

Supplementary material:

<https://doi.org/10.1109/MSP.2021.3052487> -> media

Teaching Digital Signal Processing by Partial Flipping, Active Learning, and Visualization

Keeping students engaged with blended teaching



©SHUTTERSTOCK.COM/LUCKYSTEP

The effectiveness of teaching digital signal processing (DSP) can be enhanced by reducing lecture time devoted to theory and increasing emphasis on applications, programming aspects, visualization, and intuitive understanding. An integrated approach to teaching requires instructors to simultaneously teach theory and its applications in storage and processing of audio, speech, and biomedical signals. Student engagement can be enhanced by having students work in groups during class, where they can solve short problems and short programming assignments or take quizzes. These approaches will increase student interest and engagement in learning the subject.

Introduction

DSP is used in numerous applications, such as communications, biomedical signal analysis, health care, network theory, finance, surveillance, robotics, and feature extraction for data analysis. Learning DSP is more important than ever before because it provides the foundation for machine learning and artificial intelligence.

The DSP community has benefited tremendously from Oppenheim's views of education [1], [2] and from his many field-shaping textbooks. Teaching an engineering class in general and the DSP class in particular is very different today from 30 years ago when computers, tools, and data were not available in abundance. Shulman captures how classes with significant mathematical content were taught in the past [3]. He describes specifically how a professor teaches fluid dynamics: "He is furiously writing equations on the board, looking back over his shoulder in the direction of the students as he asks, of no one in particular, 'Are you with me?' A couple of affirmative grunts are sufficient to encourage him to continue..." This is a form of teaching that engineering shares with many of the other mathematically intensive disciplines and professions; it is not the 'signature' of engineering." The author is right in that, although some instructors teach engineering this way, it is not and should not be our teaching signature. When I taught the DSP class at the University of Minnesota (UMN) using the Oppenheim–Schafer textbook [4] in the fall of 1989,

my teaching signature was close to what Shulman describes. But over the last three decades, my teaching signature has changed significantly. In this article, I will describe my teaching signature as I practice it today.

The objectives of teaching are threefold: 1) teach the necessary mathematical theory and derivations, 2) introduce sufficient applications and visualize the results by programming the application, and 3) present intuitive insight about the observations from the programming experiments. Thus, signal processing is as much about listening to sounds and visualizing temporal and spectral representations as about theoretical problem solving.

My teaching signature can be described as “blended teaching.” I teach mostly by writing down the content in class. This helps the students write down what I am writing; this then helps them develop the same thought process as I do when I derive these results. I then switch to PowerPoint slides, and I show graphs and plots and play audio sounds to see how a signal sounds after a certain filtering operation or how different types of filters change the signal to different forms. We filter some music and sinusoids and discuss some MATLAB code. This blended teaching keeps the students engaged. I try to assign homework problems that relate to real applications. I remind myself of the threshold concepts [5]. While thresholds differ for different students, I try to cover as many if not all of these concepts. In signal processing, we have many threshold concepts. They range from myriads of math tricks to challenges in applying the same concepts to different problems. The most challenging part of teaching is to really pretend that we are not experts but novices. Then we can teach other novices more effectively.

Challenges in teaching DSP

When I took the DSP class at the University of Pennsylvania in 1983 using the Oppenheim–Schafer textbook [6], it was taught as an advanced graduate course at that time. The DSP class is taught today as a senior elective at most universities.

There is a desire to teach the class as a practical class, where signals, sounds, and images can be manipulated using DSP. This manipulation should be integrated into the lecture as well as homework. One of the challenges is that the textbooks are rich in theory but do not provide a sufficient number of practical applications. The textbook by Mitra, however, provides numerous applications related to multirate and sample-rate alteration [7]. Because the class is often taught with an emphasis on theory, many students lose interest in taking it. Such a class is an elective class. So to increase enrollment, students should find the class interesting and practical. We also need to train students to acquire the practical skills that will help them in their jobs in industry. However, we also need to teach the mathematical rigor for students. In the absence of an ideal textbook, this places a burden on instructors to design application examples to be covered during lectures and applications to be assigned as

part of the homework. We have already seen some local success in this direction [8].

There is also a need to increase student engagement and interest. This requires instructors to deviate from traditional teaching and adopt some form of flipped teaching, where students familiarize themselves with some material before coming to the class either by reading the content or listening to video lectures [9], [10]. This frees up time in class so students can work together to solve theoretical or practical problems.

To increase attendance, short quizzes can be assigned during lectures. Assigning group quizzes can enhance student engagement by allowing students in the same group to discuss and learn from each other. Thus, taking a quiz is as much about

learning as about earning a grade in the class.

Often the homework can be frustrating if the students do not learn the “tricks.” Students find the lectures easy, but they find it harder to solve problems. Thus, some of the tricks to solving the problems need to be taught during lecture. This requires working out some of the problems that would have been assigned as homework. Another approach is to provide solutions to problems that are similar to the homework problems. Studying these solutions will be very helpful to the students in preparing them for their homework. The same is also true for programming problems. Starter codes for programming assignments should be provided to the students. This will help them in solving their programming assignments. Some students have strong theoretical skills but are less inclined to solve programming problems.

Finally, often there is a gap between the homework assignments and exams. Homework problems are often time consuming and require more calculations, whereas examinations cover short problems that take less time but are thought provoking and nontrivial. Students need to develop skills in solving problems that are similar to those in the tests. The aforementioned quizzes during lectures can be very helpful to students in preparing for exams.

An integrated approach to teaching

There is debate in the community about the interrelationship between innovation and education [11]. This section describes examples of how signal processing can be taught more effectively via an integrated approach that emphasizes learning of the theory, application, and intuition.

Mathematical derivations

Many decades ago, the entire class was spent on deriving the mathematical theory, and the homework problems were also mostly mathematical. Many DSP homework problems involve “tricks” that are not taught in the class, but students are expected to figure them out. As a student, I enjoyed figuring out these tricks; however, many students lose interest in learning DSP as they cannot work them out. Thus, there is a

Learning DSP is more important than ever before because it provides the foundation for machine learning and artificial intelligence.

need to spend lecture time solving problems where the tricks are explained. This reduces the time required for all of the derivations. Fortunately, students can read the textbook for this part. In general, the amount of lecture time used to derive theory needs to be reduced.

The threshold concepts come into play when explaining the tricks. Students may have forgotten some of the concepts. Typically, in a junior-level class on signals and systems, I spend a week teaching functions, scaling and shifting of functions, complex numbers, and trigonometric identities.

Another aspect of deriving theory is to first explain the results intuitively and then derive the theory. At other times, it may be easier to compute the result using MATLAB and then explain the result theoretically. This achieves two objectives: students develop practical skills, and they then relate the experimental result to theory. This makes the theory more relevant. As one example, I illustrate the fast Fourier transform (FFT) properties using Table 1. Students then verify the properties using MATLAB (see Problems 5 and 6 in the section “In-Class Group Activity”). Variations of Table 1 can also be used for homework or group activity.

Applications and visualization in MATLAB

I explain several applications of the theory during the lecture. In addition, I assign programming problems for applications as part of the homework. For example, when describing digital filters in terms of, say, low-pass, bandpass, and high-pass, I take a sound or audio file, filter it with different passbands, and then listen to the filtered sound. These sounds are either embedded into the PowerPoint presentation or obtained from MATLAB

during the class. I also provide the codes for the filtering operations. Students can use the code for solving their homework.

In the first DSP class, I ask students to record their speech as they describe themselves for a few minutes. They turn in their speech as part of the homework. Then, in subsequent weeks, I ask them to filter their own speech. We have also downloaded publicly available bird sounds from websites, and the students use these sounds for their homework.

In another application for speech or audio compression, we compute the FFT of the signal. Then we retain low-frequency content and compute the inverse FFT (IFFT). Then we listen to the sound. This is explored in a homework problem (Problem 1).

The MATLAB problem in Problem 1 relates to audio compression. In this problem, students explore the principles of audio compression where the high-frequency content is discarded. The MATLAB codes for the functions `fft_compress` and `fft_expand` are provided to the students.

Problem 1

- 1) Load the audio file, referred to as $x[n]$. Let $X[k]$ be its discrete Fourier transform (DFT). Compute $X[k]$ using the `fft` command.
- 2) Compress the FFT $X[k]$ using the `fft_compress` function and a percentage of compression = 10% (0.10). This retains only the first 10% of the spectrum.
- 3) Using the compressed sound file from step 2, apply the `fft_extract` function to reconstruct the original audio file. Save the reconstructed audio sound file and play it. Refer to this signal as $x_1[n]$. Comment on your observations.

- 4) Generate an error file which is the difference between the original audio file, $x[n]$, and the reconstructed audio file, $x_1[n]$. Call this error signal $e[n]$. Save the error sound file and play it. Comment on your observations.

Comment: $e[n]$ contains the higher frequency content of $x[n]$.

- 5) Observe that the error signal contains frequency components in the mid-band and no frequency components at low frequency. Shift left the frequency components of the error signal by k_0 samples and compute the IFFT of the shifted frequency-domain signal. Save the generated sound file and play it. Call this signal $x_2[n]$. Comment on your observations.

Comment:

$$X_2[k] = E[k + k_0]$$

$$x_2[n] = e[n] e^{-j\frac{2\pi}{N}nk_0}.$$

Table 1. The DFT properties.

$x[n]$	$X[k]$
(A, B, C, D, E, F)	(6a, 6f, 6e, 6d, 6c, 6b)
(A, F, E, D, C, B)	(6a, 6b, 6c, 6d, 6e, 6f)
(A*, B*, C*, D*, E*, F*)	(6a*, 6b*, 6c*, 6d*, 6e*, 6f*)
(A, F, E, D, C, B)	(6a, 6b, 6c, 6d, 6e, 6f)
(a*, b*, c*, d*, e*, f*)	(A*, F*, E*, D*, C*, B*)
(a, f, e, d, c, b)	(A, F, E, D, C, B)
(a*, f*, e*, d*, c*, b*)	(A*, B*, C*, D*, E*, F*)
(a, -b, c, -d, e, -f)	(D, E, F, A, B, C)
(a, -f, e, -d, c, -b)	(D, C, B, A, F, E)
(a, 0, b, 0, c, 0, d, 0, e, 0, f, 0)	(A, B, C, D, E, F, A, B, C, D, E, F)
(A, B, C, D, E, F, A, B, C, D, E, F)	(12a, 0, 12f, 0, 12e, 0, 12d, 0, 12c, 0, 12b, 0)
(A, 0, B, 0, C, 0, D, 0, E, 0, F, 0)	(6a, 6f, 6e, 6d, 6c, 6b, 6a, 6f, 6e, 6d, 6c, 6b)
(A/6, 0, F/6, 0, E/6, 0, D/6, 0, C/6, 0, B/6, 0)	(a, b, c, d, e, f, a, b, c, d, e, f)
(A/12, F/12, E/12, D/12, C/12, B/12)	(a, 0, b, 0, c, 0, d, 0, e, 0, f, 0)
(A/12, F/12, E/12, D/12, C/12, B/12)	
(D, 0, E, 0, F, 0, A, 0, B, 0, C, 0)	(6a, -6f, 6e, -6d, 6c, -6b, 6a, -6f, 6e, -6d, 6c, -6b)

Example given in class: Let the discrete Fourier transform (DFT) of a six-point complex sequence $\{a, b, c, d, e, f\}$ be another complex sequence $\{A, B, C, D, E, F\}$. Then complete the table below. The sequences in bold were given, and students were asked to find the corresponding pairs; the sequences in red correspond to solutions.

The signal $x_2[n]$ is complex and differs from $e[n]$ and is a modulated version of $e[n]$. Thus, if we listen to its magnitude, it will sound different from $e[n]$.

- 6) Multiply the signal $x_2[n]$ with a complex exponential ($e^{j(2\pi/N)k_0n}$), where k_0 corresponds to the shift in frequency performed in step 5. Save the generated sound file and play it. Call this signal $x_3[n]$. Comment on your observations.

Comment:

$$x_3[n] = x_2[n] e^{j\frac{2\pi}{N}nk_0} = e[n].$$

$x_3[n]$ is the same as $e[n]$.

Solution: Solutions with MATLAB codes are provided in the supplementary materials that appear with this article on IEEE Xplore. ■

I introduce practical applications while describing theoretical concepts. For example, while introducing the definition of *autocorrelation* of a real signal, I provide examples of *photoplethysmogram (PPG)* and respiration-rate signals. Then I discuss how to *compute the heart rate and respiration rate from these two signals using autocorrelation by looking at the zero-crossings*. We then compute the DFT of the signals and verify if the frequency obtained by the autocorrelation is the same as that from the DFT. This is illustrated in Problem 2, where the PPG signal is used to compute the heart rate. The respiration-rate signal is not included in Problem 2, but the approach is similar. This helps connect the theoretical expression for autocorrelation to a practical application.

Problem 2

The PPG signal captured from a sensor at a 100-Hz sampling frequency is provided in this problem. The length of the signal is 1,024 samples (10.24 s). The data file (ppg_100hz_1024samples.csv) for this problem is given in the supplementary

materials that appear with this article on IEEE Xplore. The data are part of a PPG signal (ppg_100hz_1024samples.csv). The PPG signal is used in this problem to compute the heart rate.

- 1) Compute the 1,024-point FFT of the signal and plot the absolute values of the single-sided FFT with a stem plot: Find the frequency in hertz of the highest magnitude in the FFT of the PPG. Note that the frequency corresponding to the highest magnitude represents the heart rate.
- 2) In this part, we compute the heart rate using autocorrelation: This is accomplished by finding the difference of the first and third zero-crossings, which corresponds to the time period of the signal. This information is used to compute the heart rate in hertz from the PPG.

Solution: The MATLAB code for this problem is available in the supplementary materials that appear with this article on IEEE Xplore. A diagram containing results from the two parts of the problem is presented in Figure 1. The heart rate can be estimated by:

- 1) Finding the highest peak from the DFT spectrum: The fundamental frequency is 1.0742 Hz for the PPG signal, which results in a heart rate of 64.45 beats per minute (bpm).
- 2) Considering the interval between first and third zero-crossings: A lag difference of 94 at a 100-Hz sampling rate = 0.94 s or 63.8 bpm. Note that the values from steps 1 and 2 are almost the same. ■

Fortunately, numerous large collections of data and signals are now publicly available. These data and signals can be used as part of the homework or class projects. In my DSP class, I have used intracranial electroencephalogram signals for seizure detection from the UPenn and Mayo Clinic's Seizure Detection Challenge on Kaggle [12]. Students use the same signals for solving different

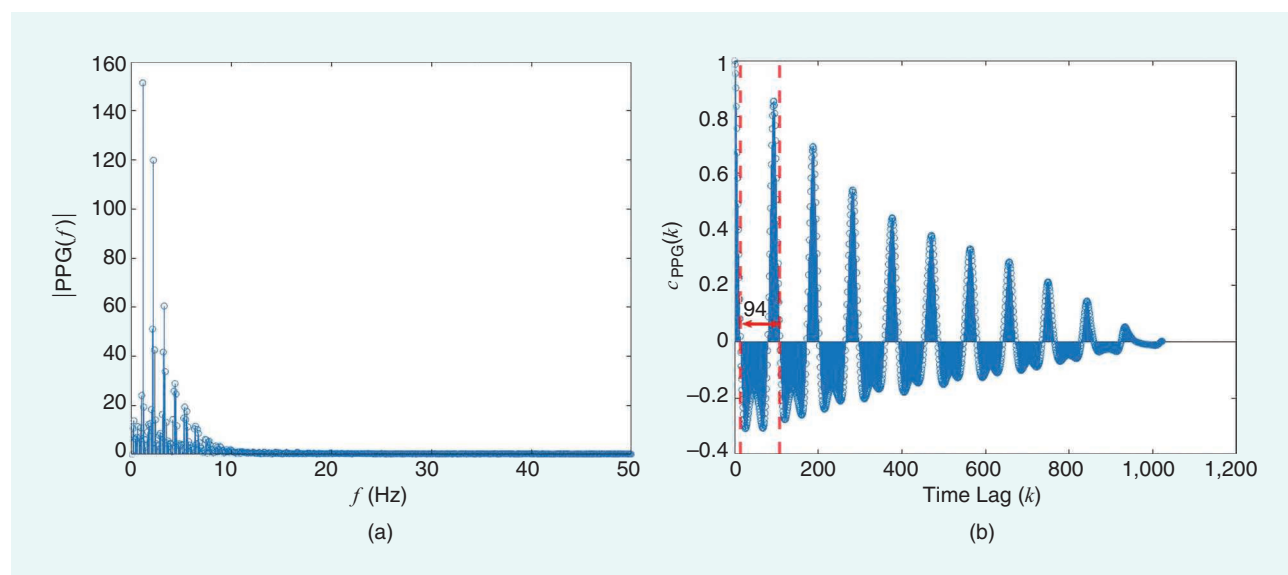


FIGURE 1. (a) The DFT and (b) normalized autocorrelation for the PPG signal.

programming problems assigned over many weeks and compute time-domain and frequency-domain features. These problems are described in “Programming Assignments Using Intracranial Electroencephalogram Data” (see Problems S1–S3).

The roles of theory and application sometimes can be interchanged. We first describe an application using MATLAB. Then we make an observation and then derive the theory.

Intuitive insight

It is important to explain the results from MATLAB experiments intuitively. For example, the effects of scaling and shifting a signal in the spectral domain can be quickly observed. The theory can then be explained. I included the following Problem 3 as part of a homework assignment.

Problem 3

Two signal processing systems are shown in Figures 2 and 3, where $x_1[n]$ and $x_2[n]$ are audio sounds, and $H(z)$ is a 100th-order finite-impulse response low-pass filter with cutoff frequency $\pi/2$. For each system, load the input audio sounds, use MATLAB to obtain the sounds $z[n]$, $y_1[n]$, and $y_2[n]$,

and listen to these sounds. Use the *freqz* command to plot the spectrum of the sounds $x_1[n]$, $x_2[n]$, $z[n]$, $y_1[n]$, and $y_2[n]$ for each system. Compare the output signals obtained using the two DSP systems.

Solution: Multiplication of $x_2[n]$ by $(-1)^n$ results in a shift in the frequency domain by π . Thus, the signal $z[n]$ contains the audible $x_1[n]$ along with the shifted version of $x_2[n]$, which is inaudible to the human ear. The second multiplication of $z[n]$ by $(-1)^n$ results again in a shift in the frequency domain by π , thus making it possible to listen to $x_2[n]$ at $y_2[n]$. However, $y_2[n]$ also contains the high-frequency content of $x_1[n]$. This can be avoided by band-limiting the input signals using system 2 shown in Figure 3. ■

The intrigue of the previous problem lies in the fact that the sound $z[n]$ does not seem to contain $x_2[n]$, whereas it is audible in $y_2[n]$. Students think that the signal $x_2[n]$ is lost, and they are surprised that it can be recovered. I then explain this mathematically and intuitively. This problem can illustrate the basic concepts of audio steganography.

Blended teaching and active learning

Almost all students today have their own laptops that they can bring to class. Thus, it is easy for them to learn in an active

Programming Assignments Using Intracranial Electroencephalogram Data

This is a description of programming assignments that use intracranial electroencephalogram (EEG) data from the Seizure Detection Challenge on Kaggle organized by the University of Pennsylvania and the Mayo Clinic [12]. The students were assigned one specific subject and a specific electrode/channel signal from that subject. The students were asked to extract various time- and frequency-domain features and comment on the suitability of these features for discriminating ictal (during seizure) and interictal (baseline) clips to detect seizures. The sample solutions for these problems, provided in the supplementary materials that appear with this article on IEEE Xplore, use the training data from channel 1 for the EEG clips from Dog 1, which has a total of 596 clips of which 178 are ictal, i.e., those corresponding to seizures. Each clip is a 1-s recording.

Problem S1. This problem explores time-domain signal processing. Extract and plot the eight time-domain features listed below for the assigned subjects. Use the *stem()* command to plot the value of each feature for all of the clips. Observe the plots and comment on whether the given feature could be used to detect seizures.

- 1) *Measures of central tendency:* arithmetic mean, median, and mode.
- 2) *Energy:* the energy for a sequence $x[n]$ of length N is given by

$$E = \sum_{n=1}^N |x[n]|^2.$$

- 3) *Total length of the curve (sum of distances between successive points):*

$$L = \sum_{n=2}^N |x[n] - x[n-1]|.$$

- 4) *Hjorth parameters:*

$$\text{Activity}(x(t)) = \text{var}\{x(t)\}$$

$$\text{Mobility}(x(t)) = \sqrt{\frac{\text{var}\left\{\frac{d}{dt}x(t)\right\}}{\text{var}\{x(t)\}}}$$

$$\text{Complexity}(x(t)) = \frac{\text{Mobility}\left(\frac{d}{dt}x(t)\right)}{\text{Mobility}(x(t))}.$$

Solution: The MATLAB codes for this problem are provided in the supplementary materials that appear with this article on IEEE Xplore. The results varied based on the assigned subject and electrode. In many cases, the energy, total length, and Hjorth activity were good indicators of seizure. ■

Problem S2. In this problem, we will generate an analytic signal from the EEG signal. The analytic signal is a complex signal whose real part is the signal itself and whose imaginary part is the signal filtered by the Hilbert transform filter. Using this, we generate two new features listed below. Plot the features using the *stem()* command

learning environment where they can write short programs or solve short problems during the class. Students can learn from each other by working in groups.

Flipped classes have been used to teach DSP effectively [13]. At UMN, I taught the undergraduate DSP class EE-4541 in an active learning classroom in the fall of 2013 [14], [15]. The students sat around tables in groups of three. The classroom was equipped with a camera for the instructor and there were TV screens near each table, as shown in Figure 4. I sometimes used pen and paper to derive or explain theoretical results. At other times, I used PowerPoint slides. Students had access to my PowerPoint slides a week before the class, and they were asked to review them before coming to class.

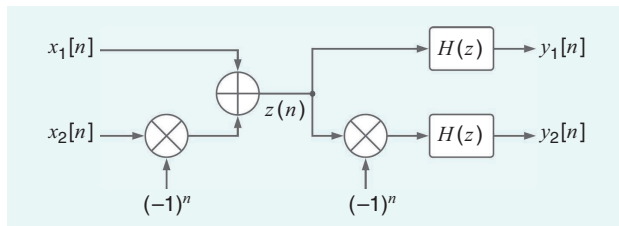


FIGURE 2. Signal processing system 1 for Problem 3.

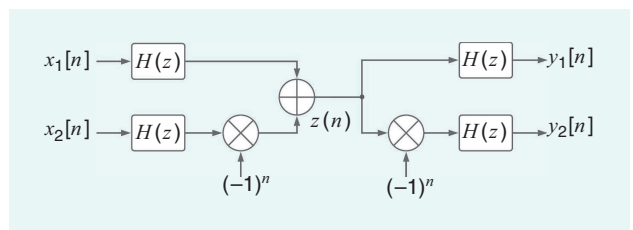


FIGURE 3. Signal processing system 2 for Problem 3.



FIGURE 4. An active learning classroom.

and comment on whether the given feature could be used to detect seizures.

- 1) *Mean instantaneous amplitude of delta band*: Filter the original EEG clips using a bandpass filter to obtain the signal in the delta band (1–4 Hz). Using the Hilbert transform, obtain the discrete-time analytic signal (complex valued), the magnitude of which provides the instantaneous amplitude. Use its average as the feature.
- 2) *Mean instantaneous frequency of alpha band*: Filter the original EEG clips using a bandpass filter to obtain the alpha-band (8–12 Hz) signal. Using the Hilbert transform, obtain the discrete-time analytic signal (complex valued), the angle of which provides the instantaneous phase. The derivative of the unwrapped instantaneous phase scaled by the sampling frequency yields the instantaneous frequency. Use its average as the feature.

Solution: The MATLAB codes for this problem are provided in the supplementary materials that appear with this article on IEEE Xplore. The results varied based on the assigned subject and electrode. In many cases, the mean instantaneous amplitude of the delta band is a good indicator of seizure. ■

Problem S3. We explore the following three methods listed to observe the power spectral density (PSD) of the

EEG clips and find out if PSD is a useful feature for seizure detection.

- 1) *Spectrogram*: Combine all the clips (ictal followed by interictal) to form a single time series. Use the spectrogram command with a window of 100 sample segments and an 80-sample overlap to view the frequency spectrum. Show the output as a surface plot with time on the x -axis, frequency on the y -axis and spectrum (in decibels) along the third axis.
- 2) *Welch PSD estimate*: Combine the ictal clips and interictal clips separately to form two different time series. Using the *pwelch* command, obtain and plot the PSD estimate of the two signals on the same graph for a normalized frequency range of $[0 - \pi]$. Use the *smoothdata()* function to smooth the plot and identify the normalized frequency range that has the maximum difference between the PSD estimates of the ictal and interictal series.
- 3) *Average PSD as a feature*: Using the *stem()* command, plot the average of the PSD estimate obtained using the *pwelch* command for each clip. Observe and comment on whether this feature could be used to detect seizures.

Solution: The MATLAB codes for this problem are provided in the supplementary materials that appear with this article on IEEE Xplore. The results varied based on the assigned subject and electrode. In many cases, the PSD estimate is a good indicator of seizure. ■

I assigned a quiz in the first class. The students were divided into three groups based on their performance in the quiz: top, middle, and bottom. Groups consisting of three students each were created by randomly picking one student from each of the top, middle, and bottom groups. This course was taught twice per week, where each class was of 75-min duration. Out of the 75 min, the last 15 min were reserved for either a group activity or a group quiz. During the group activity, the students were assigned short problems and short MATLAB assignments to work on in their groups. In this approach, a student who needed help could learn from another student in the group. The group quizzes also consisted of short problems and short MATLAB programming problems. The group quiz and group activity alternated from one class to another during the semester. At the end of the group activity, I was able to provide intuitive insights and solutions to the problems at the end of the lecture.

In-class group activity

I designed the group activity problems such that students could first either learn the tricks needed to solve problems or compute the final result by MATLAB before the theory was presented. Other problems were designed to use MATLAB to verify what was learned from theory. Some examples of group activity are described next.

Problem 4

Consider the following sinc function:

$$x[n] = \left(\frac{\sin \frac{n\pi}{4}}{n\pi} \right).$$

Using MATLAB, plot the discrete-time Fourier transform (DTFT) of the 10 signals listed below.

- 1) Plot DTFT $X(e^{j\omega})$ using MATLAB.
- 2) Let $x_1[n] = x[n - 10]$. Plot $X_1(e^{j\omega})$.
- 3) Let $x_2[n] = x[-n]$. Plot $X_2(e^{j\omega})$.
- 4) Let $x_3[n] = nx[n]$. Plot $X_3(e^{j\omega})$.
- 5) Let $x_4[n] = e^{j(n\pi/6)}x[n]$. Plot $X_4(e^{j\omega})$.
- 6) Let $x_5[n] = (-1)^n x[n]$. Plot $X_5(e^{j\omega})$.
- 7) Let $x_6[n] = x[n] * x[n]$. Plot $X_6(e^{j\omega})$.
- 8) Let $x_7[n] = x^2[n]$. Plot $X_7(e^{j\omega})$.
- 9) Let $x_8[n] = x[2n]$. Plot $X_8(e^{j\omega})$.
- 10) Let $x_9[n] = \begin{cases} x[n/2], & n \text{ is even} \\ 0, & n \text{ is odd} \end{cases}$.

Solution: The MATLAB codes to this problem are provided in the supplementary materials that appear with this article on IEEE Xplore. The students were asked to explain the discrepancy between impulse magnitudes in the MATLAB result and the theoretical result for step 4 ($201/2\pi$ versus 1). ■

As another example of group activity, I will ask the students to observe the properties of the DFT by solving Problem 5 using MATLAB. Then they solve Problem 6 based on their observations from Problem 5. Once the students understand the properties, I derive some of them theoretically in class. Finally, students explore the use of the DFT properties

shown in Table 1 as part of their homework. In this table, the sequences in red correspond to the solutions and are assigned as homework.

Problem 5

Evaluate the following using MATLAB:

- 1) FFT [2, 3, 4, 5, 6]
- 2) FFT [FFT [2, 3, 4, 5, 6]]
- 3) FFT [2, 3, 4, 5, 6, 0, 0, 0, 0, 0]
- 4) FFT [2, 3, 4, 5, 6, 2, 3, 4, 5, 6]
- 5) FFT [2, 2, 3, 3, 4, 4, 5, 5, 6, 6]
- 6) FFT [4, 5, 6, 2, 3]
- 7) FFT [2, -3, 4, -5, 6].

Solution: The FFT of a sequence can be computed using the `fft` command in MATLAB. ■

Problem 6

Let $(a, b, c, d, e) \Leftrightarrow (A, B, C, D, E)$. Write general expressions for the FFTs listed below in terms of (A, B, C, D, E) based on the observations from Problem 5.

- 1) FFT [FFT [a, b, c, d, e]]
Solution: $[5a, 5e, 5d, 5c, 5b]$. ■
- 2) FFT [a, b, c, d, e, 0, 0, 0, 0, 0]
Solution: $[A, *, B, *, C, *, D, *, E, *]$. Here $*$ denotes interpolated values and hence cannot be expressed in terms of A, B, C, D, E . ■
- 3) FFT [a, a, b, b, c, c, d, d, e, e]
Solution: $[A, B, C, D, E, A, B, C, D, E][1 + e^{-j(k\pi/5)}]$ for $k = 0, 1, \dots, 9$. ■
- 4) FFT [a, b, c, d, e, a, b, c, d, e]
Solution: $[2A, 0, 2B, 0, 2C, 0, 2D, 0, 2E, 0]$. ■
- 5) FFT [d, e, a, b, c]
Solution: $[A, Be^{-j(4\pi/5)}, Ce^{-j(8\pi/5)}, De^{-j(12\pi/5)}, Ee^{-j(16\pi/5)}]$ using the property $x([n - 2])_{<5>} \leftrightarrow e^{-j(4k\pi/5)}X[k]$. ■
- 6) FFT [a, -b, c, -d, e]
Solution: Interpolation with two-and-a-half-sample delay. Here the input can be expressed as $(-1)^n x[n] = x[n]e^{-j(2\pi n/5)(5/2)}$. ■
- 7) FFT [a, e, d, c, b]
Solution: $[A^*, B^*, C^*, D^*, E^*]$, using the property $x([-n])_{<5>} \leftrightarrow X^*[k]$. ■

In-class group quiz

Students take a group quiz lasting 15 min once a week. This engages the students in the group to solve the problems together. It also reduces the pressure of taking quizzes for individual students. An example of a group quiz is given in Problem 7. Group quizzes help the students to prepare for the examinations.

Problem 7

Evaluate the following. Note that these time-domain convolution problems are easier to solve in the frequency domain.

- 1) $\frac{\sin(\frac{n\pi}{4})}{n\pi} * \frac{\sin(\frac{n\pi}{8})}{n\pi}$.

Solution: The frequency-domain representation of a sinc signal is a rectangular function. The solution then involves multiplying two rectangular functions and then taking an inverse Fourier transform, which is another sinc function. [See Figure 5(a).] ■

$$2) \frac{\sin\left(\frac{n\pi}{4}\right)}{n\pi} * \left(\frac{\sin\left(\frac{n\pi}{2}\right)}{n\pi} - \frac{\sin\left(\frac{n\pi}{3}\right)}{n\pi} \right).$$

Solution: See Figure 5(b). ■

$$3) \frac{\sin\left(\frac{n\pi}{4}\right)}{n\pi} * \left(\delta[n] - \frac{\sin\left(\frac{n\pi}{8}\right)}{n\pi} \right).$$

Solution: The same as before; we take advantage of the fact that the Fourier transform of the δ function is 1. See Figure 5(c). ■

$$4) \left(\delta[n] - \frac{\sin\left(\frac{n\pi}{4}\right)}{n\pi} \right) * \left((-1)^n \frac{\sin\left(\frac{n\pi}{4}\right)}{n\pi} \right).$$

Solution: Note, $e^{-jn\pi} = (-1)^n$. Thus, the signal is shifted in the frequency domain by π . See Figure 5(d). ■

$$5) \left((-1)^n \frac{\sin\left(\frac{2n\pi}{3}\right)}{n\pi} \right) * \sin\left(\frac{n\pi}{4}\right).$$

Solution: See Figure 5(e). ■

Evaluation metrics

Group comparison

Data were collected from EE-4541 students at UMN in Fall 2012 (41 respondents out of 77 enrolled) and Fall 2013 (62 respondents out of 87 enrolled). The data collection was approved by the Institutional Review Board (IRB) at UMN under the “Exempt” category. The metrics for Fall 2012 serve as the baseline for the comparison.

The two groups of students did not differ significantly on any available demographic variables, including undergraduate–graduate status, year in the university, ethnicity, sex, age, cumulative grade point average, and composite ACT score. We can conclude that, as far as can be determined from the available data, the students in the two sections of EE-4541 can be validly compared to one another.

Outcome analyses

The metrics used to understand the efficacy include *engagement*, *enrichment*, *flexibility*, *effective use*, *classroom/course fit*, *confidence*, and *student learning outcome (SLO)*. For each measure, a number of criteria and questions were chosen for the students, and they were asked to grade each question as Strongly Agree, Agree, Disagree, or Strongly Disagree, corresponding to numerical scores of 3, 2, 1, and 0, respectively. A brief description of each of the metrics is presented next.

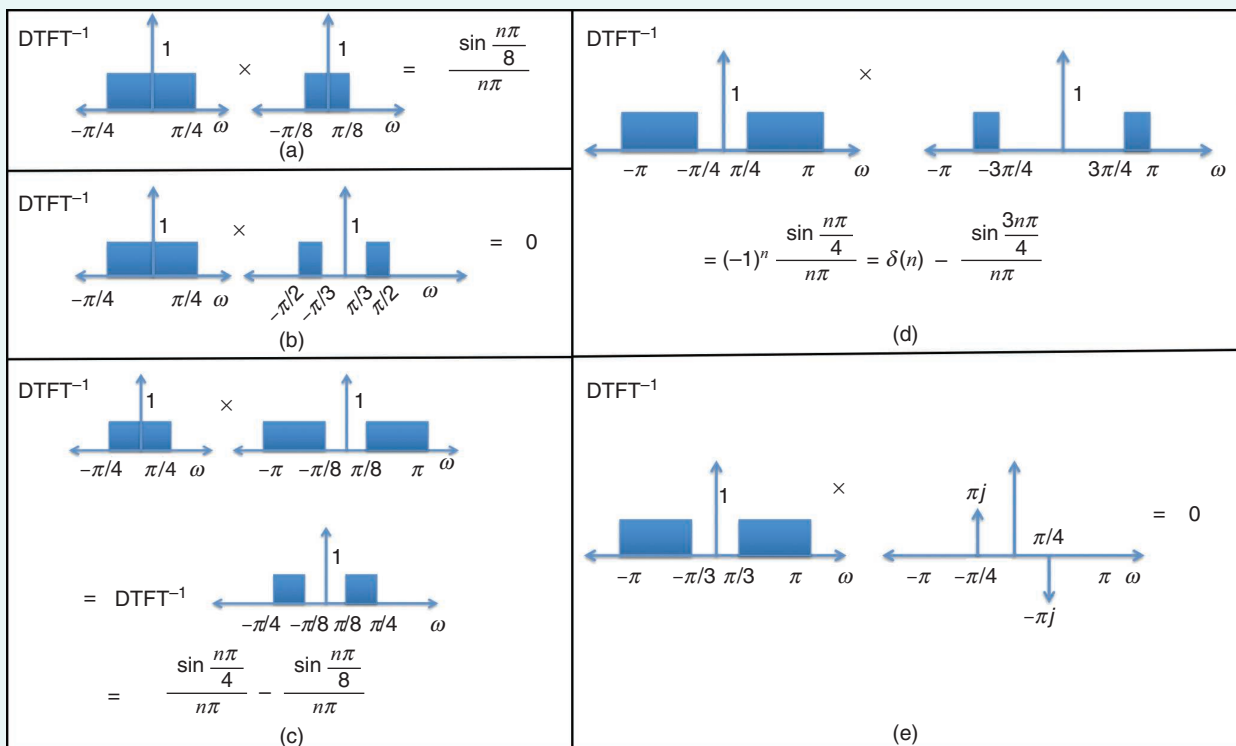


FIGURE 5. The Problem 7 solutions.

1) *Engagement:*

- Encourages my active participation
- Promotes discussion
- Helps me develop connections with my classmates
- Helps me develop connections with my instructor
- Engages me in the learning process.

2) *Enrichment:*

- Enriches my learning experience
- Makes me want to attend class regularly
- Increases my excitement to learn.

Flexibility:

- Facilitates multiple types of learning activities
- Nurtures a variety of learning styles.

Effective use:

- The instructor is effective in using the technology available in the classroom for instructional purposes.
- The instructor is effective in using the classroom for instructional purposes.

Classroom/course fit:

The criteria used for the classroom/course fit metric are listed below:

- The classroom is an appropriate space in which to hold this particular course.
- The in-class exercises for this course are enhanced by the features of this classroom.

Confidence:

The students rate the course based on the following criteria:

- Course helps develop confidence in working in small groups
- Helps students develop confidence in analyzing
- Helps student develop confidence in presenting
- Helps develop confidence in writing
- Improves confidence that the student can speak clearly and effectively.

3) *SLO:*

- Helps me develop professional skills that can be transferred to the real world

- Helps me to define issues or challenges and identify possible solutions
- Prepares me to implement a solution to an issue or challenge
- Helps me to examine how others gather and interpret data and assess the soundness of their conclusions
- Deepens my understanding of a specific field of study
- Assists me in understanding someone else's views by imagining how an issue looks from his or her perspective
- Helps me to grow comfortable working with people from other cultures
- Improves my confidence that I can speak clearly and effectively
- Encourages me to create or generate new ideas, products, or ways of understanding
- Prompts me to incorporate ideas or concepts from different courses when completing assignments
- Enabled the instructor to make intentional connections between theory and practice in this course.

Bivariate tests

Independent-samples t-tests were conducted to compare the learning metrics of the students taught in the Fall 2013 semester using partial flipping and active learning versus those in the Fall 2012 class with traditional learning. The results are summarized in Table 2. On all aggregated variables derived from student responses, statistically significant differences (at the $p < 0.05$ level or better) were found between the mean scores of the two groups, favoring the Fall 2013 class (see Table 2). The group-level difference was the highest in the categories of engagement, flexibility, and confidence. The next highest categories include classroom/course fit and SLO. There is still room to improve the scores in the enrichment and effective use categories.

Current trends

While MATLAB is used in many universities and industries, it is not an open source environment. There is great interest in teaching DSP using Python or Octave as they do not require licenses. However, students have to write code from scratch for many DSP functions, unlike in MATLAB, where students can use numerous in-built functions. Nevertheless, using Python is more desirable as most open source libraries for machine learning functions are written in Python. There is also growing interest in teaching DSP for embedded systems such as smartphones so that students can design apps for cell phones [16]. For example, they can write DSP programs in Python for smartphones to analyze biomedical signals such as electrocardiograms. Most commercial products like smartwatches already have this capability. We should create DSP lab courses to teach app design for either Android or iOS operating systems to prepare students for the rapidly changing job environment.

The entire world was disrupted by COVID-19 during the initial white paper submission of this article (2020

Table 2. Student evaluation metrics.

Variable	Semester	N	Mean Score	p
Engage	Fall 2012	41	2.345	0
	Fall 2013	62	2.960	0.019
Enrich	Fall 2012	41	2.565	
	Fall 2013	62	2.854	0
Flexibility	Fall 2012	41	2.329	
	Fall 2013	62	3.169	0.028
Effective	Fall 2012	41	2.793	
	Fall 2013	62	3.129	0.004
Fit	Fall 2012	41	2.598	
	Fall 2013	62	3.024	0.001
Confidence	Fall 2012	41	2.327	
	Fall 2013	62	2.672	0.007
SLO	Fall 2012	41	2.497	
	Fall 2013	62	2.768	

February) and submission of the full paper (submitted June 2020 and revised 2020 December). Almost all classes in the second half of the Spring and Fall semesters of 2020 were taught using remote learning. This provided a challenge and opportunity to redesign various courses. Many laboratories were redesigned so that the students could perform the experiments at home. I provided lecture notes and recorded videos of the EE-4541 class from Fall 2017 [10] to my students in the Fall 2020 semester. All programming problems discussed in this article were assigned as homework in the Fall 2020 class. Active learning is still possible using breakout rooms in a remote learning environment such as Zoom; however, it is better suited for an in-person class.

Conclusions

We argue that DSP can be taught effectively by using visualization, active learning, and partial flipping. Application examples enable visualization, where instructors can play different sounds and illustrate plots of time-domain and spectral-domain features. This will increase student engagement, interest in the class, and understanding of the subject. Use of speech, audio, and biomedical signals in the class and as part of the homework can connect the theory to applications and better prepare students for jobs in industry. Future DSP textbooks should include application examples and connect the theory to applications. However, instructors can use the application examples presented in this article to supplement the textbook.

Acknowledgment

The partial-flipping experiment was carried out when the author participated in the 2012–2013 Office of Information Technology (OIT) Faculty Fellowship Program at UMN. The EE-4541 class was taught in the Fall 2012 semester without flipping (baseline) and in Fall 2013 with partial flipping in an active learning classroom. The author is most grateful to Kim Wilcox and Lauren Marsh from UMN OIT, who coordinated the program. The data collection for the OIT Faculty Fellowship Program was approved by the UMN IRB (principal investigators: D. Christopher Brooks, Kim Wilcox). The statistical analysis of the metrics was carried out by J.D. Walker. The evaluation reported in the section “Evaluation Metrics” is his contribution to this article. All faculty fellows met multiple times during the two-year program. The faculty group discussed many aspects of teaching, ranging from use of apps to applying concepts of game theory [17] and crowdsourcing in teaching. The author is grateful to Bhaskar Sen for his help in preparing this article. Zisheng Zhang and Sai Sanjay Balaji were teaching assistants for the class and developed the MATLAB codes. The author thanks Prof. M. Sabarimalai Manikandan of the Indian Institute of Technology, Bhubaneswar, India, for providing the PPG application example and the PPG signal. This article has supplementary downloadable material available at 10.1109/MSP.2021.3052487, provided by the author.

Author

Keshab K. Parhi (parhi@umn.edu) received his Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1988. He has been with the University of Minnesota, Minneapolis, Minnesota, 55455, USA, since 1988, where he is currently the Distinguished McKnight University Professor and Edgar F. Johnson Professor of electronic communication in the Department of Electrical and Computer Engineering. He has published more than 650 papers, is the inventor of 31 patents, and has authored the textbook *VLSI Digital Signal Processing Systems* (Wiley, 1999). He served as the editor-in-chief of *IEEE Transactions on Circuits and Systems, Part I* during 2004 and 2005. He is a Fellow of IEEE and a fellow of the American Association for the Advancement of Science, the Association for Computing Machinery, and the National Academy of Inventors.

References

- [1] A. V. Oppenheim, “A personal view of education,” *IEEE Signal Process. Mag.*, vol. 9, no. 2, pp. 69–72, 1992. doi: 10.1109/79.127285.
- [2] A. V. Oppenheim, “One plus one could equal three (and other favorite clichés),” *IEEE Signal Process. Mag.*, vol. 23, no. 6, pp. 10–12, 2006. doi: 10.1109/SP-M.2006.248707.
- [3] L. S. Shulman, “Signature pedagogies in the professions,” *Daedalus*, vol. 134, no. 3, pp. 52–59, 2005. doi: 10.1162/0011526054622015.
- [4] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 1st ed., Englewood Cliffs, NJ: Prentice Hall, 1989.
- [5] J. Meyer and R. Land, *Threshold Concepts and Troublesome Knowledge: Linkages to Ways of Thinking and Practising Within the Disciplines*. Edinburgh: Univ. of Edinburgh, 2003.
- [6] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*, 1st ed., Englewood Cliffs, NJ: Prentice Hall, 1975.
- [7] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 4th ed., New York: McGraw-Hill, 2010.
- [8] S. Kanna, W. von Rosenberg, V. Goverdovsky, A. Constantinides, and D. P. Mandic, “Bringing wearable sensors into the classroom: A participatory approach [SP education],” *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 110–130, 2018. doi: 10.1109/MSP.2018.2806418.
- [9] T. A. Baran, R. G. Baraniuk, A. V. Oppenheim, P. Prandoni, and M. Vetterli, “MOOC adventures in signal processing: Bringing DSP to the era of massive open online courses,” *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 62–83, 2016. doi: 10.1109/MSP.2016.2556004.
- [10] K. K. Parhi. UMN EE-4541 Digital Signal Processing (Fall 2017). Accessed: Dec. 5, 2020. [Online Video]. Available: <https://www.youtube.com/playlist?list=PLT1QA48lhQLPwL4Y-XTzKeQBV1AYKv>
- [11] V. Solo, M. S. Greco, D. Mandic, P. Djuric, A. Spanias, and M. F. Bugallo, “Innovation starts with education: ICASSP 2019 education panel [SP forum],” *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 135–147, 2019. doi: 10.1109/MSP.2019.2923280.
- [12] “UPenn and Mayo clinic’s seizure detection challenge.” Kaggle. <https://www.kaggle.com/c/seizure-detection> (accessed Sept. 12, 2020).
- [13] B. Van Veen, “Flipping signal-processing instruction [SP education],” *IEEE Signal Process. Mag.*, vol. 30, no. 6, pp. 145–150, 2013. doi: 10.1109/MSP.2013.2276650.
- [14] K. K. Parhi. “Teaching digital signal processing by partial flipping.” *Cultivating Change Series*. <https://cultivatingchangeseries.com/ffp/teaching-digital-signal-processing-by-partial-flipping/> (accessed Feb. 1, 2020).
- [15] K. K. Parhi. Teaching Digital Signal Processing using Blended Teaching. (Jan. 29, 2014) Accessed Feb. 1, 2020. [Online Video]. Available: <https://youtu.be/gwrlSY4OVe8>
- [16] R. W. Schaffer. “EE 264 digital signal processing.” Stanford. <https://web.stanford.edu/class/ee264/> (accessed Feb. 1, 2020).
- [17] R. Epper, A. Derryberry, and S. Jackson. “Game-based learning: Developing an institutional strategy <https://library.educause.edu/resources/2012/8/gamebased-learning-developing-an-institutional-strategy> (accessed: June 14, 2020).