Teaching Aid for Digital Signal Processing A Case Study on Negative Frequency, Aliasing, Fourier Transform, and Filtering

Lavanya M S
Dept. of ECE
The National Institute of Engineering
Mysuru, India

Vishwanath M K
Dept. of ECE
The National Institute of Engineering
Mysuru, India

***Abstract*—Signal processing has a lot of concepts like negative frequency among others. Normally students find it difficult to grasp these concepts. In this paper, we propose to demonstrate some of the concepts through experimentation. We have selected concepts of negative frequency, aliasing, and signal decomposition among others. We feel that the students would be able to grasp the concepts and appreciate the signal processing subject in a better way.**

Keywords—Sampling, aliasing, negative frequency, Fourier Transform, DSP.

I. INTRODUCTION

Earlier, engineering education primarily focused on conventional classroom teaching approaches where most of the courses offered were concentrated on helping students understand the basic theoretical concepts of different areas in engineering. Digital signal processing (DSP) is one such important area, which has developed rapidly over the past few years and has a wide range of applications in many interesting areas such as space science, communication, robotics, medical, military, automotive, consumer electronics, etc. It also provides a basic foundation for booming areas such as artificial intelligence and machine learning [1] and hence learning the course has become most important in engineering education. Hence many universities include DSP as an integral part of their undergraduate course, mainly in the field of electrical and electronics engineering. But the course still has more focus on basic theory and mathematical representations rather than practical implementations [2], and hence, students find it very difficult to grasp the concepts

A major shift has to be made from conventional methods to including the laboratory as an integral part of the course to make the course more interesting and enjoyable. Laboratory courses play an important role in bridging the gap between conventional learning methods and real-world problems. It also helps in bridging the gap between industry and academia. In this regard, various universities and institutions include the DSP laboratory as a part of the DSP course. DSP laboratory courses encourage students to develop numerical algorithms based on mathematical representations using tools such as MATLAB. The availability of many open-source tools such as GNU Octave and SciLab has made it far more affordable and easier for students to learn the course effectively. These tools are also available on the cloud (online) and hence they are far more easily accessible to students.

In this paper, we propose to demonstrate experiments on the basic concepts of DSP such as sampling, aliasing, Fourier transforms and the FFT to help students grasp the concepts in a better way. The open-source Octave simulation tool is used to demonstrate these experiments. Harmonics and filtering are demonstrated using CCS studio and DSK 6713 kit.

A review of literature on the relevant topics and various challenges in teaching DSP are described in sections II and III respectively. Section IV explains conventional and experimental methods in teaching DSP and the results obtained from experiments.

II. REVIEW OF LITERATURE

Faculty from different engineering institutions and universities have developed various algorithms and methods to explain various DSP concepts, such as filters, Fourier transforms, DFT, FFT, etc.

Keshab K. Parhi et al [1] have proposed that the effectiveness of teaching digital signal processing can be enhanced by reducing lecture time devoted to theory and increasing emphasis on applications, programming aspects, visualization, and intuitive understanding. The author has observed that student engagement and interest in learning the subject can be enhanced by engaging students to work in groups during the class where the students can solve problems and programming assignments or take quizzes.

Fernando A. Mujica et al [2] have described their efforts to develop an open-source “Stanford Lab in a box” system that helped students to see how fundamental concepts of DSP can be implemented in real-time on a fixed-point processor. The authors have mentioned that the system was easy to use with simple Arduino-like programming and helped in providing a new dimension to the teaching of DSP.

Shang et al [3] have designed an experimental system based on TMS320VC5509 consisting of many functions such as signal acquisition, serial communication speech signal processing, image processing, etc. The authors suggest that the system can be applied to the practical teaching of DSP courses and can be used as a development platform for the DSP system. They have also claimed that the students agreed that the lab component enhanced their learning of theoretical concepts.

Hemakumar [4] has proposed the usage of skits and employing audio-visual tools for learning DSP. Based on the pre-test and post-test scores of different groups, the author has observed that groups using both skits and audio-visual tools performed better than the groups using only audio-visual tools. He has also observed that usage of these interactive teaching aids has helped in increasing the student's interest to learn, visualize and understand the key concepts of DSP.

Divya Joseph et al [5] have presented the use of a windows store app that can be used as a teaching aid for undergraduate DSP courses. They have designed an app consisting of six modules, four of which are demonstrating problems on linear convolution, circular convolution, FFT, FIR filter design, etc., and the other two forming a DSP toolkit based on a hypothetical 16-bit floating-point processor, with a simple instruction set. The authors have claimed that the app has received good feedback from students.

Guo Luo et al [6] have introduced an engineering practice reform of experimental teaching into engineering application cases, which can be used to apply theoretical knowledge to process actual signals. The authors have observed significant improvement in students' learning ability and their capability to process actual engineering signals through these reforms.

In the literature reviews above, all the authors have suggested the usage of various processors, simulation software, apps, and usage of different approaches like skits and group activities to stimulate the student's interest in learning the DSP course. But, teaching the course to a classroom consisting of students with different thinking levels poses various challenges.

III. CHALLENGES IN TEACHING DSP

Teaching and learning courses like DSP that involves mathematical detail is always a challenge for both educators and students. Several textbooks [7][8] used at the undergraduate level of study incorporate many abstract concepts to make the students understand the basic theory and its mathematical implementations. Students find it difficult to grasp the concepts of signals and their behavior in different domains. Many students struggle to pass the introductory levels of the course and hence are reluctant to choose electives in related domains. To enhance students' enrolment in higher-level signal processing courses and to motivate them to pursue careers in related domains, it is necessary to stimulate their interest by making them understand the basic concepts thoroughly

IV. CONVENTIONAL AND EXPERIMENTAL METHOD

To help students to understand the basic concepts of signal processing such as aliasing, negative frequency, FFT, etc, simple experiments are designed using open-source GNU octave software. Also, the concepts of filtering and Fourier Transforms are explained with the help of an experiment performed using the DSK6713 kit.

A. Aliasing and Negative Frequency

1. Conventional Methods

A real-time signal is always a random signal that can be mathematically modeled as the sum of multiple complex sinusoids of different frequencies and amplitudes. Understanding the aliasing of a sinusoid can help us in understanding the concept of the aliasing of real-time signals.

From Euler's formula, a complex sinusoid is defined in equation (1)

$$\sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j} \qquad \text{------------(1)}$$

Where $\theta = \omega t + \phi$, and $\omega = 2\pi f$. From the above equation, we can observe that a complex sinusoid is a function of both positive and negative frequencies. When it comes to understanding the concepts of sampling and aliasing of real-time signals, negative frequencies should be considered. But the whole concept is beyond the level of understanding of students. Now, if we say that the term negative implies the direction of rotation of the signal, and it represents the phase, it may be easy for students to understand. The phase of the signal changes depending on the rate of change in frequency. If the reconstructed signal has a lesser frequency and shifted phase, it represents aliasing at negative frequencies [9]

2. Experimental Method

The following algorithm demonstrates the concepts of negative frequency and aliasing

a. Define the frequency of the message signal
b. Define the frequency of the sampling signal
c. Plot a sinusoidal signal with message signal frequency
d. Sample the signal at different frequencies
e. For different frequencies with step size 0.5, plot the input signal and the sampled signal
f. Observe the waveforms

The negative frequencies and effect of aliasing were observed for different frequencies from 0.5-22Hz. The effect of negative frequencies could be observed at frequencies 9-11Hz and 19-21 Hz. The below figure shows the aliasing effect at 9 Hz

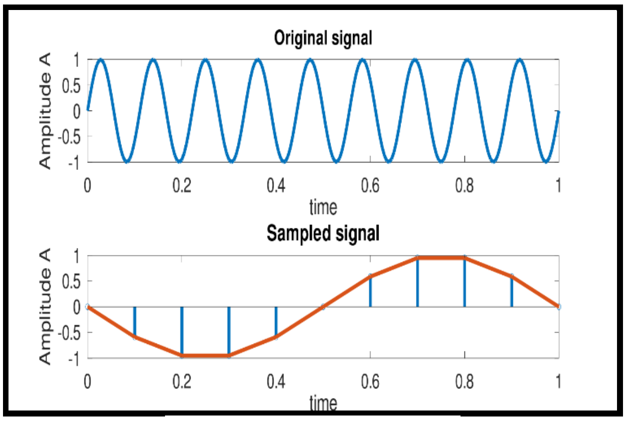

Fig 1: Negative Frequency at 9 Hz

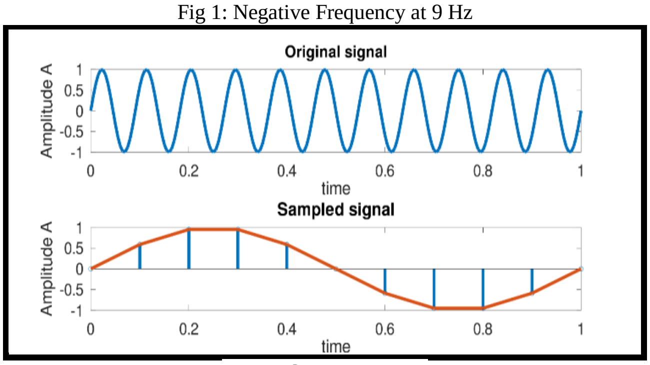

Fig 2: Aliasing at 11 Hz

From Fig 1, it can be observed that, when the analog input is 9 Hz, the sampled output is 1Hz, but the phase is reversed. This change of phase demonstrates the concept of negative frequency. Also, from Fig 2, it can be observed that, when the input is 11Hz, the sampled output is 1Hz. That is, the sampled version of a high-frequency component is a low-frequency component. This phenomenon where a high-frequency component is getting mapped to low-frequency is aliasing.

B. FFT Time Computations

1. Conventional Method

Discrete Fourier Transform [DFT] is used to convert time-domain signals to frequency domain signals.
The general equation for DFT given in equation (2)

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-j2\pi kn/N} \text{ --------(2)}$$

From equation (2), it can be seen that solving the DFT involves complex mathematical computations such as complex multiplications and addition. To compute an N-point DFT, we require N^2 complex multiplications and N(N-1) complex additions. Using Fast Fourier Transform [FFT], the number of complex multiplications and additions required to compute an N point DFT reduces to $(N/2)\log_2 N$ and $N \log_2 N$. Fast Fourier Transform is the most efficient way to perform complex mathematical computations. It increases the speed and accuracy of computations [10]

2. Experimental Method

To demonstrate that FFT consumes less time than DFT to perform computations, a code is written using the following algorithm

a) Define the value of N for computation of DFT
b) Generate a random signal
c) Start the timer
d) Compute DFT and FFT of the random signal using inbuilt commands
e) Stop the timer
f) Note down the elapsed time

The DFT and FFT were performed for 4096, 8192 computations (i.e, the value of N), and the time required for computations were observed. The results obtained were as mentioned below

i) For N=8192
    Using FFT:
    Elapsed time is 0.0385551 seconds.
    Using DFT
    Elapsed time is 1.18462 seconds.

ii) For N=4096
    Using FFT:
    Elapsed time is 0.0385189 seconds.
    Using DFT:
    Elapsed time is 0.169281 seconds.

From the results, it can be observed that the time required for computations using DFT is much greater than that of FFT. Also, the time has rapidly increased with the increase in number of computations using DFT. Hence, we can conclude that FFT increases the speed of computations and is hence more efficient for complex mathematical computations.

C. Filtering and Fourier transform

1. Conventional Method

Any signal can be represented in the time domain and frequency domain. The time domain is the language of the real world. From the time of birth, most of our experiences are developed in the time domain. But owing to the simplicity of frequency domain analysis, most of the signal analysis is carried out in the frequency domain, and hence it is important to know some important aspects of the frequency domain. The frequency domain is not real and it is just a mathematical construct [11]. The sine wave is the language of the frequency domain, which means that the only kind of waveforms that exist in the frequency domain are sine waves and hence very unique. Certain properties make sine waves very unique.

i) They are well defined mathematically.
ii) Any waveform in the time-domain can be described by the combination of sine waves.
iii) Each component of the sine wave can be separated from every other component.
iv) They have a value everywhere, which means that they can be used to describe any real-world waveform.

From the property ii, it can be seen that, to represent a digital signal in the time domain, we can combine multiple components of sine waves, known as harmonics, as illustrated in Fig.3 [11]

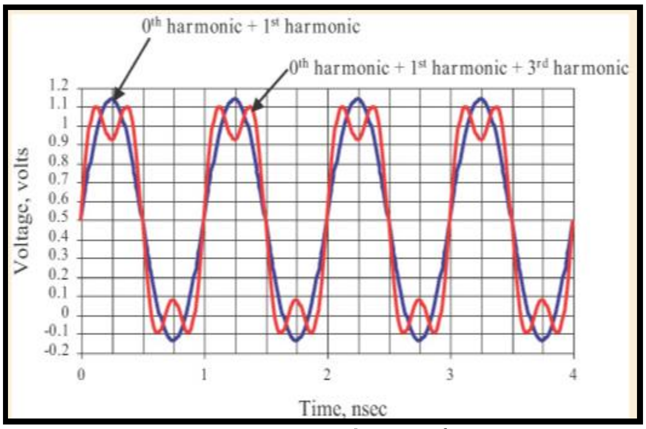

Fig 3: Square as a combination of sine

Hence a square wave is the summation of different order harmonics of sine waves [11]

2. Experimental Method

DSK-6713 kit is used to implement a digital filter; through which the harmonic content of a square wave can be demonstrated. A real-world audio signal is passed through this low-pass filter. A simple code for LPF is written in C-language using code composer studio for DSK6713 kit and filter coefficients are obtained from MATLAB. The input and output waveforms are observed in a CRO for 100 Hz and 400 Hz

Fig 4: 100 Hz square wave and its filtered output

Fig 5: 400 Hz square wave and its filtered output

If a very low-frequency square wave of 100Hz is passed through the filter, it can be observed that the output will be almost a square wave (Fig 4). Since the filter passes more harmonics of the sinusoid at low frequencies, reconstruction resembles a square wave, that is, it resembles the original waveform. Each harmonic will give the sinusoid at the edge of the passband. At 400 Hz (Fig 5), it can be observed that the output resembles a sine wave. From this, it can be explained that the Fourier representation breaks the signal into its sinusoidal components. This experiment demonstrates that the square wave can be decomposed into many harmonics. The experiment can also be performed with other waveforms like a triangular wave.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed methods to demonstrate experiments related to the basic concepts of DSP such as DFT, FFT, Aliasing, Negative frequency, filters, and different domains in which signals can be represented. This has proven to be an effective approach for imparting the concepts to the students and increasing their understanding, Further, more real-time applications can be implemented to encourage students to pursue careers in related fields.

VI ACKNOWLEDGEMENT

The authors wish to convey our heartfelt thanks to Dr. Narasimha Kaulgud, Professor, Department of Electronics and Communication Engineering, The National Institute of Engineering, Mysuru, for his constant support and encouragement throughout this endeavor.

REFERENCES

[1] K. K. Parhi, "Teaching Digital Signal Processing by Partial Flipping, Active Learning, and Visualization: Keeping Students Engaged With Blended Teaching," in IEEE Signal Processing Magazine, vol. 38, no. 3, pp. 20-29, May 2021

[2] Fernando A. Mujica, William J. Esposito, Alex Gonzalez, Charles R. Qi, Chris Vassos, Maisy Wieman, Reggie Wilcox, Gregory T. A. Kovacs, and Ronald W. Schafe, "Teaching digital signal processing with Stanford's Lab-in-a-Box," IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE), pp. 307-312, 2015.

[3] Shang, Qiuli and W. Liu. “Multi-function DSP Experimental System Based on TMS320VC5509.” *DEStech Transactions on Social Science, Education and Human Science*, 2017.

[4] Hemakumar, B “Teaching digital signal processing using skits – A case study.”, International Journal of Current Engineering and Scientific Research (IJCESR), Volume-5, pp 35-40 issue-1, 2018

[5] Diya Joseph, Prateek Chawla, Harish Chander, Tarun Kumar Rawat “A multipurpose toolkit for teaching DSP in an undergraduate course”, Computer Applications in Engineering Education. Wiley Publishers, Volume 25, Issue 3, pp. 530–541, May 2017

[6] Luo, Guo & Huang, Lihua & Zhan, Xiaohong & Ye, Jiesheng, “The Course Reform of Digital Signal Processing for Engineering Education”, International Conference on Education Science and Social Development,2018

[7] Monson H Hayes, “Digital signal processing”, Tata McGraw-Hill, New Delhi, 3rd edition, 2008.

[8] Oppenheim and Schaffer, “Discrete-Time Signal Processing”, PHI,2014.

[9] Steven W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, Second Edition, California Technical Publishing, 1999

[10] Proakis and Manolakis, “Digital Signal Processing – Principles algorithm and application”, Pearson Education 4th Edition, 2007.

[11] Eric Bogatin, “Signal and Power Integrity-simplified”, Pearson 3rd edition, 2018