband attenuation needed
Kaiser	Adjustable	Adjustable (depend on β)	Adjustable	Adjustable	Custom requirements, versatile

II.6 Conclusion:

Windowing is a technique used in signal processing to mitigate spectral leakage and improve the frequency resolution of signals. When a signal is sampled, it is essentially multiplied by a rectangular window function, which abruptly ends at the edges.

FIR filters are a type of digital filter characterized by a finite number of coefficients and a finite impulse response. Unlike infinite impulse response (IIR) filters, FIR filters have a limited duration for their impulse response, making them inherently stable and free from feedback. This characteristic is particularly advantageous in applications where stability and phase linearity are essential. This simple method involves truncating the ideal infinite impulse response using a windowing function, such as a Hamming, Hanning or Blackman window. Although simple, the windowing method can result in less than optimal filter performance in terms of pass-band ripple and transition bandwidth.

A crucial approach in digital signal processing is the synthesis of Finite Impulse Response (FIR) filters using the window method. This method provides a reliable way to create filters with desired frequency characteristics. This chapter explored several synthesis techniques, with a particular emphasis on the window approach, and included a thorough review of FIR digital filters, emphasizing its particular significance. We explain the mathematical underpinnings and practical implications of these decisions by introducing the notion of windowing and clarifying the relevance of several window functions, namely Rectangular, Hamming, Hanning, Blackman, and Kaiser. The study also demonstrated how window functions impact resolution versus spectral leakage trade-offs, main lobe width, and side lobe levels in signal processing. Furthermore, the impact of window length on the design process's performance of FIR filter design is explored through filter coefficient calculation, frequency response analysis, and comparative examples of different windowed filters. Practical design considerations emphasize the selection of appropriate window functions based on application requirements. Ultimately, this chapter underscores the critical role of window functions in achieving optimal FIR filter performance, tailored to specific signal processing needs.

CHAPTER III:
De-noising Speech Signal Using an FIR Filter Based on an Innovative Windows

III.1 Introduction:

In speech signal processing, FIR filtering using window functions is essential for tasks such as noise reduction, frequency selective filtering, feature extraction, and anti-aliasing. It helps suppress background noise, enhance specific frequency components, shape the spectrum for better feature extraction, and prevent aliasing before down-sampling. However, this method has disadvantages including increased computational complexity and memory usage due to longer filter lengths, trade-offs between main-lobe width and side-lobe attenuation affecting frequency resolution, some residual spectral leakage, static filters that do not adapt to changing signal characteristics, and the challenge of selecting appropriate window functions and parameters. Despite these drawbacks, careful design and selection can optimize FIR filters for specific applications in speech processing. So, under these considerations, the implementation of FIR filters using innovative windows functions has become necessary for representing a significant advancement in speech signal processing. This approach will not only improve the quality and clarity of speech signals but also paves the way for more efficient and versatile audio processing solutions in the future [31].

In this chapter, we have applied the principles and techniques discussed in the previous chapters to a practical problem of reducing noise in speech signals. We mainly discuss how to design a suitable FIR filter to achieve a better filtering effect using an innovative window, denoted as fractional window and monitor its performance by observing the simulation results and playing back the speech signal before and after de-noising by MATLAB software. This chapter introduces the use of an innovative window to design FIR filters and demonstrates its effectiveness in improving speech signal quality by minimizing noise. Through this work, the aims were to advance both the theoretical understanding and practical capabilities of DSP, with a particular focus on improving the processing of noisy speech signals, from basic concepts through advanced filtering techniques to real-world signal processing solutions.

III.2 Speech Signal Processing:

Speech signal processing is a field that involves the analysis, synthesis, and manipulation of speech signals using digital and analog techniques. It encompasses a variety of tasks including speech recognition, speech synthesis, and enhancement of speech quality. The primary objective is to transform or interpret spoken language for various applications, such as improving communication in noisy environments, enabling voice-activated systems [32].

The speech signal processing chart is as below:

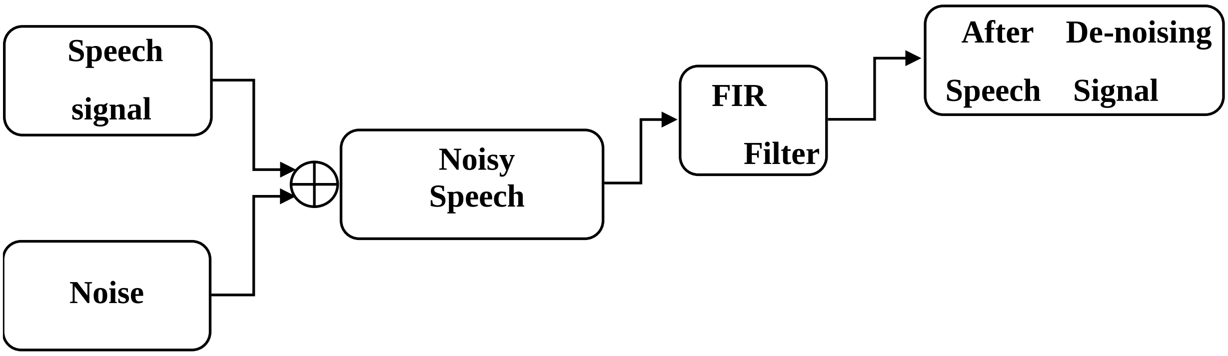


Figure 3.1: Speech signal de-noising diagram**.**

III.2.1 Speech Signal Collection:

A speech signal is a recorded representation of spoken language that can be modified, processed and examined. Typically, a microphone records an analogue signal, which is then converted to digital form using sampling and quantization for digital processing. Speech signals are defined by changes in amplitude and frequency over time and carry information that is used in many different fields, including audio enhancement technologies, speech recognition systems and telecommunications [13]. Human perception of speech signals in noisy environments is an important area of research in speech signal processing and has significant implications for various fields. However, it is worth noting that, despite the development that speech and audio processing systems have achieved, research in these fields is increasing to provide new and more efficient solutions in the areas mentioned above and several others, such as acoustic noise reduction to improve the quality, accuracy and interpretability of data in various applications, by using techniques such as filtering.

III.2.2 Noisy speech signal:

Signal noise is an unwanted component that interferes with the true representation of a signal; noise can significantly affect the quality and accuracy of signal analysis and processing. Understanding the sources and characteristics of noise, as well as methods for its reduction, is crucial for enhancing signal fidelity in various applications [18].

Noising a speech signal typically involves adding noise to the clean speech signal to simulate different real-world conditions. This can be useful for testing the robustness of speech recognition systems, improving speech processing algorithms, or studying the effects of noise during signal restoration using FIR filters.

Speech signal processing faces challenges from various noise sources that can degrade signal quality. Background noise, such as wind, traffic, and conversations, often accompanies the desired speech signal. Quantization noise arises during analog-to-digital conversion due to finite resolution. Channel noise introduces interference and attenuation during signal transmission. To study and manage these interferences, different types of noise are used including the Gaussian noise, which follows a normal distribution characterized by its mean and variance. Effective filtering techniques are crucial to mitigate the impact of these noise sources on speech signals.

The process of adding noise to a speech signal, entail carefully blending the clean speech signal with the noise signal. The waveform and frequency content of the speech signal can be altered by noise, which can also introduce distortions. This may have an impact on the signal's quality and naturalness.

To create a gaussian noise signal in MATLAB that has the same length as the original speech signal, use the formula "noise = 0.1 * randn (). In speech signal processing, Gaussian noise is commonly used to model random processes that affect the quality of speech signals during transmission or recording.

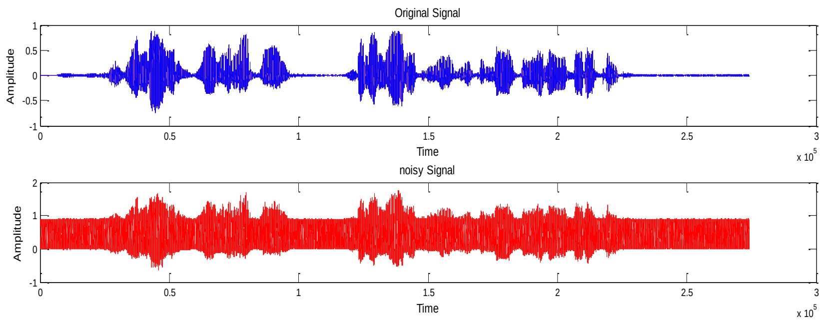


Figure 3.2: Both the noisy and the original speech signal.

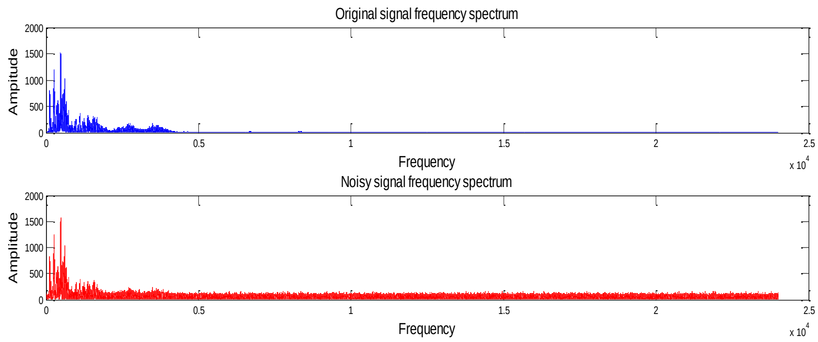


Figure 3.3: Frequency domain representation of the noisy and the original speech signal.

The effect is very obvious when adding noise to the signal. This can be clearly seen, in the time domain graph Figure (3.2); that it was nearly impossible to discern the original speech signal. In other hand, the speech signal becomes obviously noisy, the original speech content becomes blurred which is not easy to identify.

III.2.3 Distorting and Restoring a speech signal:

Human speech typically has a frequency between 300 and 3400 Hz. However, the gathered speech signal is frequently mingled with a great deal of noise, making it exceedingly challenging to filter the noise while maintaining the speech signal's quality. As a result, selecting the indicators that correspond to the filter's final filtering effect during design is especially crucial.

Effective noise reduction is essential for maintaining the integrity and clarity of signals in digital signal processing. By understanding the sources and types of noise and applying appropriate noise reduction techniques, it is possible to significantly improve signal quality. This is particularly important in applications such as speech recognition, telecommunications and audio processing, where high fidelity signal representation is critical [33].

In the real communication system, the information to be transmitted is often susceptible to noise pollution, it's very necessary to filter the noise before outputting the speech signal. The aim is to improve the clarity and intelligibility of speech by minimizing background noise and other unwanted components. FIR filters when designed with windowing functions can achieve this. Where using a low-pass filter to process a noisy signal demonstrated how specific noise components could be effectively removed, resulting in a significantly improved signal.

III.3 Key aspects of the innovative windows:

Finite Impulse Response (FIR) filters play a crucial role in digital signal processing, especially for applications like noise reduction in speech signals. Traditional window functions, such as Hamming, Hanning, and Kaiser, each offer different trade-offs between main lobe width and side lobe attenuation. However, these windows often fail to achieve the optimal balance required for specific applications [24], such as high-fidelity speech signal processing. The trade-off between these two contradictory characteristics, for specific applications, has led to the development of other innovative windows [4-11]. The fractional window is one of the innovative windows designed to achieve an adjustable shape, in order to have a compromise between the width of the main lobe and the attenuation of the secondary lobes [12]. In this work, we will attempt to use this window, which has not yet been studied or exploited; to synthetize FIR digital filters designed for de-noising audio Signal.

III.3.1 The fractional windows:

The following formula defines the fractional window [12]:

$$w_{fract}(k) = \frac{1}{2\pi} \frac{\sin[(1-\alpha)\pi]}{\left[cosh\left[\left(K - \left(\frac{N-1}{2}\right)\right)\alpha\right] - cos[(1-\alpha)\pi]\right]} \ldots\ldots\ldots (\mathrm{III.1})$$

Where, N is the number of samples (window length) and α is a real number such that 0 < α < 1. The α coefficient is used to vary the shape of the fractional window in order to modify its spectral characteristics (Figure 3.4).

The primary motivation for developing a new window function stems from the need to improve the performance of FIR filters in noise reduction and signal enhancement.

III.3.2 The fractional windows properties:

From the representation of the fractional window, we see that this one is characterized by the two parameters α and N, so we can draw the characteristics of this window according to these two parameters [12].

To illustrate the effect of increasing (N) on the window, we represent the Hamming window with a different number of samples, (N) is respectively (36, 71, and 120) Figure (3.4 (a)). The Hamming window shows that increasing the number of samples (N) leads to an extension of the duration of the window without modifying the distribution of the amplitude of the samples (0.08 for Hamming). Concerning the fractional window, Figure (3.4 (b)) shows that only the values of the two extreme samples vary in response to (N), while the values of the intermediate samples remain constant; where we note that the amplitude of the two extreme samples decreases with respect to (N) are respectively 0.5012, 0.128 and 0.0003.

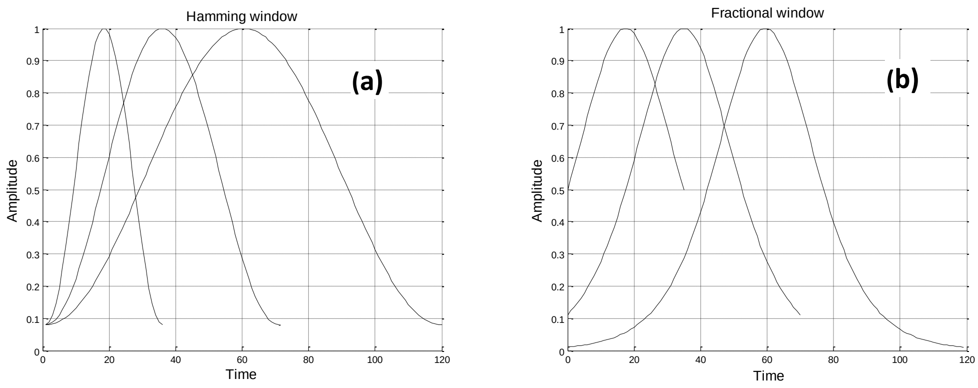


Figure 3.4: Effect of N on windows, **(a):** Hamming window, **(b):** Fractional window.

The number of samples has an effect on the shape of the window. One characteristic of this window is that the width of its main lobe and the amplitude of its secondary lobes also depend on the number of samples (N). Figures (3.5) and (3.6) show how the number of samples (N) affects the spectrum of the fractional window compared to the Hamming window.

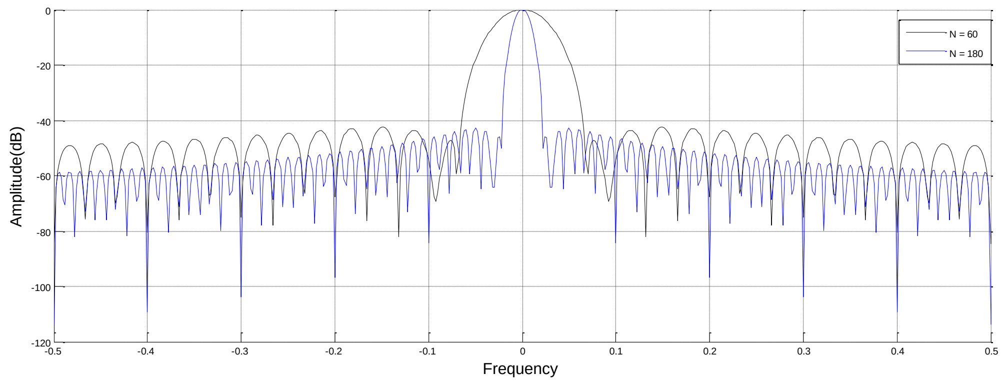


Figure 3.5: Hamming window’s frequency spectrum where N=60 and N=180.

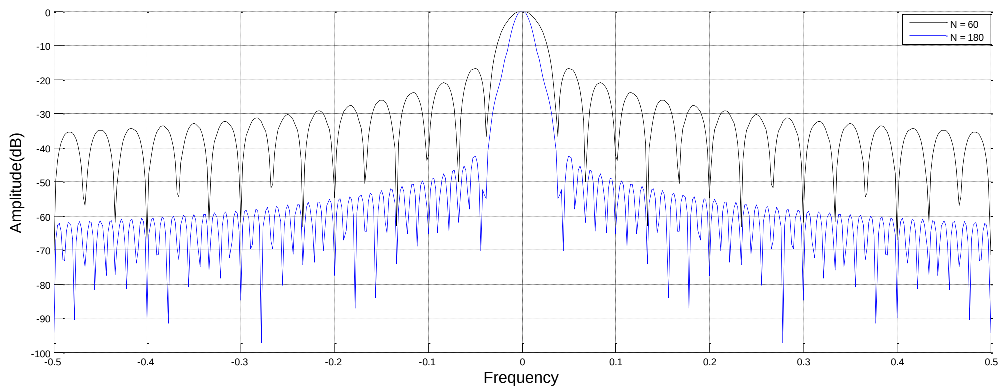


Figure 3.6: Fractional window’s frequency spectrum where N=60 and N=180.

Note that the Hamming window in Figure (3.5) maintains side lobes constant in amplitude, while the width of the central lobe decreases with the increase in (N). So, for the Hamming window, (N) only affects the width of the central lobe.

For the fractional window shown in Figure (3.6), the side lobes roll off approximately 6 dB with a slightly stronger attenuation for N=60, the first side lobe has dropped from -17 dB for N=60 down to -42 dB for N=180.

Noting that, the maximum amplitude of the secondary lobes and the width of the central lobe are crucial parameters in defining a window function's spectral characteristics. Concerning this window function, we can manipulate these properties by adjusting its shape using a fractional order coefficient

α. The spectrum of the fractional window will change accordingly for different values of α, as shown in Figure (3.8).

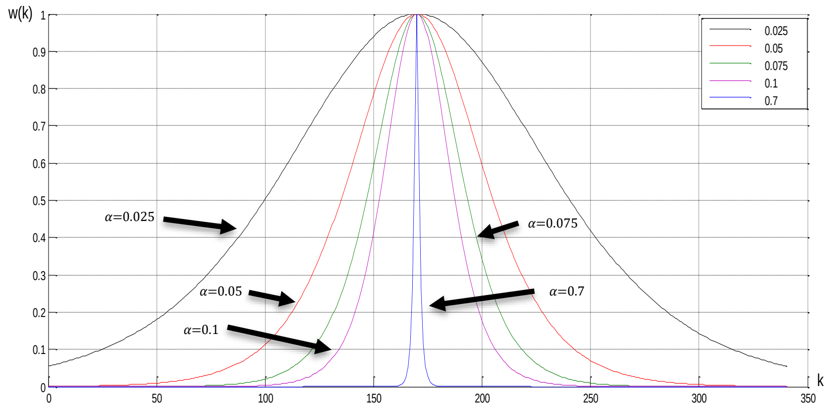


Figure 3.7: The fractional window chape for different values of α.

Fractional window spectrum represented by figure (3.8) shows that, the greater value of α increase the main lobe width and decrease the ripple ratio. The side-lobe roll-off ratio lobes approximately remain constant with the increment of the value of α. Results have been summarized in table (3.1).

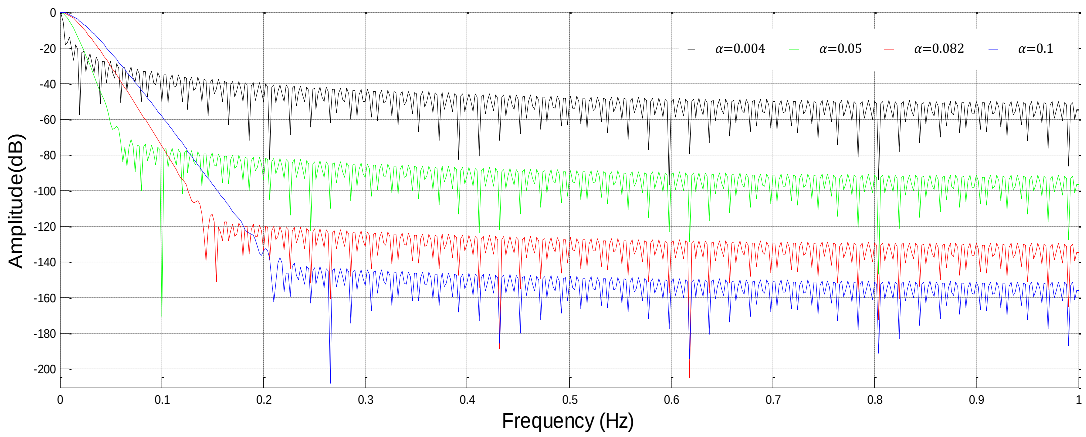


Figure 3.8: Frequency response of the fractional window for different values of α.

Table 3.1: Spectral parameters of fractional window for various α.

Parameter α	**Main-lobe width**	**Ripple ratio (dB)**
0.004	03.01/N	-14 dB
0.05	15.05/N	-65 dB
0.0823	44.50/N	-114 dB
0.1	60.90/N	-132 dB

- **Summary:**

Increasing number of samples N affects the Hamming and fractional windows in differ ways. While the Hamming window maintains side lobe amplitude constant, the fractional window attenuates them, providing better performance in terms of reducing unwanted artifacts. This makes fractional windows the best option for applications that require effective side-lobe suppression.

For conventional windows, the number of samples N only affects the width of the central lobe; increasing N reduces the size of the lobe, but the maximum amplitude of the secondary lobes remains constant. While, the fractional window not only reduces the width of the central lobe, but also attenuates the secondary lobes as N increases. This property allows for the reduction of undesirable frequency artifacts, improving the overall quality of filtering or spectral analysis.

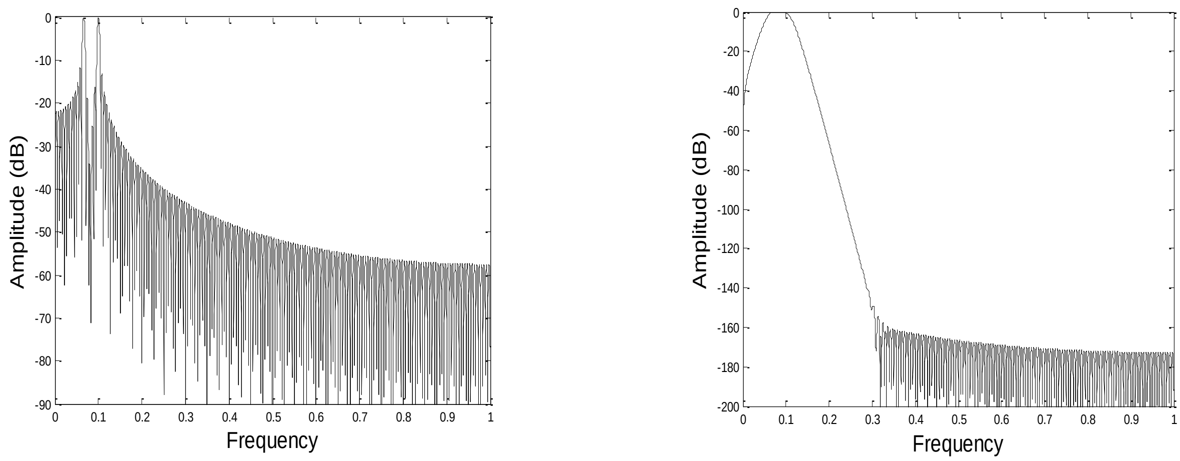
Higher α values (e.g., α = 0.1) lead to smaller secondary lobes, enhanced side lobe suppression, and reduced spectral leakage. However, these values result in a slightly wider central lobe compared to very high α values, which may reduce frequency resolution. Conversely, lower α values (e.g., α = 0.004) result in a narrow central lobe, which can offer better frequency resolution in some contexts. The downside is that these values produce much higher secondary lobes, leading to significant spectral leakage, making them less suitable for applications requiring clean spectral separation or handling signals with a high dynamic range.

III.3.3 Fractional windows properties in spectral analysis:

Adjustable window functions such as fractional window offer a flexible approach to managing spectral leakage and optimizing filter characteristics in signal processing because of its nature [32]. The fractional window allows for better control over the trade-off between frequency resolution and spectral leakage. The possibility of reducing the width of its main lobe makes it possible to set the resolution of the analysis. To illustrate how fractional window affects the ability to discriminate very close frequencies, the example of separating two very close frequencies involves two frequency components with the following parameters:

- close frequencies, f1=2000 Hz and f2=3000 Hz (The amplitudes of the two components are equal) with frequency difference ΔF= 1000 Hz.
- Sampling frequency: Fs= 60 kHz.
- Number of points (window length): N= 361.

Figure (3.9), presents the resolution obtained by varying the value of α.



(a): for α = 0.004. **(b):** for α = 0.0923.

Figure 3.9: Fractional window-weighted signal spectrum.

When α is small, the window function tends to have a narrower main lobe and higher side lobes. This configuration allows better discrimination between closely spaced frequencies, resulting in strong frequency resolution.

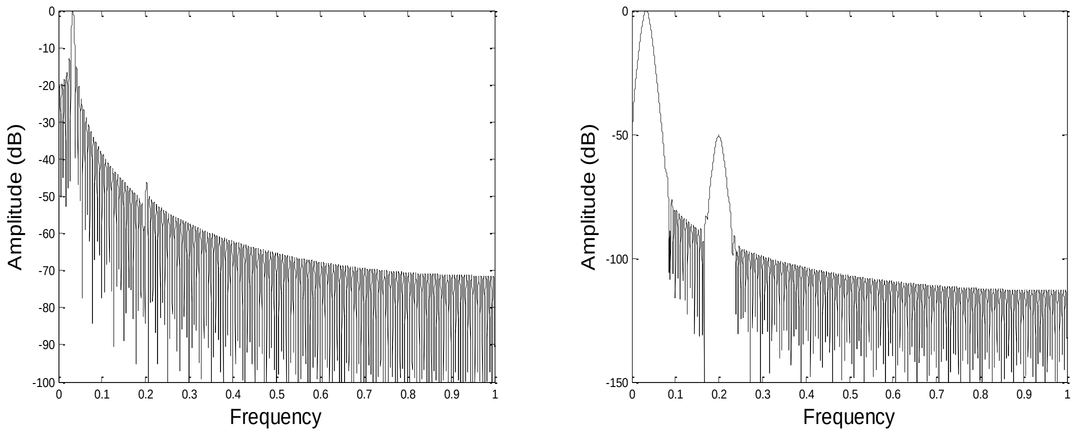
Increasing α, increases the width of main lobe and reducing side lobes attenuation. A wider main lobe does not enhance the ability to resolve closely spaced frequencies.

- **Resolving Distant Frequencies and amplitude discrimination:**

The possibility of attenuating the amplitudes of the secondary lobes makes it possible to fix the dynamics of the analysis. We can use the example of separating two distant frequencies, f1= 1000 Hz and f2= 6000 Hz, with very distinct amplitudes (ratio of 0.03) to show how fractional window influences the ability to precisely represent distant frequencies with very different amplitudes. This confirms the validity of the selected parameters. Sometimes adjusting these values can improve the results. Figure (3.10) presents the resolution obtained by varying the value of α.

With a small α values, the spectral plot likely shows limited peaks, it does not allow good dynamics to be obtained because the secondary lobes decrease very slowly. The weaker signal at 6000 Hz may appear less distinct due to the overlapping side-lobes from the stronger 1000 Hz, as we clearly observe in Figure (3.10) (a).

Conversely in Figure (3.10) (b), increasing α broadens the main lobe and also reduces the side lobes, minimizing spectral leakage, enhancing the ability to resolve and accurately represent the amplitudes of distant frequencies.



(a): for α = 0.004. **(b):** for α = 0.0923.

Figure 3.10: Dynamics Issue.

- **Summary:**

The analysis shows that to achieve accurate and reliable spectrum analysis in signal processing, the trade-off between resolution and leakage is managed by changing α in the fractional windows. The main lobe becomes narrower and the side lobes become larger as α decreases. This results in a more accurate and improved amplitude representation of frequencies, but also makes them less suitable for applications that require sharp spectral separation.

As α increases, Side lobes become suitable for signal processing and spectrum analysis applications that deal with high dynamic range signals (For components with a very large amplitude ratio, only the fractional window that provides good dynamic range).This highlights the importance of choosing the right α values to optimize frequency resolution and signal representation in spectral analysis

III.3.4 Comparison of Fractional Windows:

Rectangular, Hanning, Hamming and Blackman windows are types of fixed window functions. Main lobe width of these windows can be adjusted by varying the window length, while the secondary lobes remain constant. In the opposite, fractional window is adjustable because its spectral characteristics and shape can be adjusted according to the desire of the designer by varying two more independent parameters.

The features of the fractional window change based on α and N, making direct comparisons with other windows challenging. To address this, a new parameter β = α × N is introduced. This parameter defines the characteristics of the fractional window $W_{fract}(k)$. Each value of β provides a precise measure of the main lobe width or side lobe amplitude. By using the β parameter, the characteristics of the fractional window are normalized and fixed, enabling straightforward and fair comparisons with other windows. The β parameter allows the fractional window to replicate or approximate the intermediate characteristics of conventional windows, making it a versatile and flexible window for various applications. (For more details on the properties of β, refer to ANNEXE II). Table (3.2) shows comparisons between the fractional window and other windows.

Table 3.2: Comparison of fractional window and regular window in function of β.

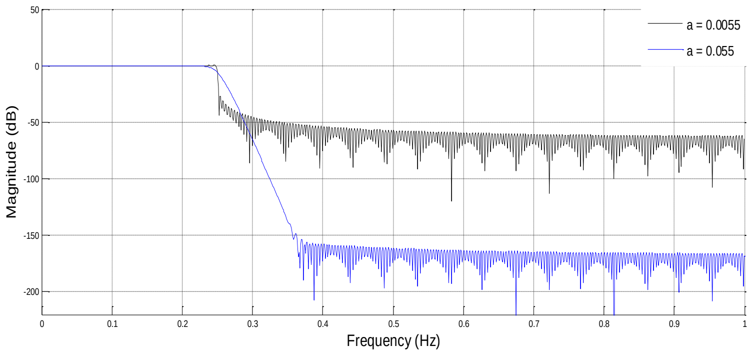
β	Main lobe width	Max amplitude of sidelobes	Equivalent window
1	2/N	-13 dB	Rectangular
6.9	3.9/N	-28.9 dB	Hanning
8.4	6.6/N	-41 dB	Hamming
11.5	10.8/N	-54 dB	Blackman

III.4. Effect of filter order and α parameters on Low-pass FIR filter design using fractional windows:

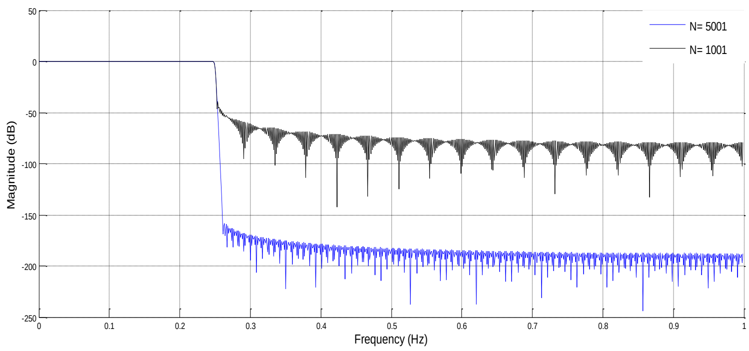
This analysis should help us to understand the trade-offs and behavior of FIR filters for different values of N and α. Figures (3.11) (a), (b) and (c) show the comparison between different fractional windows for a fixed parameter each time (window length or α parameter).

In these figures it should be noted that the highest side lobe amplitude, which gives the ripple ratio, for the windows occurs in the first side lobe, except that it does not occur in the first side lobe for α = 0.055 and N = 2001 in Figure (3.11) (c).

(a): number of coefficients equal to 501.



(b): parameter α = 0.0055.



(c): parameter α = 0.055.

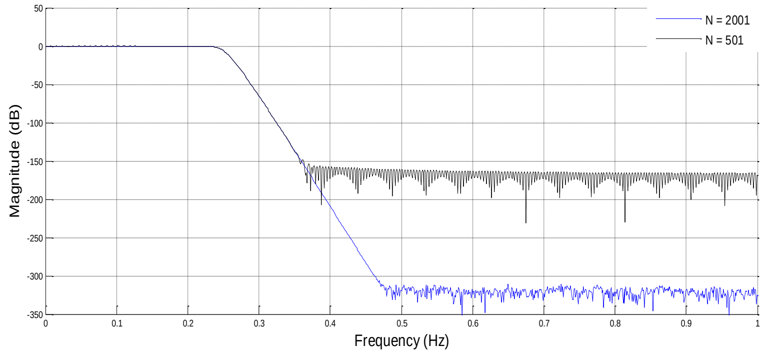


Figure 3.11: Fractional FIR representation for different values of N and α .

The results of the above figure are summarized in Table (3.3).

Table 3.3: Comparison of the variation in characteristics of the stop-band of fractional RIF filters.

Parametres α and N	**Main-lobe width (*2π)**	**Ripple Ratio**	**Side-lobe Roll-off Ratio**
α = 0.0055, N = 501	0.2529	-26.94 dB	37.20 dB
α = 0.055 , N = 501	0.3662	-154.40 dB	13.90 dB
α = 0.0055, N = 1001	0.2529	-39.13 dB	40.13 dB
α = 0.0055, N = 5001	0.2617	-158.70 dB	30.30 dB
α = 0.055 , N = 2001	0.4700	-308.50 dB	09.60 dB

The parameter α controls the trade-off between the width of the transition band and the ripple levels in the pass-band and stop-band. A smaller value of α results in a sharper transition band, but also increases the ripple in the pass-band and stop-band. Conversely, increasing α widens the transition band but reduces the ripples. While increasing the filter order N improves the filter performance by making the transition band sharper, it increases the computational complexity and delay introduced by the filter and reduces the stop-band ripples.

III.5 Evaluation of Low-pass FIR filter designed using fractional windows:

The desired specifications for a window function are a minor main lobe width and a reduced side lobe ripple ratio, but these requirements are contradictory since a narrower main lobe result in poorer side lobe rejection. However, the ability to modify the shape of the fractional window by adjusting the parameters N and α allow for tuning its spectrum. This flexibility in adjusting the α parameter enables fine-tuning of filter's performance, making it suitable for a wide range of applications. In this section we will design low pass FIR filters using fractional window and plot its frequency response. then, comparisons are made with those designed with Kaiser window to justify its performance in de-noising speech signal using low-pass FIR filtering.

Why Kaiser? Because it shares with the fractional window in being adjustable, and that the Kaiser window is preferred in speech signal filtering due to its adjustable shape parameter that provides control over the main-lobe width and side-lobe attenuation [33].

This example compares the fractional window with the commonly used adjustable Kaiser window. By varying adjustable parameters, main lobe width and ripple ratio can be adjusted. The comparison of the low pass FIR filter design using Kaiser for β = 2 and a fractional window for $\alpha = 0.012$ is shown in Figure (3.12). Consider that the length of the filter is N = 361 and the normalized cut-off frequency is fc = 0.125.

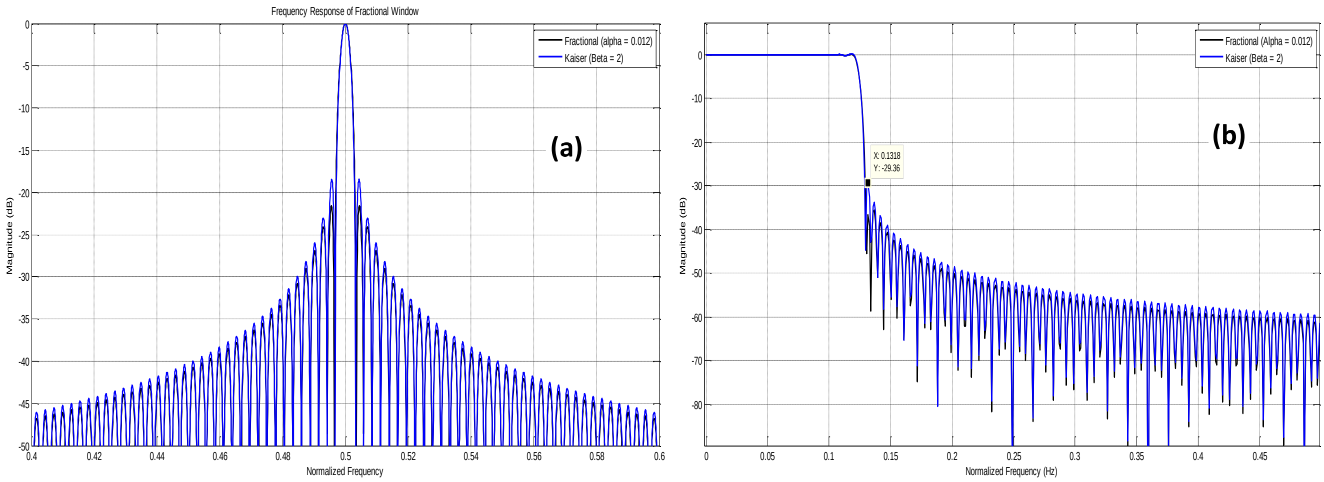


Figure 3.12: (a) Fractional and Kaiser spectrum, (b) Low-pass FIR magnitude by two windows.

In this example, we have chosen a fractional window with characteristics similar to those of the Kaiser window. This choice showed that FIR filter designed by fractional window represents an attenuated stop-band compared to that obtained by Kaiser (-35 dB and -29 dB). Both windows gave almost the same width of the transition-band.

- **Results achieved:**

The simulation example show that the behaviors of low-pass FIR filters designed using fractional and Kaiser windows are very similar and coincide well for the design of filters with good characteristics. The major difference is that Kaiser window might be more computationally intensive due to the need for calculating more complex functions or higher order terms (The Kaiser window requires calculating the modified Bessel function, which is computationally intensive, especially for high order) [33]. This is what is unique about the fractional window, because it can follow the properties of regular and even adjustable windows without requiring great complexity or computing power.

III.6 Evaluation a speech signal de-noising using fractional windows-based FIR filter:

Speech signal filtering is a fundamental process in Digital Signal Processing (DSP) that is applied in various fields such as telecommunications, audio engineering and speech recognition systems. Speech signal filtering aims to improve the quality, intelligibility and usability of speech by removing unwanted components, such as noise, and enhancing desired features. This process involves several techniques and methodologies tailored to the specific requirements of speech applications, including digital filters [14].

When designing FIR filters, it's crucial to balance transition band sharpness and ripple magnitude while considering available computational resources. Incorrect parameter selection can result in suboptimal performance, with insufficient side lobe suppression or an unnecessarily broad main lobe. These factors must be carefully evaluated, particularly in applications with stringent requirements.

In example below, we are using 5 seconds distorted audio stream is analyzed in the frequency and temporal domains using MATLAB, and assessment the process of signal restoring using FIR filters made by Fractional and Kaiser windows.

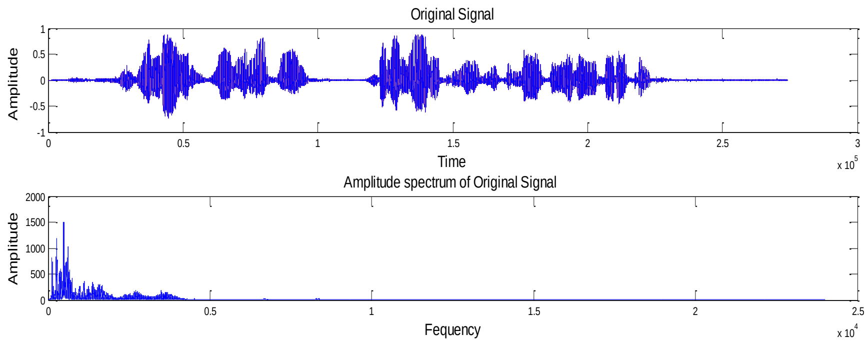


Figure 3.13: The original speech signal’s representation.

It can be seen in Figure (3.13) that the speech signal amplitude is not more than 0.9. Frequency domain diagram of the simulation results, which can be seen that the frequency of the speech signal is mainly concentrated in the low frequency part, the maximum amplitude of the speech signal frequency does not exceed 1500.

The filter design equations for the two windows to meet the given example specification are established and the results obtained is discussed. The filter’s performance was evaluated by calculating the Signal-to-Noise Ratio (SNR) of the original, noisy, and filtered signals. For the filters evaluation, in the beginning we choose the speech signal pre-defined and represented in Figure (3.13) and adding a sum of high-frequency notes as noise to simulate a noisy speech signal. The characteristic of the distorted speech signal is shown in Figure (3.14) for temporal representation and Figure (3.15) for frequency representation.

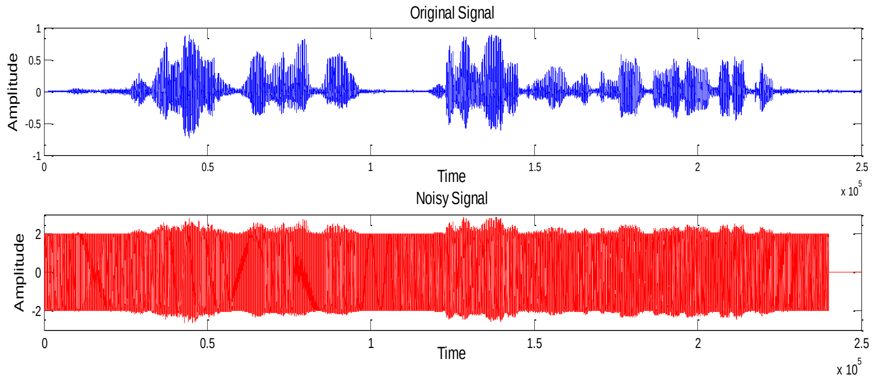


Figure 3.14: Noisy and original speech signal.

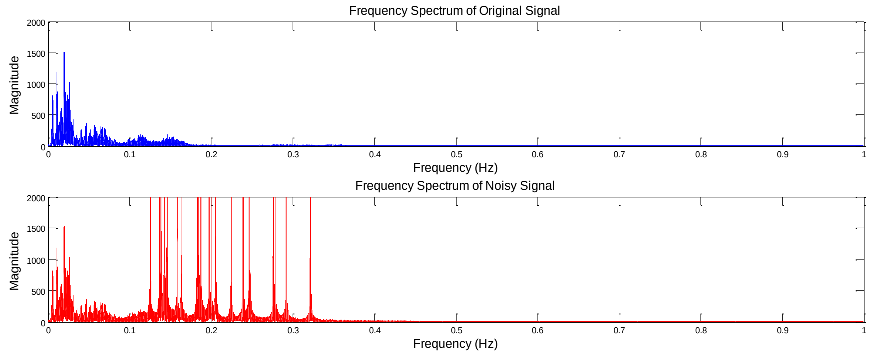


Figure 3.15: Spectrum of noisy and original speech signal.

By observing the simulation results in Figure (3.14) and (3.15), the effect is very obvious when adding noise into the signal. Which can be seen in the time domain graph that the original speech signal almost could not be revealed, deeply buried in the noise.

- Use FFT to obtain the signal spectrum, then analyzed the characteristics of the time domain and frequency domain.

- Finally, play back the speech signal.

- Complete a system to analyze and process the voice signal using MATLAB.

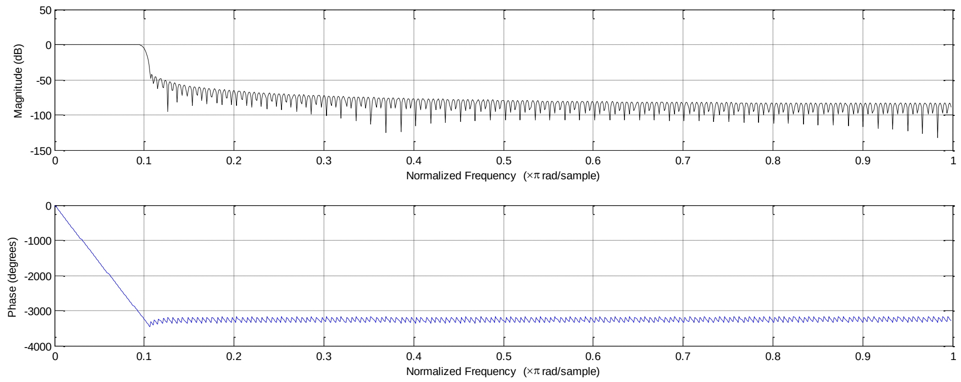
- Evaluating the performance of filters by calculating SNR.

- Do spectrum analysis of the original voice signal and the noisy voice signal well?

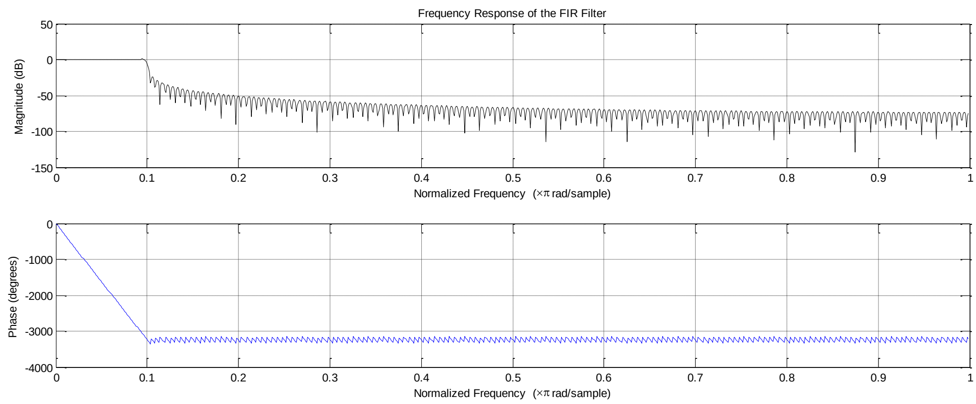
Designing a low-pass FIR filter using a window involves several steps, which include specifying the filter requirements, creating the window, and designing the FIR filter using this chosen window. As a first step, we design a two low-pass FIR filters using a fractional and kaiser window and plot its frequency response. The fractional window's FIR filters are compared to those other kaiser windows after putting them under the identical circumstances.

In this example, an FIR low-pass filter was designed and tested using fractional and Kaiser windows. The filter's performances were evaluated by both calculating SNR and audio comparison to the original, noisy, and filtered signals.

The performance requirements which can be determined by connecting the actual situation are: Pass-band boundary is 2700 Hz. Stop-band boundary is 3000 Hz, pass-band ripple is less than 30 dB. Since the stop-band attenuation is less than 120 dB. Kaiser window for β = 3.60; Fractional window for α = 0.005, is chosen to design FIR low-pass filters. Note that in figures below the normalized cutoff frequency is F_{c_norm} = 0.10 with N = 361, and 5001 respectively.

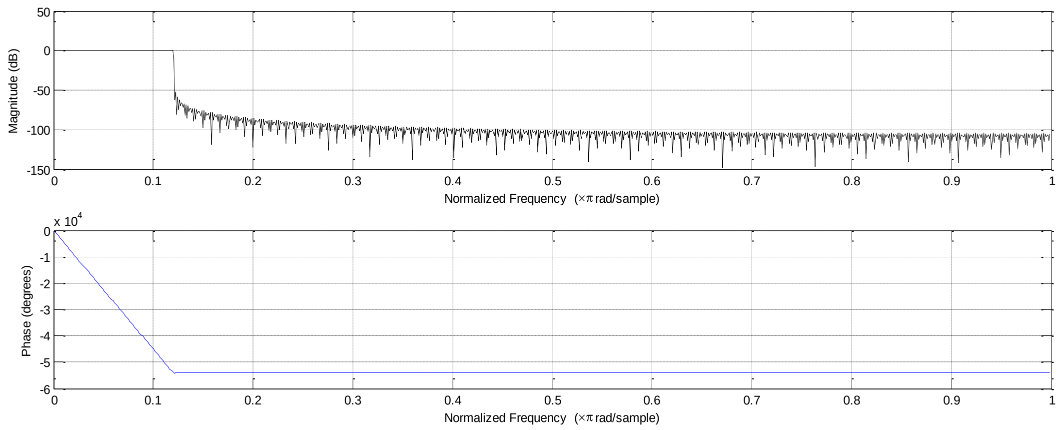


(a) Kaiser window.

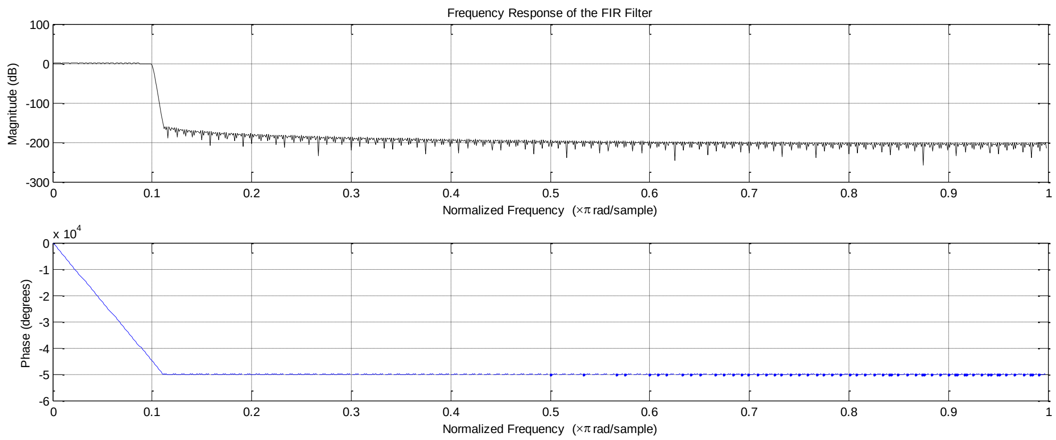


(b) Fractional window.

Figure 3.16: Magnitude frequency response and phase response for N = 361.



(a) Kaiser window.



(b) Fractional window.

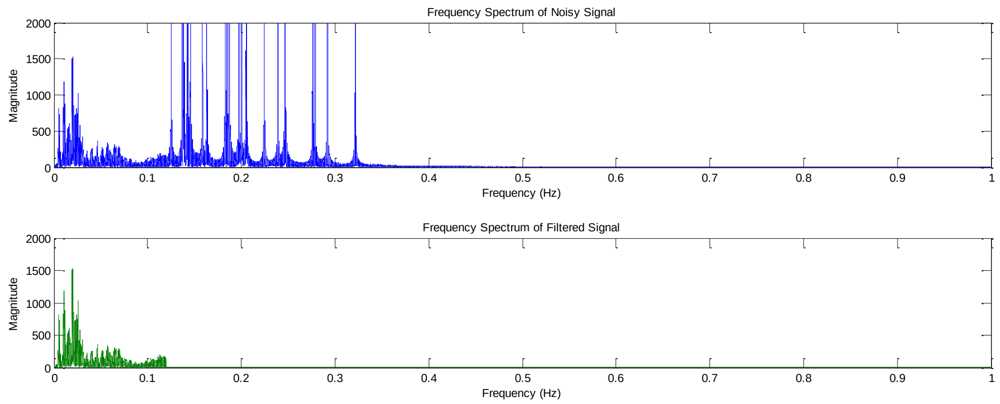
Figure 3.17: Magnitude frequency response and phase response for N = 5001.

The following table shows the variation in SNR of the FIR filters made by fractional and Kaiser windows.

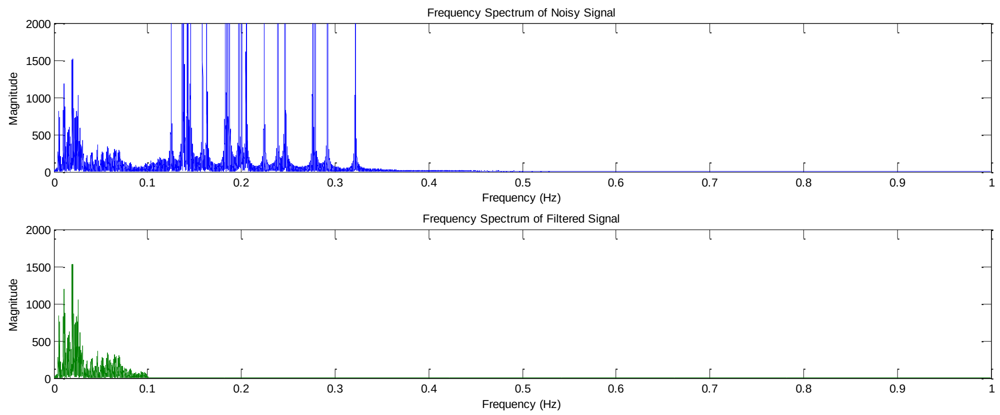
Table 3.4: Comparison of the variation in characteristics of the kaiser RIF filters

Windows	N	α	β	Sidelobe Roll-off ratio	SNR
Kaiser	361	0.005	3.6	- 47.03 dB	3.7071 dB
Fractional				- 35.5662 dB	2.8847 dB
Kaiser	5001			- 52.37 dB	2.10055 dB
Fractional				- 183.38 dB	3.6545 dB

By increasing β in Kaiser window, you can achieve better side lobe attenuation, thus improving the ripple ratio and reducing spectral leakage. However, this comes at the cost of a wider transition band. Selecting the optimal β value can be non-possible and may require experimentation or domain-specific knowledge. Which reduces speech resolution [33]. The filter can be improved by changing its parameters such as filter rank and cutoff frequency to further improve the SNR.

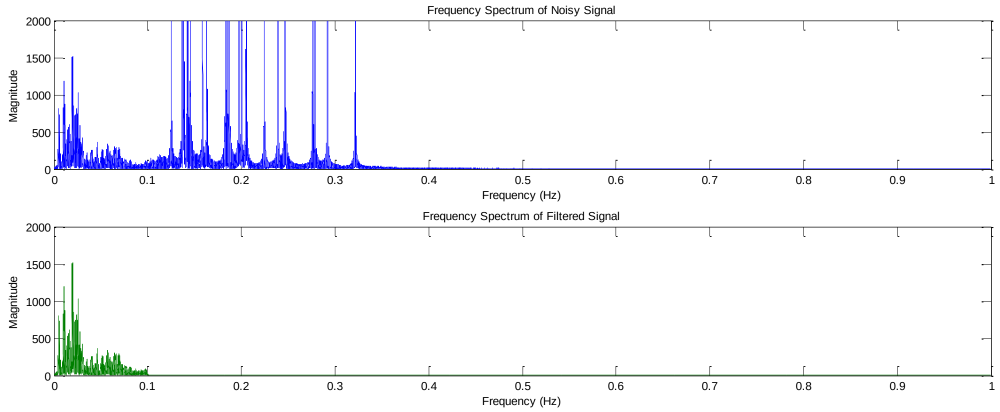


(a) Result obtained using Kaiser FIR filter.

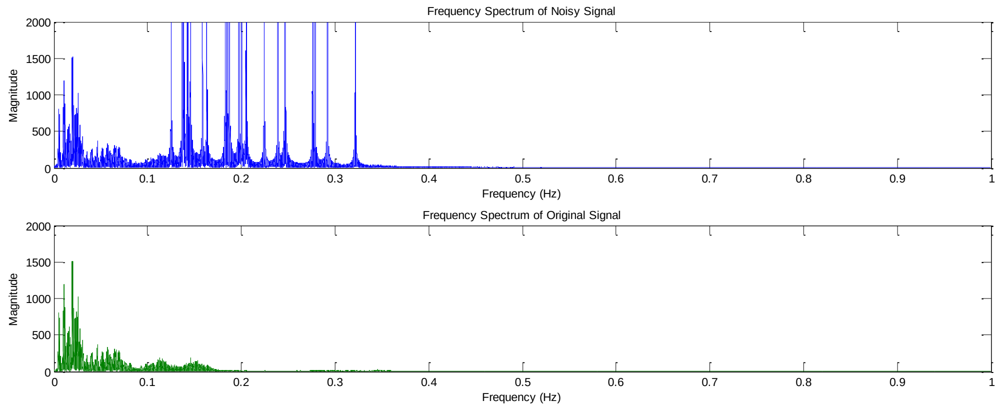


(b) Result obtained using Fractional FIR filter.

Figure 3.18: Respectively the noisy and filtered speech signal for N = 361.



(a) Result obtained using Kaiser FIR filter.



(b) Result obtained using Fractional FIR filter.

Figure 3.19: Respectively the noisy and filtered speech signal for N = 5001.

As shown in Figure (3.19) (b). In terms of frequency domain, the filtered signal's waveform is nearly identical to that of the original speech signal. By using the audio comparison, it is possible to determine that the speech information content is integrity and the original speech signal was very clear.

Upon evaluating the results obtained and represented in the figures (3.18), and (3.19), you can see that the fractional window, with an appropriately chosen α parameter, can achieve a better ripple ratio, providing lower side lobes and thus reducing spectral leakage more effectively than Kaiser window. This is even for large values of order N where longer filters increase the computational burden and delay for Kaiser window, which can be problematic in real-time applications. In contrast to fractional window, which demonstrated excellent and robust performance under such exceptional circumstances, you will be able always to tailor the window to your specific needs, optimizing the ripple ratio for your specific application.

The study concludes that the fractional window is showing increasingly better result among all the existing windows mentioned above. For the low-pass filter, the presented window-based filter still shows better result. While comparing with Kaiser filter, the presented filter in Figure (3.17) is showing better ripple ratio and side-lobe roll-off ratio. For Kaiser filter, the ripple ratio is still higher and side-lobe roll-off ratio is lower than the presented filter which implies affirmative result for the proposed filter. Moreover, in the case of voice data de-noising, the presented filter is showing better performance than others.

III.7 Conclusion

An evaluation of a new weighting window, called the fractional window, is presented. A study is made of this window and the characteristics obtained are compared with those of existing windows, with the aim of using the latter in various applications and particularly in the digital FIR filtering of speech signals. From the experiment it has been obtained that the presented window-based filter has a better filtering result than other mentioned filters for filtering audio speech data.

Regarding usual windows, the characteristics are fixed; each window has well-specified characteristics, therefore a limited application. The width of the main lobe is limited, it cannot be less than 2/N for all windows. For the fractional window, the shape can be varied by manipulating the parameters N and α. The possibility of changing the shape of the fractional window makes it possible to modify its spectrum, which varies these characteristics, therefore a wide field of application.

The implementation of FIR filters using innovative window functions represents a significant advancement in speech signal processing, bringing substantial improvements in audio quality and clarity. These advanced window functions effectively mitigate issues such as spectral leakage, leading to more precise frequency response and enhanced speech intelligibility. By providing superior noise reduction and customizable filter designs tailored to specific speech signal characteristics, these innovative approaches ensure more intelligible audio. Furthermore, the optimized performance and computational efficiency of FIR filters with advanced windowing make them suitable for real-time applications across various domains, including telecommunications, voice-controlled systems, and hearing aids. As digital signal processing and deep learning techniques continue to evolve, the development of adaptive and hybrid filtering methods will further enhance the capabilities and efficiency of speech signal processing solutions, paving the way for more versatile and high-performance audio technologies in the future.

We hope that this introductory text about digital filters made by fractional window will be useful in further investigation and applications of digital filters for speech signal processing.

Conclusion

The motivation behind the choice of this window is its ability to change the shape through a fractional coefficient, which makes it able to form into any known window, as well as the possibility of reducing the attenuation of stop-band low-pass FIR filter with increasing samples number N.

In order to evaluate this window for digital speech signal filtering, we have tried to announce some basic notions about digital signal processing, including digital systems, sampling, and Fourier analysis to fully understand the mechanism of FIR digital filters.

An overview of the synthesis of FIR filters using the window method shows that this type of filters is characterized by two essential parameters, the transition band width, which depends on the width of the main lobe of the window and the stop-band attenuation, which depends on the secondary lobes amplitude of the window. Concerning the usual or fixed windows, only the main lobe width which varies according to the window main lobe width, except the maximum amplitude of the secondary lobes is constant for each window. To this end, a new generation of adjustable windows has appeared to give flexibility in filter design or spectral analysis. The fractional window is one of these windows, which is characterized by two parameters adjusting its shape, which is translated in the spectral domain by a great flexibility on the compromise between the width of the main lobe and the attenuation of the secondary lobes.

These conditions have led us to make better use of this window in speech signal filtering. The design of the FIR filters using this window showed the ability to separate a strongly noisy speech signal and gave encouraging results compared to those obtained with the Kaiser window. On the contrary, it surpassed it in terms of simplicity and time execution because of its formula based on hyperbolic cosine functions, but the Kaiser window uses a more complicated formula that requires more time to execute.

As a perspective, we propose to deepen the study by introducing additional parameters into the fractional window representation formula. This could enhance its versatility and efficacy across a broader range of applications beyond speech signal processing. For instance, in biomedical signal processing, the fractional window can improve the detection of QRS complexes which are crucial for diagnosing various cardiac conditions. Additionally, in image filtering, the fractional window can be beneficial for designing filters that enhance edge detection in images, which is vital for computer vision applications.

REFERENCES :

[1] Fifty Years of Signal Processing: The IEEE Signal Processing Society and its Technologies, 1948 – 1998, Link (Last access: 06/06/2024): https://signalprocessingsociety.org/our-story/society-history

[2] J. Eyre and J. Bier, “The Evolution of DSP Processors”, IEEE Signal Processing Magazine, Vol. 17. No.2, (2000).

[3] R.W. Hamming, “Digital Filters”, Dover Publications. INC, New York, 1989,3rd ed, ISBN-13:978-0-486-65088-3.

[4] A. Gupta, J. Afridi, S. Shriwastava and V.V. Thakre, “A Two Variable Adjustable Window Function Approach to Design FIR Filter”, M. Pandit et al. (Eds.): ICSISCET 2019, PALO 13, pp. 123–131, (2020).

[5] K. Avci and A. Nacaroğlu, “Cosine Hyperbolic Window Family with its Application to FIR Filter Design”, 3rd International Conference on Information and Communication Technologies: From Theory to Applications, IEEE, (2008).

[6] M. Das, R. Kumar and B.C. Sahana, “Implementation of Effective Hybrid Window Function for E.C.G Signal Denoising”, Traitement du Signal. Vol. 37, No. 1, pp. 119-128, (2020).

[7] T. Karmaker, Md.S. Anower, M.A.l.G. Khan, Md.A. Habib, “A New Adjustable Window Function to Design FIR Filter and Its Application in Noise Reduction From Contaminated ECG Signal”, IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 21 - 23 Dec 2017, Dhaka, Bangladesh, (2017).

[8] V. Kumar, S. Bangar, S.N. Kumar and S. Jit, “Design of Effective Window Function for FIR Filters”, IEEE International Conference on Advances in Engineering &Technology Research (ICAETR - 2014), Unnao, India, (2014).

[9] H. Rakshit and Muhammad Ahsan Ullah, “An Adjustable Novel Window Function with its Application to FIR Filter Design”, 1st International Conference on Computer & Information Engineering, Rajshahi, Bangladesh, (2015).

[10] Md.A Hossin, M. Shil, V. Thanh, N. Tung Son, “An Adjustable Window-Based FIR Filter and Its Application in Audio Signal De- Noising”, IEEE, 3rd International Conference on Robotics and Automation Engineering, (2018).

[11] Md.N. Islam Khan, S.R. Sabuj, M. Shil, “An Adjustable FIR Filter to Denoise the Noisy Voice Signal in Air Traffic Communication”, IEEE International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), (2019).

[12] H. Nezzari, “Introduction d’une Fenêtre du Pondération Fractionnaire et son utilisation en Filtrage Numérique RIF”, Master’s thesis, University Of Constantine, Constantine, 2005.

[13] W. Wang, “Introduction to Digital Signal and System Analysis”, University of Sussex, Brighton, England, (2012), ISBN 978-87-403-0158-8.

[14] L. TAN, “Digital Signal Processing Fundamentals and Applications”. Academic Press of Elsevier, DeVry University, Decatur, Georgia, (2008), ISBN 978-0-12-374090-8.

[15] John G. Proakis and Dimitris G. Manolakis, “Digital Signal Processing, Principles, Algorithms, and Applications”, Prentice-Hall, Inc, Upper Saddle River, N ew Jersey, (1996), ISBN 0-13-3TM33fl-cl

[16] Steven W. Smith, “The Scientist and Engineer's Guide to Digital Signal Processing”, 2end Edition, California Technical Publishing, San Diego, (1999), ISBN 0-9660176-6-8.

[17] M. Achouri, “Etude Complète d’un Filtre Numérique défini par sa Fonction de Transfer H(z) ”, Master mimory, University Cheikh Larbi Tebessi, Tebessa, 2003.

[18] N. Wu and B. Wang, “Process and Analysis of Voice Signal by MATLAB”, Bachelor thesis in Electronics, Faculty of Engineering and Sustainable Development, University of GAVLE, Sweden, June 2014.

[19] M. Bellanger, “Traitement numérique du signal, Théorie et pratique”, 8e édition, Dunod, Paris, (2006), ISBN 2 10 050162 3.

[20] D. F. Elliott and K. R. Rao, “Fast Transforms-Algorithms, Analyses and Applications”, Academic Press, New-York, (1982), ISBN 0-12-237080-5.

[21] Julius O. Smith III, “Mathematics of the Discrete Fourier Transform (DFT)”, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, California, 2004, Link: (Last access: 02/06/2024)
http://tomlr.free.fr/Math%E9matiques/Fichiers%20Claude/Auteurs/aaaDivers/Smith%20J%20-%20Mathematics%20Of%20The%20Discrete%20Fourier%20Transform.pdf

[22] E. Chu and A. George, “Inside the FFT black box”. CRC Press LLC, New York, (2000), ISBN 0-8493-0270-6

[23] L. Wanhammar, T. Saramäki, “Digital Filters Using MATLAB”, Springer Nature Switzerland AG, (2020), ISBN 978-3-030-24063-9.

[24] L Jiufel and X Zhijiang and L Xinyi, "Asymmetric Windows and Their Application in Frequency Estimation", Chongqing University, 2015 doi:10.1260/1748-3018.9.4.389

[25] K. Prabhu, “Window Functions and Their Applications in Signal Processing”, Boca Raton, CRC Press, 2014 ISBN 978-1-4665-1583-3.

[26] Nezih C. Geckinli, “Some Novel Windows and a Concise Tutorial Comparison of Window Families”, IEEE Transaction on Acoustics, Speech and Signal Processing, vol ASSP-26, NO 6, december 1978.

[27] M. D. Lutovac, D.V. Tošić and B.L. Evans. “Filter Design for Signal Processing Using MATLAB” Beijing: House of Electronics Industry, 2004.

[28] Ohlmann K. C, “Principles of Digital Audio”, McGraw-Hill, 2010.

[29] P. Guillaume, Conference Paper “Design of Optimal FIR Filter with Arbitrary Magnitude and Phase Specifications”, The IEEE International conference on Acoustics, Speech, and Signal Processing (ICASSP), 2021.

[30] E. Punskaya, “Design of FIR Filters”, Some material adapted from courses, University of Cambridge, Link (Last acces: 03/06/2024):
https://www.robots.ox.ac.uk/~gari/teaching/cdt/A3/readings/Filtering_and_FFT/3F3_5_Design_of_FIR_Filters.pdf

[31] D. G. Manolakis, and V.K. Ingle, and S. M. Kogon, “Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing”, Artech House Publishers, 2005 ISBN 978-1580536103.

[32] H. Perez-Meana, “Advances in Audio and Speech Signal Processing: Technologies and Applications”, Natonal Polytechnc Insttute, Mexco, 2007 ISBN 978-1-59904-134-6 (ebook).

[33] Y. Wei, and X. Fan, Conference Paper " De-noising the Speech Signal with FIR Filter Based on Matlab", 4th International Conference on Electrical & Electronics Engineering and Computer Science (ICEEECS), 2016.

ANNEXE I

α	N	25	60	70	100	150	300	500	700	1000	1500	2000	2500	3000	3500
0.001	L_B	2.00	2.00	2.01	2.02	1.97	2.05	1.95	2.05	1.95	2.19	1.95	3.66	5.86	11.96
	L_H	0.60	0.60	0.60	0.59	0.60	0.62	0.54	0.59	0.61	0.54	0.73	0.91	0.37	0.43
	A	13...21	13.25	13.25	13.26	13.26	13.34	13.39	13.64	13.65	14.10	14.75	20.14	25.06	32.67
0.005	L_B	2.00	1.99	2.01	2.00	2.05	2.05	2.19	2.39	2.92	6.59	7.81	12.20	21.97	23.93
	L_H	0.60	0.60	0.60	0.59	0.60	0.62	0.67	0.59	0.85	0.91	1.22	1.52	1.83	2.14
	A	13.22	13.28	13.30	13.35	13.46	14.09	15.66	18.22	22.25	35.23	38.66	55.14	74.04	81.57
0.01	L_B	2.00	2.02	2.01	2.00	2.05	2.19	2.68	4.10	7.81	16.11	27.34	42.72	55.66	75.20
	L_H	0.60	0.60	0.60	0.59	0.60	0.62	0.79	0.93	1.09	1.64	2.19	2.74	3.30	3.85
	A	13.23	13.38	13.43	13.62	14.09	16.81	22.24	28.96	43.61	64.38	85.81	108.69	123.99	146.39
0.012	L_B	2.00	2.02	2.01	2.05	2.05	2.34	3.41	6.49	11.71	23.43	39.06	59.81	79.10	107.67
	L_H	0.60	0.60	0.60	0.62	0.60	0.69	0.79	0.93	1.34	2.01	2.68	3.35	4.03	4.70
	A	13.24	13.44	13.51	13.78	14.46	18.56	28.13	37.87	54.85	81.55	105.80	135.32	151.22	181.80
0.015	L_B	2.00	2.02	2.01	2.05	2.12	2.63	6.34	8.20	15.62	35.15	55.66	87.89	123.05	167.48
	L_H	0.60	0.60	0.60	0.62	0.64	0.69	0.91	1.11	1.58	2.38	3.17	3.96	4.76	5.55
	A	13.26	13.54	13.65	14.09	15.17	22.45	37.64	42.88	64.22	102.96	126.43	165.96	195.32	232.98
0.017	L_B	2.00	2.02	2.05	2.05	2.19	2.78	6.59	11.62	19.53	43.21	71.28	107.42	155.27	203.37
	L_H	0.60	0.60	0.60	0.62	0.64	0.76	1.03	1.28	1.83	2.74	3.66	4.57	5.49	6.41
	A	13.28	13.63	13.77	14.33	15.75	26.12	41.31	54.65	72.29	114.48	145.55	180.35	221.89	250.70
0.02	L_B	2.00	2.05	2.05	2.09	2.27	3.36	7.81	15.03	27.34	55.66	95.70	147.70	210.94	271.73
	L_H	0.60	0.60	0.60	0.62	0.64	0.76	1.15	1.62	2.31	3.47	4.63	5.79	6.96	8.12
	A	13.30	13.78	13.97	14.75	16.78	28.13	43.41	64.43	88.79	127.47	170.57	216.41	250.94	250.75
0.04	L_B	2.02	2.13	2.22	2.44	3.36	11.42	27.58	51.26	95.70	208.00	301.75	384.52	458.50	548.58
	L_H	0.60	0.63	0.64	0.69	0.78	1.35	2.25	3.16	4.51	6.77	9.03	11.29	13.55	15.81
	A	13.58	15.44	16.29	20.08	28.11	54.67	88.58	124.61	170.86	256.75	256.77	258.64	250.69	250.68
0.05	L_B	2.03	2.22	2.32	2.78	6.37	15.82	39.79	75.53	147.46	275.39	433.59	544.43	0.05	781.01
	L_H	0.61	0.65	0.67	0.74	0.89	1.64	2.74	3.84	5.49	8.23	10.98	13.73	16.48	19.23
	A	13.78	16.79	18.24	25.81	37.53	64.15	106.24	151.40	215.23	256.82	256.76	257.87	250.67	250.66
0.09	L_B	2.13	2.95	3.72	7.03	14.79	47.46	123.53	235.49	357.42	509.76	678.71	847.16	1016.60	1184.33
	L_H	0.63	0.76	0.82	1.03	1.48	2.96	4.94	6.92	9.88	14.83	19.77	24.71	29.66	34.61
	A	15.15	26.86	28.51	42.33	63.37	117.26	193.81	263.65	262.79	256.69	256.71	256.85	250.65	250.65
0.1	L_B	2.16	3.39	4.06	7.81	15.82	59.32	151.61	267.62	436.03	626.95	745.11	930.17	1321.29	1546.63
	L_H	0.63	0.79	0.88	1.13	1.66	3.33	5.55	7.77	11.10	16.66	22.21	27.77	33.33	38.88
	A	15.64	28.089	28.65	43.03	63.42	132.57	214.42	263.59	262.79	256.68	256.70	256.79	250.65	250.65

Variation of fractional window characteristics as a function of α and N

ANNEXE II

β	L_B	L_H	A	β	L_B	L_H	A	β	L_B	L_H	A
1	1.99	0.61	13.62	**6.4**	3.76	0.83	28.68	**11.8**	11.21	1.31	54.31
1.2	2.06	0.61	13.79	**6.6**	3.83	0.83	28.83	**12**	11.50	1.35	54.50
1.4	2.06	0.61	13.98	**6.8**	3.91	0.87	28.89	**12.2**	11.58	1.35	54.59
1.6	2.06	0.61	14.22	**7**	4.06	0.87	28.82	**12.4**	11.65	1.38	54.56
1.8	2.06	0.61	14.47	**7.2**	4.20	0.87	28.62	**12.6**	11.72	1.38	54.41
2	2.14	0.61	14.76	**7.4**	4.42	0.90	28.20	**12.8**	11.87	1.42	54.14
2.2	2.14	0.65	15.08	**7.6**	6.41	0.90	38.03	**13**	12.02	1.46	53.72
2.4	2.14	0.65	15.45	**7.8**	6.41	0.94	39.11	**13.2**	12.24	1.46	53.12
2.6	2.21	0.65	15.86	**8**	6.49	0.94	39.87	**13.4**	14.67	1.49	63.18
2.8	2.21	0.65	16.30	**8.2**	6.56	0.98	40.45	**13.6**	14.82	1.49	63.56
3	2.21	0.65	16.78	**8.4**	6.64	0.98	40.99	**13.8**	15.04	1.53	63.85
3.2	2.29	0.65	17.33	**8.6**	6.71	1.01	41.51	**14**	15.41	1.57	64.05
3.4	2.29	0.68	17.92	**8.8**	6.86	1.01	41.98	**14.2**	15.48	1.57	64.16
3.6	2.36	0.68	18.56	**9**	7.00	1.05	42.39	**14.4**	15.56	1.60	64.16
3.8	2.43	0.68	19.29	**9.2**	7.15	1.05	42.73	**14.6**	15.63	1.60	64.04
4	2.43	0.68	20.07	**9.4**	7.52	1.09	43.01	**14.8**	15.70	1.64	63.80
4.2	2.51	0.72	20.95	**9.6**	7.59	1.09	43.20	**15**	15.85	1.68	63.43
4.4	2.58	0.72	21.93	**9.8**	7.67	1.12	43.28	**15.2**	16.00	1.68	62.92
4.6	2.58	0.72	23.04	**10**	7.74	1.12	43.26	**15.4**	18.73	1.71	71.98
4.8	2.65	0.72	24.29	**10.2**	7.89	1.16	43.10	**15.6**	18.95	1.71	72.29
5	2.73	0.76	25.76	**10.4**	8.04	1.16	42.81	**15.8**	19.39	1.75	72.49
5.2	2.88	0.76	26.36	**10.6**	8.26	1.20	42.34	**16**	19.39	1.79	72.59
5.4	2.95	0.76	26.85	**10.8**	10.54	1.20	52.24	**16.2**	19.46	1.79	72.59
5.6	3.02	0.79	27.31	**11**	10.62	1.23	52.77	**16.4**	19.54	1.82	72.47
5.8	3.17	0.79	27.74	**11.2**	10.69	1.23	53.26	**16.6**	19.61	1.86	72.24
6	3.32	0.79	28.11	**11.4**	10.84	1.27	53.69	**16.8**	19.69	1.86	71.87
6.2	3.69	0.83	28.43	**11.6**	10.99	1.31	54.04	**17**	19.83	1.90	71.38

Variation of fractional window characteristics as a function of β