Foundation University Journal of Engineering and Applied Sciences

FUJEAS
Vol. 5, Issue 1, 2024
DOI:10.33897/fujeas.v5i1.843

Research Article

Article Citation:
Soomro et al. (2024). "Exploring the Art of Sampling and Reconstruction - A PBL Approach to Signal Processing". *Foundation University Journal of Engineering and Applied Sciences* DOI:10.33897/fujeas.v5i1.843

Published by Foundation University Islamabad.
Web: https://fui.edu.pk/

Exploring the Art of Sampling and Reconstruction - A PBL Approach to Signal Processing

Abdul Hameed Soomro [*], **Zahoor Ahmed Mahar, Rahman Ali, WaliUllah Kalwar, Hira Asghar, Mutahira Soomro**

Department of Electrical Engineering, The University of Larkano, Sindh, Pakistan.

[*] **Corresponding author**: abdul.hameedr@uolrk.edu.pk

Abstract:

Digital signal processing has facilitated the digital representation, analysis, and transmission of analog signals. This work presents a Project-Based Learning (PBL) approach to encourage students to work on real-world projects or challenges to gain knowledge and skills in the field of signal sampling and reconstruction, focusing on their significance in multidimensional domains where they are applied, such as communication systems, image processing, or audio signal processing. Sampling is how a continuous-time signal is transformed into a discrete format, i.e., when we select values at different time points. This requires taking samples of signal amplitudes at uniformly spaced intervals, which creates a stream of quantities. But the main difference is that to extract information from an analog signal, we need samples; there is no other way. This process is referred to as sampling rate frequency, so it is the number of samples collected during some period of time. Reconstruction, on the other hand, means doing a conversion of it from time-discrete to its continuous-time form. This operation generated an approximated signal that is continuous from those sampled values. Different types of reconstruction methods, such as ideal interpolation, zero-order hold, or since interpolation, are chosen based on signal features and need. Given that the reconstruction process is limited because it is based on only a finite window of samples, it becomes clear how important accurate sampling and reconstruction are in preserving the original quality of a signal, minimizing distortion during this part of the audio chain. The Nyquist–Shannon sampling theorem, also called as Nyquist criterion or sometimes as Shannon sampling theorem, defines a good minimum rate at which a band-limited signal to be sampled so that it can be reconstructed without the loss of information. It is important to note that it would have a big effect on the systems that can be developed, and that are both efficient and dependable. In brief, learning sampling and reconstruction is arguably the most basic of signal processing concepts. Appropriate sampling allows to keep the integrity and quality of a signal across a wide range of applications, i.e., from new communication technologies with diminished bandwidth, multimedia systems to many other subjects that aim for innovative high-performing digital systems.

Keywords: Project-Based Learning (PBL), Sampling and Reconstruction of Signal, Digital Signal Processing.

1. Introduction

Sampling and reconstruction are fundamental operations in signal processing that underlie many of the techniques used throughout science and engineering. These operations have been studied extensively over the years, yielding both theoretical understanding and practical applications. One of the earliest and most important contributions is the

Nyquist–Shannon sampling theorem, which stems from Harry Nyquist's work up to 1928, and Claude Shannon's publications in 1949. Based on this, the first principle (Shannon theorem) states that in order to accurately sample and reconstruct a signal, the sample rate must be higher than twice the maximum frequency available [2]. It has since become a fundamental concept in digital signal processing, and much research has been conducted as a result [3]. Research in this field has gone far beyond simplistic uniform sampling [2], [4]. Random and adaptive sampling has been proposed by researchers, where the samples are selected intelligently depending on whether the signal is smooth or has fluctuations in accordance with the needs of the application. At the same time, compressed sensing has recently been introduced as an approximate solution to signal acquisition that permits high fidelity, low-precision reconstruction of a sparse signal from samples rather than the traditional approach, for significant savings in both acquisition and storage [5]. Signal reconstruction has also seen developments. While basic methods like linear or polynomial interpolation are still valuable, advanced methods such as wavelet-based reconstruction or compressed sensing algorithms are better for complex or noisy data [6]. Applications are numerous: in image processing, methods like pixel interpolation, super-resolution, and in painting enhance details or fill in missing data. In audio, sampling and reconstruction are key to compression, noise reduction, and synthesis. As with communication systems, these principles also apply to the recording and reproduction of sound in general, including the use of analog-to-digital (ADC) and digital-to-analog (DAC) conversion for wireless transmission, broadcasting, and streaming media. A more recent trend is the application of machine learning in the sampling and reconstruction pipeline. Deep learning frameworks, such as generative adversarial networks (GANs) and auto-encoders, can be used to learn more complex relationships between partial data and original signals, potentially achieving better speed and accuracy results [9]. In conclusion, the fields of sampling and reconstruction are far from being settled [10]. While the Nyquist–Shannon theorem continues to provide a theoretical foundation, there is active research on new sampling patterns, improved reconstruction algorithms, and smarter methods. In the future, we will likely see a convergence of traditional signal processing with artificial intelligence-driven approaches to design systems that are more accurate, robust, and efficient than ever before [11].

The rest of the paper is organized as follows. Section 2 presents the design learning of sampling and reconstruction of signals. Section 3 describes the design learning of filtering. Methodology is discussed in Section 4, while results and discussion are presented in Section 5. Lastly, Section 6 concluded the work.

2. Design Learning of Sampling and Reconstruction of Signals

Signals sampling and reconstruction refer to the processes of sampling and reconstruction of signals [12]. In other words, it can be defined as the design and implementation of methods for signal sampling and reconstruction. This covers the making of ways and steps for catching and bringing back whole-time signs in a digital way. Sampling, in the setup and study of signals, includes changing an analog signal into a series-by-time form by taking steady checks of the signal [13]. Signal processing follows steps that allow us to analyze, modify, and transmit signals within digital systems. In the first step, sampling extracts distinct samples from a continuous signal at regular intervals. The frequency at which this occurs is referred to as the sampling rate [14]. Measuring the signal's amplitude at set moments creates a series of distinct samples, like quick snapshots that digitally capture the original sound [8]. To rebuild a signal faithfully, the Nyquist–Shannon sampling theorem states to sample at least twice as fast as its highest pitch, like taking two crisp snapshots for every single wave crest [15]. The discrete samples are converted back to the continuous signal through the use of restoration methods. Through the use of interpolation and signal processing algorithms, these techniques endeavor to restore the initial continuous form of the signal. The continuous signal through several methods, such as piecewise linear interpolation, polynomial interpolation, and reconstruction, has been figured out from the discrete samples [16]. The primary parameters that change the selection of a reconstruction algorithm are the characteristics of the signal, the desired accuracy of the reconstruction, and the complexity of the calculation [17]. Sampling and reconstruction methods pop up everywhere these days across different industries. Take telecom as an example, where these techniques are used all the time when converting analog signals to digital formats, which makes processing easier and also helps with storage and transmission [18].

Handling audio data really comes down to nailing the basics of sampling and reconstruction, as recording music into compressed files and playing it back cannot be achieved without getting those core steps applied. People sometimes forget how much tweaking goes into even standard approaches, though new methods keep popping up now and then, which keeps advancing the approaches. Despite many advancements, researchers are still finding smarter ways to capture and rebuild signals without losing quality. However, these two processes, i.e., sampling and reconstruction, pretty much underpin the whole audio tech world [19]. People working on sampling tech these days are pushing hard to get better results with less effort [4]. Signal processing really comes down to nailing the sampling and reconstruction parts. For example, the microphone of a smartphone is used to capture sound waves. That continuous signal gets sliced into digital data points through sampling. Then it is reconstructed later without losing quality. Without that process, computers couldn't analyze or transmit stuff effectively. The better we get at these techniques, the clearer things become. Like streaming HD video without lag or medical imaging, catching tiny details. Teams are always finding smarter ways to sample faster while using less power, which matters for battery-powered devices. Best practices involve balancing sampling rates with storage needs. Too high and you waste resources, too low and you miss critical info. A common approach is to use adaptive methods that adjust based on signal complexity. Results show this maintains fidelity while keeping systems efficient across telecoms, robotics, and in many other domains. That's why sampling and reconstruction stay core to the field, and enable actual real-world tech from voice assistants to satellite comms. The trick is making the digital version behave like the original analog signal did, which sounds simple but takes serious engineering chops to pull off reliably at scale [14]. Figure 1 displays the curriculum properties for signal construction and sampling.

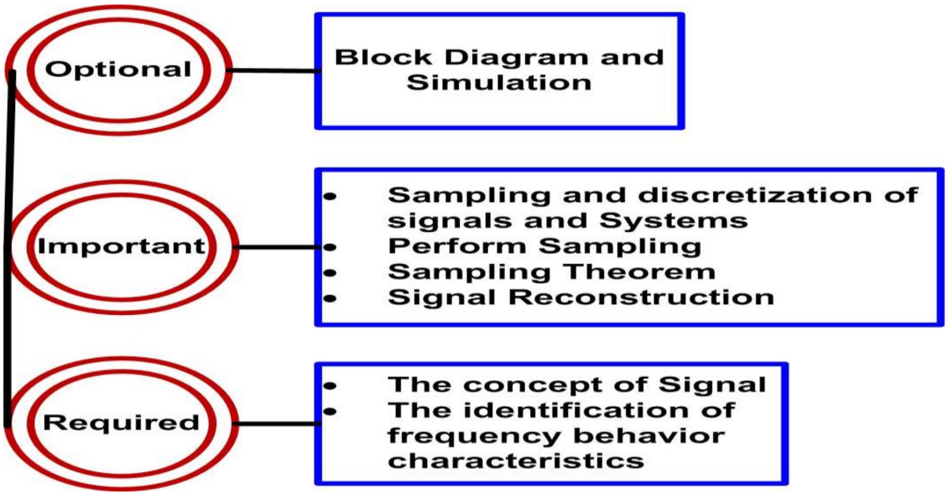

Figure 1: Sampling and reconstruction of signals

2.1. Process of Reconstruction in Signal Processing

Reconstruction in signal processing refers to the process of recreating a continuous signal from its discrete samples. This is commonly encountered in various applications like digital audio processing, image processing, and telecommunications. The reconstruction process typically involves interpolation or filtering to reconstruct the continuous signal from its discrete samples [2]. The continuous musical signal is first sampled at a regular interval to obtain discrete samples. The sampling rate must be at least twice the highest frequency present in the signal according to the Nyquist theorem to avoid aliasing [2].

So high-pass filtering works by applying a filter that cuts out lower frequencies while letting higher ones through. Basically, it takes whatever signal you are working with and blocks the low-end signals. These filters usually set a cutoff point, and anything below that gets reduced or just removed entirely. Most common filters include Butterworth filters, Chebyshev types, and the elliptic ones, too. Each one has its own tradeoffs. Butterworth gives a smooth response but needs more components, while Chebyshev ripples in exchange for steeper roll-off. On the other hand, Elliptic does both passband ripple and stopband stuff, but designs get trickier. The selection of the filter is based on the underlying application, depending on whether phase response is critical or needs sharp attenuation. The real-world implementations often involve op-amps or digital algorithms these days. Another important aspect is component tolerances that affect the actual cutoff frequency in analog circuits [20].

Once filtered out the low-frequency parts, reconstructing the signal happens by either interpolating or running filters on those cleaned-up samples. There are multiple options available here. One could stick with basic zero-order hold for simplicity, use linear interpolation as a middle-ground approach, or go fancier with cubic spline methods when you need tighter accuracy. Different situations call for different tools; basically, it depends on what balance of speed versus precision the application needs at that moment anyway [21].

So high-pass filters in music signals are mainly about cutting down or getting rid of those low-end frequencies, i.e., the rumbly bassy stuff, while letting the crispy highs pass through untouched. Artists and engineers use this for a bunch of reasons. Sometimes it's practical, like stripping out background hum from a recording that got picked up by mics, other times it's more creative, trying to carve space in a mix so certain instruments don't clash. Here, a careful consideration is required regarding how sharp to set the cutoff because going too aggressively, one might accidentally thin out the sound in ways that don't feel natural. It's all about finding that optimal point where the unnecessary lows get dampened without making things feel hollow or tinny, which takes some trial and error method to get the desired results [22].

2.2. Metrics used to measure the quality of Reconstructed Signals

The following metrics are utilized to measure the quality of reconstructed signals [3].

2.2.1. Signals-to-Noise Ratio (SNR)

It measures the ratio of the power of the signal to the power of the noise. A higher SNR indicates better quality reconstruction.

2.2.2. Total Harmonic Distortion (THD)

THD measures the harmonic distortion introduced during reconstruction. Lower THD values indicate less distortion and higher quality reconstruction.

2.2.3. Frequency Response

This measures how accurately the reconstructed signal matches the original signal in terms of frequency content across the spectrum. A flat frequency response indicates better reconstruction quality.

2.2.4. Interpolation Error

For interpolation-based reconstruction methods, the interpolation error measures the difference between the original signal and the reconstructed signal. Lower interpolation error indicates better quality reconstruction.

2.2.5. Subjective Listening Tests

In many cases, the perceived quality of the reconstructed signal by human listeners is also considered as a metric. Listening tests involving trained listeners can provide valuable insights into the subjective quality of the reconstruction.

3. Design Learning of Signal Filtering

A high-pass filter serves to weaken signals below a specified cutoff frequency, known as the stop band, while permitting signals above it, referred to as the pass band. The degree of attenuation varies based on the filter's design parameters. Common applications of high-pass filters include eliminating low-frequency noise, eradicating hums from audio signals, channeling higher frequency signals to suitable speakers within sound setups, and extracting high-frequency trends by filtering out low-frequency components in time-series data [20].

Filter design is the process of creating a signal processing filter that meets a specific set of requirements, which may sometimes be conflicting [22]. The objective is to find a filter configuration that sufficiently fulfills each requirement, making it practical and useful. The design process can be viewed as an optimization problem, where the goal is to minimize an error function that is influenced by each requirement. Through the use of tools, one can automate parts of design work these days. But when it comes down to it, having a pro electrical engineer in the mix usually makes the difference between okay results and stuff that actually works right. These automated systems are handy and help to handle repetitive tasks efficiently. The role of human experts is still there to deal with real-world variables, unexpected hiccups, and exceptionally tricky edge cases. An experienced engineer is able to spot issues that machines would miss, and tweak things in ways algorithms wouldn't think of [23].

Building digital filters isn't just about the math and theories learnt in textbooks. The real deal is way challenging than that when we actually try making them work in practice. That happens a lot with filter designs. Real-world noise, hardware limits, and unexpected signal quirks - they all throw hurdles in the works. That's why there's so much research going into this area. Engineers keep hitting walls with traditional methods when dealing with modern tech demands. Better algorithms, smarter optimization techniques, adaptive systems, balancing performance with computational costs – everyone's scrambling for solutions that hold up outside lab conditions. It's a constant challenge between accuracy and efficiency. Each project brings new challenges with it. What worked for audio processing might crash and burn in medical imaging systems. So, designing these filters stays challenging no matter how much theory is mastered. Researchers are still exploring approaches, trying to crack the code for specific use cases while keeping things stable and reliable [24]. The real challenge comes from all these little factors to consider to get the right performance targets. Basically, it means considering decisions about design stuff like what filter type to use, how to sequence them, dialing in those parameters just right in order to balance trade-offs on the fly. It definitely needs careful thinking through each process adopted, and how it affects the rest of things down the line [25].

For the filter design, basically, it's about designing those signal processors so they actually perform the tasks in the real setups. The goal is getting filters that hit performance targets while dealing with all the constraints that pop up when theory meets actual circuits and code. It comes down to tweaking different design parameters and trying to minimize those errors tied to each requirement. Automation tools help with some parts, but still need seasoned electrical engineers in the audio mix domain. Their input makes or breaks whether you get results that actually work in practice. Like, without that hands-on expertise, things might look good on paper but crash when real-world variables hit. The balance between automated processes and human judgment isn't optional; here it's mandatory for solutions that hold up over time [27]. The practical complexities of digital filter design motivate ongoing advanced research in this area. Regarding the stability of a filter design, it is assessed using various methods depending on the type of filter. For digital filters, stability is often determined by analyzing the filter's transfer function or difference equation. One common method is to check for the location of poles in the z-plane; if all poles are within the unit circle, the filter is stable [22, 28]. The sparsity of a signal quantifies the proportion of its coefficients or elements that are zero or close to zero. Mathematically, sparsity can be measured using metrics [16].

Finite Impulse Response (FIR) filters are inherently stable due to their structure and properties. Unlike Infinite Impulse Response (IIR) filters, which can have feedback loops that may lead to instability, FIR filters only have feed-forward structures [23]. The impulse response of an FIR filter is finite in length, which ensures bounded output for bounded input, leading to stability. Additionally, FIR filters can be designed to have linear phase response, making them suitable for applications where phase distortion must be minimized [1]. Designing high-pass filters using MATLAB involves utilizing the signal processing toolbox to implement and customize digital filters that selectively attenuate low-frequency components while allowing high-frequency components to pass through. Here is a general outline of the steps involved in designing high-pass filters using MATLAB [29]. Define the desired cutoff frequency, filter order, and any additional specifications, such as stop-band attenuation or pass-band ripple [30]. MATLAB provides functions like "butter," "cheby1," or "ellip" to design filters based on the selected method. Then plot the frequency response of the designed filter to examine its characteristics. MATLAB's "freqz" function can be used to obtain and visualize the frequency response [23]. Once the filter is designed, it can be applied to input signals using the "filter" function. Assess the performance of the high-pass filter by analyzing its output signal. Compare it to the original signal and check if the desired high-frequency components are preserved while the low-frequency components are adequately suppressed. If the filter does not meet the desired specifications, you can refine the design by adjusting the filter order, changing the cutoff frequency, or exploring different filter design methods [31]. The formula for the Cut-off Frequency of a high-pass filter is the same equation as that of the low-pass filter, and the formula for the phase shift high-pass filter is given below [29]. The high-pass filter is shown in Figure 2.

$$f_c = \frac{1}{2\pi fRC} \tag{1}$$

Where phase shift is denoted by Φ

$$\Phi = arctan \frac{1}{2\pi fRC} \tag{2}$$

Circuit gain is denoted by AV

$$A_V = \frac{V_{out}}{V_{in}} = \frac{R}{\sqrt{R^2 + X_c{}^2}} = \frac{R}{Z} \tag{3}$$

At low frequencies, the X_c is infinity then $V_{out} = 0$

At high frequencies, the X_c is zero then $V_{out} = V_{in}$

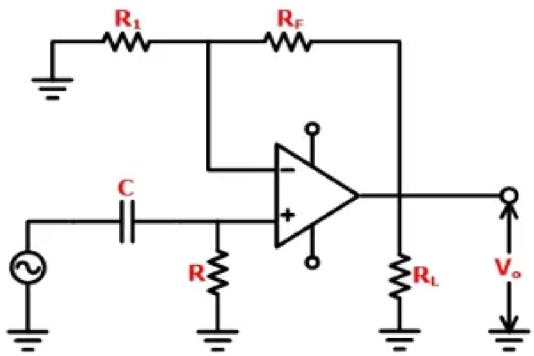

Figure 2: High-pass filter

4. Methodology

Implementing a Project-Based Learning (PBL) task involves several key steps that aim to engage learners in hands-on, inquiry-based activities to develop knowledge, skills, and problem-solving abilities. Here are the steps involved in implementing a PBL task.

First of all, clearly define the learning objectives or outcomes that you want students to achieve through the PBL task. Start by making sure everything lines up with what's actually taught in class standards, both the knowledge they must have and the skills they should acquire along the way. Create scenarios that seem authentic, such as issues that real people may encounter, rather than only textbook scenarios. Create a topic that is engaging and relevant to students, sparking their curiosity and prompting them to think about possible solutions. Gather necessary resources such as tools, books, and websites, and potentially seek advice from local experts, without providing excessive guidance. Allow them to determine which resources to utilize, even if they encounter some difficulties along the way. Planning phase needs to be structured, but not too much. Help them break big tasks into steps, and assign who does what without micromanaging. Check-ins should happen naturally, glance at their notes, ask how it's going, and nudge them back on track when they wander off. Let them present work in ways that make sense, posters, videos, maybe even teach backs to younger grades. Facilitate a post-project reflection session where students can evaluate their learning, problem-solving strategies, and collaboration skills. Encourage them to identify strengths, challenges, and areas for improvement. The next step is to implement the sampling process. This involves converting the continuous-time signal into a discrete-time signal by sampling it at regular intervals. The MATLAB function 'sample' can be used to implement this process. Results presented by a student team, using discrete devices (transistors, resistors, and capacitors) shown in Figure 3.

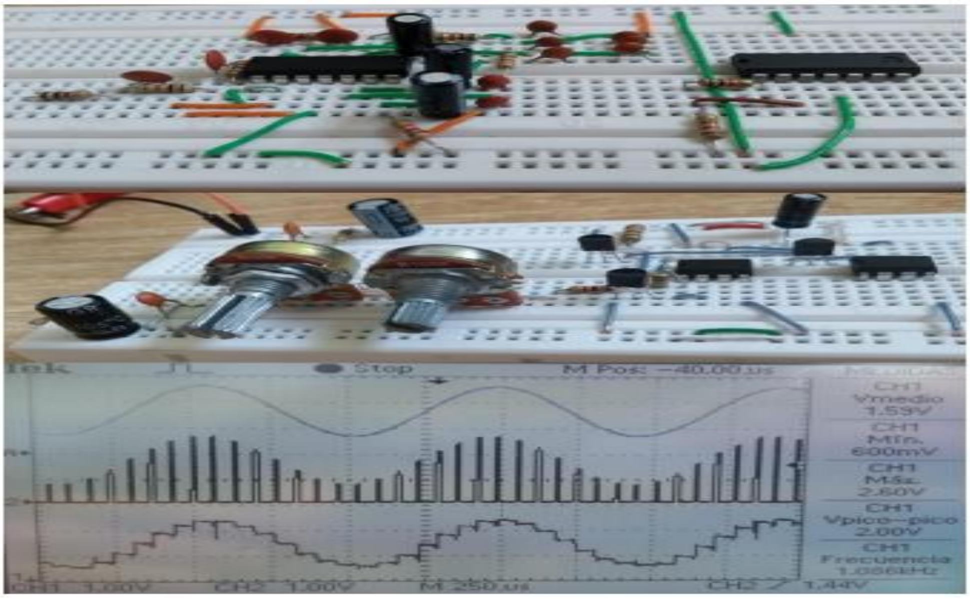

Figure 3: Results with discrete devices

5. Results and Discussions

5.1. High-Pass Filtering of Musical Signals

High-pass filtering of musical signals is a common signal processing technique used to emphasize or extract high-frequency components while attenuating or removing low-frequency components. It finds applications in various audio processing tasks, such as noise reduction, audio equalization, and instrument separation. When performing high-pass filtering on a musical signal using MATLAB or any other signal processing tool, the following steps can be followed:

- Load the musical signal into MATLAB by reading an audio file or generating a synthetic waveform. MATLAB provides functions like "audio read" or "sound sc" to facilitate audio file loading and playback.
- Determine the desired cutoff frequency for the high-pass filter. The cutoff frequency defines the point below which the low-frequency components will be attenuated. Consider the characteristics of the musical signal and the specific frequency range you want to emphasize.
- Select a suitable filter design method based on your requirements, such as Butterworth, Chebyshev, or elliptic filters. Use MATLAB's filter design functions, such as "butter" or "cheby1," to design the high-pass filter based on the chosen method and the desired cutoff frequency.
- Utilize the designed high-pass filter to process the musical signal. MATLAB's "filter" function can be employed to apply the filter to the audio data. This operation will attenuate the low-frequency components, allowing the higher-frequency elements to pass through.
- Play the filtered musical signal using MATLAB's audio playback functions like "sound" or "sound sc" to audition the result.
- Assess the impact of the high-pass filtering on the audio, focusing on the emphasized high-frequency components and the reduced low-frequency components.

If the filtering outcome does not meet your expectations, consider adjusting the filter parameters. Experiment with different cutoff frequencies, filter orders, or design methods to achieve the desired emphasis on high-frequency content while preserving the musical quality.

FIR filters offer inherent stability, making them highly desirable. Their ability to maintain waveform shape while introducing a delay in the filtered signal makes them particularly appealing. However, their long transient responses and potential computational expense can be drawbacks in certain applications. FIR filters find utility in various fields such as audio processing, biomedical engineering, and radar technology, where preserving waveform characteristics is crucial. Noteworthy design techniques for low-pass FIR filters encompass the Kaiser Window, least squares, and equi-ripple methods.

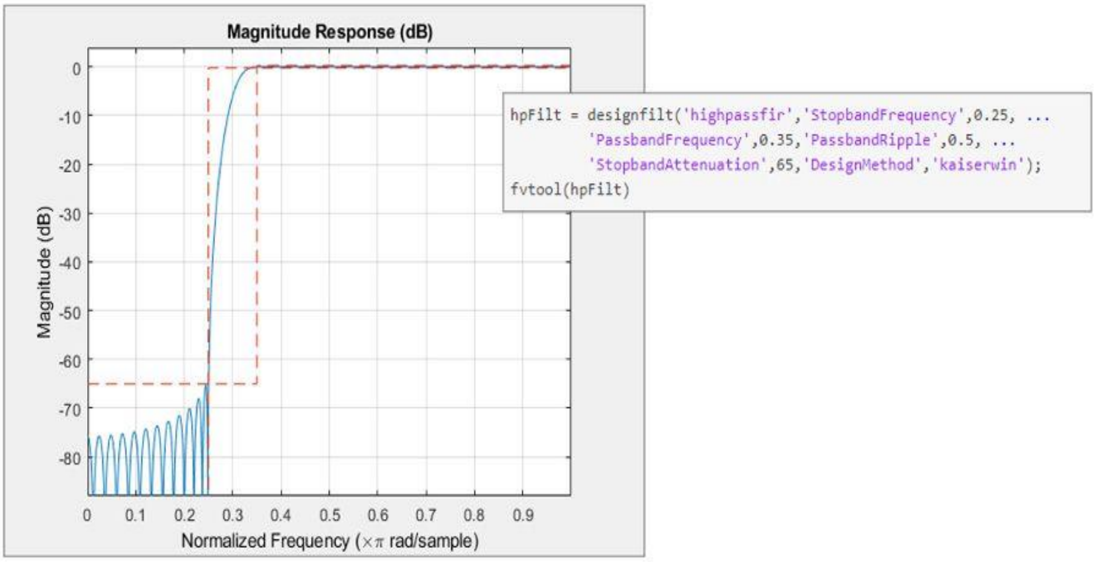

Figure 4(a): Design specifications and response of a high-pass Kaiser FIR filter in MATLAB

5.2. Discussions

We explored various sampling techniques, including uniform sampling, non-uniform sampling, and compressed sensing. We found that the choice of sampling technique has a significant impact on the

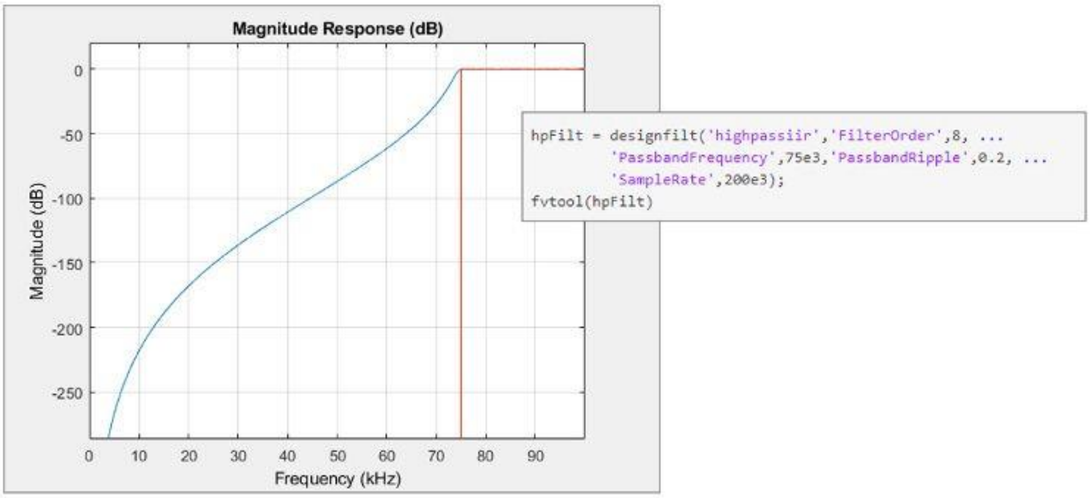

Figure 4(b): Design specifications and response of a high-pass Butterworth IIR filter in MATLAB

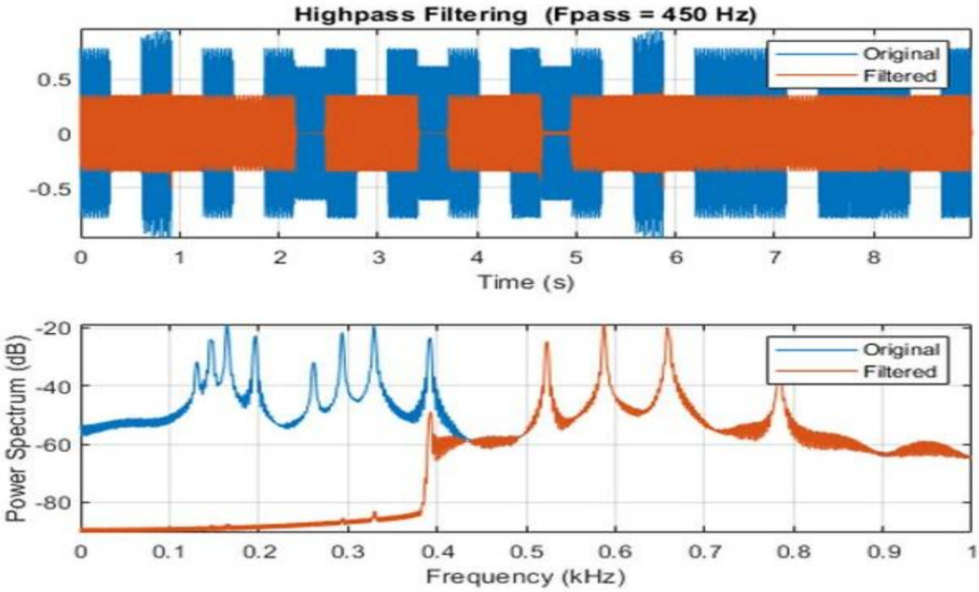

Figure 5. High-pass filtering of musical signals

quality of the reconstructed signal. Uniform sampling did well for low-frequency signals. Compressed sensing worked better when dealing with sparse representations. We tested multiple reconstruction methods, like Nyquist-Shannon and L1-minimization wavelets. L1-optimization performed exceptionally well on sparse signals while Nyquist-Shannon held up for band-limited cases. The evaluation metrics, i.e., PSNR and MSE, showed better results when combining advanced sampling with smarter reconstruction methods, especially in undersampled situations. Whereas high compression ratios sometimes wiped out details, messing with signal quality, as there are always tradeoffs: sample more, get better quality, but need more storage power. Sampling less saves resources but risks accuracy. The main idea is to align your sampling technique with the type of signal being measured. Non-uniform and compressed sensing methods excel in situations where uniform sampling is not as effective. Here, the selection of reconstruction algorithms is based on the specific requirements of sparse signals and computational constraints. Engineers consider various factors in the fields of image processing and data compression as they strive to achieve high compression rates without compromising quality. People continue to struggle with maintaining balance while cutting data without compromising accuracy. Research requires improved methods for reconstructing information in difficult circumstances.

6. Conclusions

Handling signals through sampling and reconstruction forms the backbone of how we turn real-world data into digital formats people can work with. The Nyquist-Shannon theorem basically set the rules for this whole game decades back, paving the way for the modern approaches we rely on today. Advancements made in this field have resulted in the widespread implementation of practical innovations. Take non-uniform methods or compressed sensing. These let us grab fewer samples without losing critical information, which matters big time when dealing with bandwidth limits or storage constraints. Standard interpolation still plays a role, but newer algorithms handle trickier signal types better now. Accuracy keeps improving as these tools get refined across different use cases. Once you start looking, applications are everywhere. Image processing benefits massively from these techniques, think medical imaging or video streaming, where clarity matters. The machine learning aspect has been gaining traction recently. Deep learning is making waves here by tackling incomplete or messy data that older methods struggle with. Teams are using stuff like GANs and auto encoders to fill gaps in sensor readings or clean up distorted signals automatically. Results show promise, but there's still work to do there. At its core, these processes let us preserve and use signals effectively across industries that are not changing anytime soon. What keeps things moving forward is the constant tweaking of existing methods while exploring new combos with emerging tech like AI tools we're seeing now helps push boundaries further than before.

Acknowledgement

We are thankful to the research reviewers who helped in writing and improving these guidelines.

Conflict of Interests

Publication of this research article has no conflict of interests.

Funding Statement

This research article is not a product of a funded research project.

7. References

[1] C. Zeng, J. Ye, Z. Wang, N. Zhao, and M. Wu, "Cascade neural network-based joint sampling and reconstruction for image compressed sensing," *Signal, Image and Video Processing,* vol. 16, pp. 47–54, 2022.

[2] A. Pizzo, A. de Jesus Torres, L. Sanguinetti, and T. L. Marzetta, "Nyquist sampling and degrees of freedom of electromagnetic fields," *IEEE Transactions on Signal Processing,* vol. 70, pp. 3935–3947, 2022.

[3] L. Shen, J. Pauly, and L. Xing, "NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 35, no. 1, pp. 770–782, 2022.

[4] R. E. Carrillo, K. E. Barner, and T. C. Aysal, "Robust sampling and reconstruction methods for sparse signals in the presence of impulsive noise," *IEEE Journal of Selected Topics in Signal Processing,* vol. 4, no. 2, pp. 392–408, 2010.

[5] A. Bhandari, F. Krahmer, and R. Raskar, "On unlimited sampling and reconstruction," *IEEE Transactions on Signal Processing,* vol. 69, pp. 3827–3839, 2020.

[6] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus, "Sampling and reconstruction of signals on product graphs," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 713–717.

[7] P. Feng and Y. Bresler, "Spectrum-blind minimum-rate sampling and reconstruction of multiband signals," In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996, pp. 1688–1691.

[8] J. A. Tropp, M. B. Wakin, M. F. Duarte, D. Baron, and R. G. Baraniuk, "Random filters for compressive sampling and reconstruction," In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006, pp. III-III.

[9] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Processing Magazine,* vol. 37, pp. 14–30, 2020.

[10] D. Patel and S. Sivananthan, "Random sampling of signals concentrated on compact set in localized reproducing kernel subspace of L p (R n)," *Advances in Computational Mathematics,* vol. 49, no. 6, p. 76, 2023.

[11] D. Wang, X. Liu, X. Wu, and Z. Wang, "Reconstruction of periodic band limited signals from non-uniform samples with sub-Nyquist sampling rate," *Sensors,* vol. 20, p. 6246, 2020.

[12] S. Ravishankar, J. C. Ye, and J. A. Fessler, "Image reconstruction: From sparsity to data-adaptive methods and machine learning," *Proceedings of the IEEE,* vol. 108, pp. 86–109, 2019.

[13] A. Septimus and R. Steinberg, "Compressive sampling hardware reconstruction," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 3316–3319.

[14] A. Bhandari and P. Marziliano, "Sampling and reconstruction of sparse signals in fractional Fourier domain," *IEEE Signal Processing Letters,* vol. 17, pp. 221–224, 2009.

[15] M. Mishali and Y. C. Eldar, "Sub-nyquist sampling," *IEEE Signal Processing Magazine,* vol. 28, pp. 98–124, 2011.

[16] L. Stanković, E. Sejdić, S. Stanković, M. Daković, and I. Orović, "A tutorial on sparse signal reconstruction and its applications in signal processing," *Circuits, Systems, and Signal Processing,* vol. 38, pp. 1206–1263, 2019.

[17] C. La and M. N. Do, "Signal reconstruction using sparse tree representations," in *Wavelets XI*, 2005, pp. 273–283.

[18] K. K. Sharma and S. D. Joshi, "Signal reconstruction from the undersampled signal samples," *Optics Communications,* vol. 268, pp. 245–252, 2006.

[19] I. Maravic and M. Vetterli, "Sampling and reconstruction of signals with finite rate of innovation in the presence of noise," *IEEE Transactions on Signal Processing,* vol. 53, no. 8, pp. 2788–2805, 2005.

[20] J. Kovacevic and M. Vetterli, "Perfect reconstruction filter banks with rational sampling factors," *IEEE Transactions on Signal Processing,* vol. 41, no. 6, pp. 2047–2066, 1993.

[21] A. H. Soomro, A. S. Larik, M. A. Mahar, and A. A. Sahito, "Simulation-Based Comparison of PID with Sliding Mode Controller for Matrix-Converter-Based Dynamic Voltage Restorer under Variation of System Parameters to Alleviate the Voltage Sag in Distribution System," *Sustainability,* vol. 14, no. 21, p. 14661, 2022.

[22] I. W. Selesnick, H. L. Graber, D. S. Pfeil, and R. L. Barbour, "Simultaneous low-pass filtering and total variation denoising," *IEEE Transactions on Signal Processing,* vol. 62, no. 5, pp. 1109–1124, 2014.

[23] R. S. Prendergast, B. C. Levy, and P. J. Hurst, "Reconstruction of band-limited periodic nonuniformly sampled signals through multirate filter banks," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 51, no. 8, pp. 1612–1622, 2004.

[24] Y. Ma, J. Wu, X. Jiang, B. Xu, K. Ma, H. Wu*, et al.*, "An Efficient Waveform Reconstruction Method for Digital Bandwidth Interleaving Sampling System," *IEEE Transactions on Nuclear Science,* vol. 70, no. 6, pp. 873–881, 2023.

[25] A. N. Delopoulos and S. D. Kollias, "Optimal filter banks for signal reconstruction from noisy subband components," *IEEE Transactions on Signal Processing,* vol. 44, no. 2, pp. 212–224, 1996.

[26] B.-S. Chen, C.-W. Lin, and Y.-L. Chen, "Optimal signal reconstruction in noisy filter bank systems: Multirate Kalman synthesis filtering approach," *IEEE Transactions on Signal Processing,* vol. 43, no. 11, pp. 2496–2504, 1995.

[27] M. K. Tsatsanis and G. B. Giannakis, "Principal component filter banks for optimal multiresolution analysis," *IEEE Transactions on Signal Processing,* vol. 43, no. 8, pp. 1766–1777, 1995.

[28] A. H. Soomro, G. S. Kaloi, S. A. A. Shah, and M. A. Koondhar, "Mathematical Modeling and Simulation of AC-AC Three Phase Matrix Converter with LC Filter driving Static Resistive Load," *Quaid-E-Awam University Research Journal of Engineering, Science & Technology, Nawabshah.,* vol. 20, no. 2, pp. 107–113, 2022.

[29] E. Elias, "Prototype filter design approaches for near perfect reconstruction cosine modulated filter banks-a review," *Journal of Signal Processing Systems,* vol. 81, no. 2, pp. 183–195, 2015.

[30] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal processing,* vol. 10, no. 3, pp. 219–244, 1986.

[31] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," *Proceedings of the IEEE,* vol. 78, pp. 56–93, 1990.