Module 4: Spectrum and Spectrogram of Signals

The main goal of this module is to learn how to analyze the frequency components of sampled audio signals in MATLAB. First, you will learn that sampled signals can be represented in the so-called frequency domain. Second, you will see that one can simultaneously analyze the time-frequency properties of signals using a so-called spectrogram. Both of these tools will be useful in designing and understanding wireless communication systems. ***Remember: Whenever you are stuck, have questions, or are interested in learning more details about a specific aspect, please feel free to ask us—we are here to help!!***

---

6 Frequency-Domain Representation of Signals

So far, you have learned how to sample signals and how to visualize such signals as a function of time, which is called the *time-domain representation* of signals. In this module, you will learn that one can use MATLAB to represent signals also in the frequency domain. The frequency-domain representation reveals which frequencies are present and how strong they are in a given signal. As you will see, communication systems transmit information over a given frequency (or a frequency band, which is a set of neighboring frequencies). We will illustrate this important concept by analyzing the spectrum of a range of audio signals. The *frequency-domain representation* is another way of looking at signals, in which they are viewed as a function of frequency (and not of time). In addition to this, you will also learn that it is often useful to visualize both the time and frequency domains of the same signal; this can be accomplished by visualizing so-called spectrograms that show which frequencies are present in a signal at what time instant. You have probably seen such spectrograms before as they are very similar to these well-known "frequency analyzers" in 80s or 90s boomboxes—one of the goals is to understand what they actually show.

6.1 The Spectrum of a Signal

Let us first inspect the spectrum of a very simple signal: a sine function at 440 Hz. Instead of using MATLAB functions to generate such a sine wave, we can also load one from the `examples` folder. Use the following command

```
[y,FS] = load_audio('examples/sine-440Hz.wav');
```

to load a 5 second 440 Hz sine wave that was sampled at $f_s = 44,100$ Hz. Now, as in the previous module, plot the time-domain representation of this signal. Simply use

```
plot_signal(y,FS)
```

to look at the sample waveform in the time domain—you may have to zoom in to see the actual sine wave. To listen to this waveform, type

```
play_audio(y,FS,OutID);
```

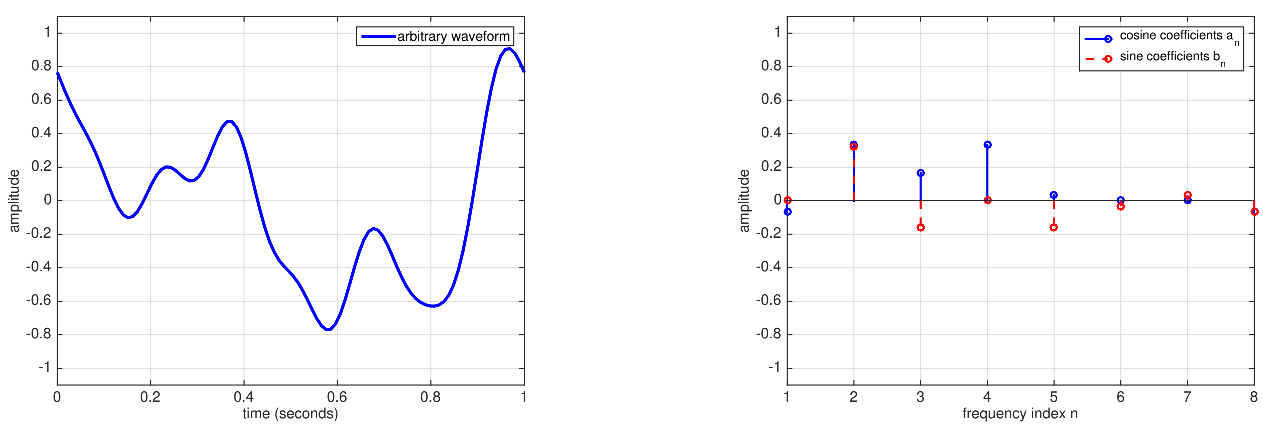
In case you forgot the output ID of your sound card, re-run the command `get_audio_info`. For this specific signal, we know that it only contains a single frequency at 440 Hz. It is interesting to note that every possible signal can be represented as a superposition of one or many sine and cosine waves at different frequencies.

Assume for a moment that you have a continuous signal that is represented as a function $f(t)$ where the time t is in the interval from 0 to 1 second. One of the central results in signal processing is that every possible such function can be represented as an infinite series (or sum) of sine and cosine waves at different frequencies:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(2\pi n t) + b_n \sin(2\pi n t) \right). \tag{4}$$

Here, the so-called Fourier coefficients (named after Jean-Baptiste Joseph Fourier, the inventor of this important result) a_n and b_n with $n \in \{0, 1, 2, \ldots\}$ characterize which frequency is represented how strong in the function $f(t)$. If you are not familiar with infinite series or the summation formula $\sum$, do not worry—it is not important to understand the mathematics behind this (you would learn this anyway in one of the first years while studying Electrical and Computer Engineering). If you are interested, feel free to ask us how the coefficients a_n and b_n can be calculated.

What is more important is to understand the intuition behind all this. Figure 8 provides an example. On the left figure, you can see a time-domain signal (think about it as an arbitrary function $f(t)$ for $t \in [0, 1]$). On the right figure, we plot the amplitude of the coefficients a_n and b_n for $n = 1, 2, \ldots, 8$. This figure shows which frequencies are present and how strong they are. You can see, for example, that at frequency index $n = 2$ we have $a_n = b_n = 1/3$. This means that the frequency with index $n = 2$ contains both a sine and cosine component which both have an amplitude of $1/3$. While the concept of frequency index is a bit abstract (and we will not go into more details here), one can also visualize the frequency components directly in Hz but then we need to know with which frequency we sampled the signal of interest. Hence, we will do that on our 440 Hz sine waveform that we stored in the variable `y`.

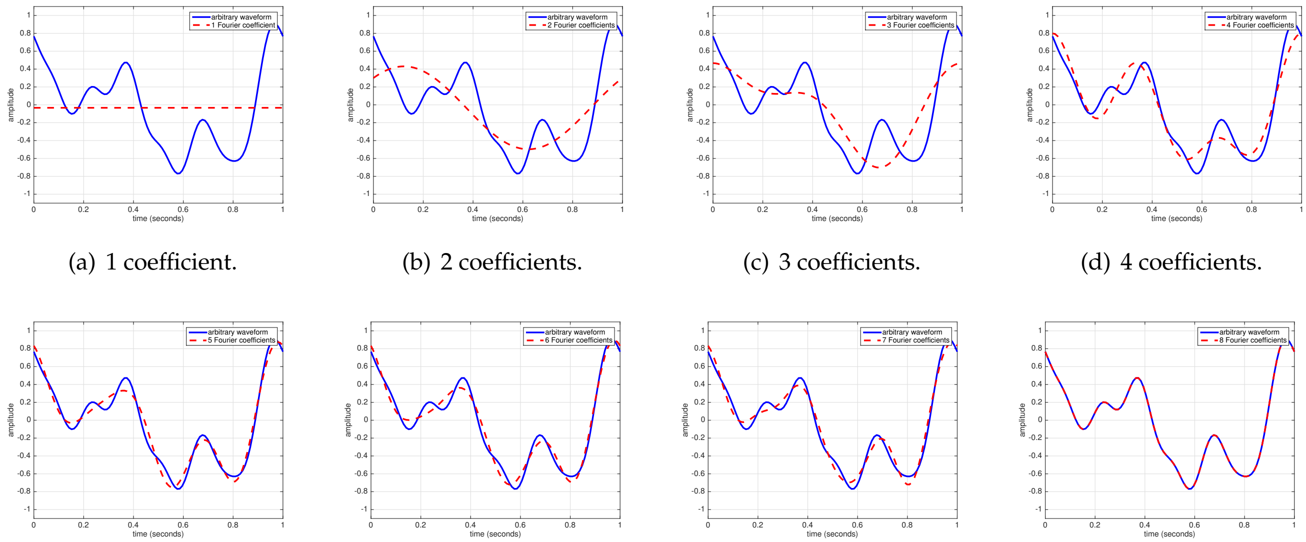


(a) Time-domain signal.

(b) Spectrum of time-domain signal.

Figure 8: Example of a time-domain signal and its frequency spectrum. The frequency spectrum shows which frequencies are present in a time-domain signal and how strong these frequencies are.

To gain some more intuition, it is interesting to show the same signal as above but by not using all eight frequencies. Figure 9 shows what happens if we sum only one frequency, two frequencies, three frequencies, etc. You can see that by superpositioning more and more sinusoids, we can approximate the function of interest. Mathematically speaking, we are taking the Fourier series in Equation (4) and do not sum to infinity but rather sum to 1, and then to 2, and then to 3, etc. Since the signal in Figure 8 contained only 8 different frequencies, we can perfectly approximate this signal with a superposition of 8 sinusoids; this is what is shown in the last figure of Figure 9.



(a) 1 coefficient. (b) 2 coefficients. (c) 3 coefficients. (d) 4 coefficients.

(e) 5 coefficients. (f) 6 coefficients. (g) 7 coefficients. (h) 8 coefficients.

Figure 9: Adding more and more sinusoids at different frequencies can approximate any function.

While the mathematics behind all this can be quite complicated, we can use MATLAB to do the hard work for us. Run the following MATLAB command:

```
plot_spectrum(y,FS);
```

You should see a new plot that shows the frequency in Hz on the x-axis and the magnitude on the y-axis; this is the frequency-domain representation of the signal stored in the vector `y`. As expected, only one frequency is present: 440 Hz. You can zoom in to confirm this fact. Note that all the other frequencies are very close to zero. This spectrum plot implies that the signal in the frequency domain consists almost exclusively of a single frequency at 440 Hz. Furthermore, you can see that the range of frequencies goes from 0 Hz to 22,050 Hz, which is the highest frequency you can represent when sampling a signal at $f_s = 44,100$ Hz. Remember that such plots show the *spectrum* of a sampled signal.[1]

Activity 14: Visualize the spectrum of other signals

The folder `examples` contains other signals. Plot the spectrum of the wav-file `sine-880Hz.wav`. Can you see which frequencies are present in that signal? Also try the signal `sine-1760Hz.wav`. Note that it is always a good idea to listen to the waveform as well, to develop intuition.

Also plot the spectrum of the wav-file `noise-white.wav`. Then, try `noise-pink.wav` and `noise-brown.wav`. What do you observe? Do these signals all contain the same frequencies?

[1] In case you are wondering how such plots are generated, the main technique is called the fast Fourier transform (FFT). This transform takes in any sampled signal and computes its spectrum. The FFT is probably the most widely used algorithm on this planet. Every cell-phone and laptop computes thousands of FFTs per second when transmitting data wirelessly!

6.2 The Spectrogram of a Signal

While analyzing the spectrum of signals is interesting and extremely important in the design of wireless communication systems, it has one key limitation. To illustrate this limitation, load the following waveform

```
[y,FS] = load_audio('examples/chirp-100Hz-to-10000Hz.wav');
```

and listen to it. This wav-file contains a sinusoid that continuously changes its frequency from 100 Hz to 10,000 Hz over time, starting with the low frequency and progressing to higher frequencies over 5 seconds. Such signals are called "chirps" and can be used to measure communication channels (or measure the acoustics of a given room). Now, plot the spectrum of this signal by typing:

```
plot_spectrum(y,FS);
```

As you will see, this audio signal contains frequencies from about 100 Hz to 10,000 Hz as the filename suggests. While the spectrum shows that these frequencies are present during the 5 second duration of the signal, it does *not* show that the frequencies actually change over time. Put simply, the spectrum tells you which frequencies are present but does not tell you anything about its evolution over time. Hence, it would be great to have a tool that visualizes both frequency and time—the *spectrogram* does exactly this!

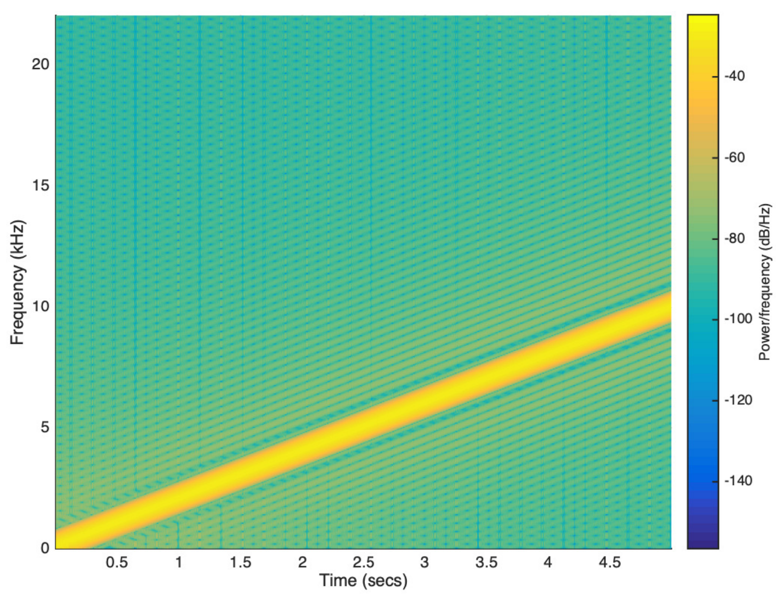


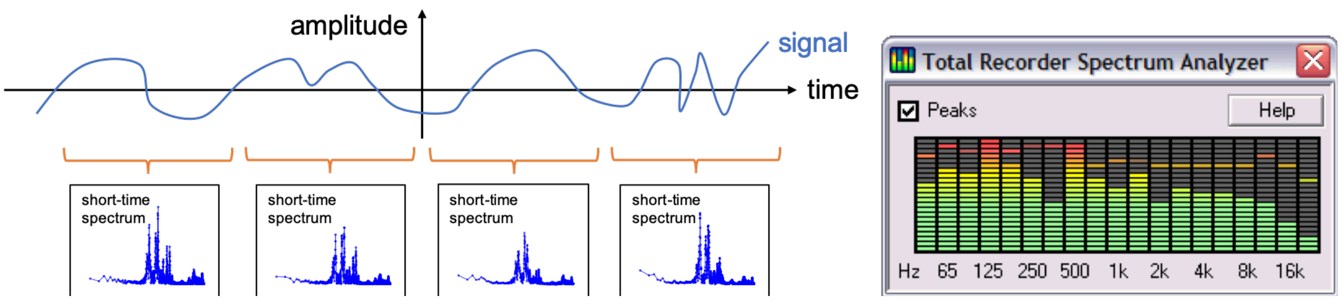
Figure 10: Example spectrogram of a 5 second chirp signal that sweeps a single sine wave from 100 Hz to 10,000 Hz. Spectrograms show what frequency components are present at what time instant.

Let us first look at a spectrogram before we explain how they are generated. In MATLAB, simply execute the following command (which was provided in the zip-file you downloaded)

```
plot_spectrogram(y,FS);
```

to plot the spectrogram of the chirp signal that is stored in the variable y. You should see a figure that looks like the one in Figure 10. The x-axis shows the time in seconds; the y-axis shows the frequency components ranging from 0 Hz to $f_s/2$ Hz. The colors encode the power (or intensity) measured in something that is called decibels (dB, for short). (You can ask one of us if you are interested in learning more about decibels, but basically it refers to the signal's magnitudes measured on a logarithmic scale.) Brighter colors indicate that certain frequencies are more strongly represented; darker colors imply certain frequencies are absent; think of these as the magnitudes of the the Fourier coefficients a_n and b_n. As you can see, there is a single frequency that starts at around 100 Hz and linearly increases towards 10,000 Hz. In words, the spectrogram shows at what time which frequencies are contained in the signal.

Figure 11 illustrates the principle of spectrogram computation. The left side shows that one takes the entire sampled signal and divides it into smaller blocks. For each smaller block, one computes the spectrum (this is called a short-time spectrum). One then generates a two-dimensional figure (as in Figure 10) where the columns correspond to a spectrum per block. The right side shows a typical spectrum analyzer computer plug-in. These plug-ins (inspired by 80s and 90s boomboxes) simply visualize the frequency spectrum for a small set of frequencies for separate blocks of the audio signal. Most of the time, the visualization is in sync with the audio signal, which allows one to see which frequencies are present at what time. The spectrogram as in Figure 10 is a static version of exactly the same idea.



(a) Principle of spectrogram calculation. (b) Typical spectrum analyzer.

Figure 11: Basics of spectrogram calculation. The signal is divided into small blocks for which the spectrum is calculated. Spectrum analyzer plug-ins (frequency equalizers) visualize the spectrogram over time.

Activity 15: Visualize the spectrogram of other signals

The folder `examples` contains a wide range of signals, including music and test sounds. Plot the spectrogram of these files and see whether you can observe in the spectrogram what you hear.

Activity 16: What is the highest frequency you can hear?

Visualize the spectrogram of the signal in the file `chirp-100Hz-to-22050Hz.wav`, which contains a chirp that goes to very high frequencies. Can you hear all frequencies up to 22,050 Hz? At what frequency can you not hear the signal anymore? Design an experiment to identify the frequency that you cannot hear anymore. To this end, synthetically generate different sine waves at different frequencies and play them back—you have learned in previous modules how to do that! At one point you should not be able to hear them anymore... What is that frequency? ***Important: Please be very cautious when playing test signals at high frequencies. Do not turn up the volume if you cannot hear a signal anymore! Note that even if you cannot hear signals at high frequencies, they are still present and reaching your ears. Also, maybe some other people in the room can hear them—also, dogs can hear them (that is how dog-whistles work).***

6.3 Why is the Spectrum of a Signal Important?

The concept of frequencies and spectrum is central in the design of wireless communication systems. Different communication systems typically operate at different frequencies. For example, FM (short for frequency-modulation) radio uses frequencies in the electromagnetic spectrum from about 88 MHz (mega-Hertz, which stands for $1,000,000$ Hz) to 108 MHz. Wireless LAN (also known as Wi-Fi or WLan) typically operates at frequencies of about 2.4 GHz (giga-Hertz, which stands for $1,000,000,000$ Hz) or even higher (e.g., 5 GHz). Many other communications standards for consumer electronics, such as Bluethooth (which are used for wireless headphones), also operate at around 2.4 GHz. All these communication systems (and many other systems used for television, emergency radio, defense, satellite communication, astronomy, weather radar, etc.) are operating simultaneously and they are occupying different frequencies. The are *sharing* the frequency spectrum.

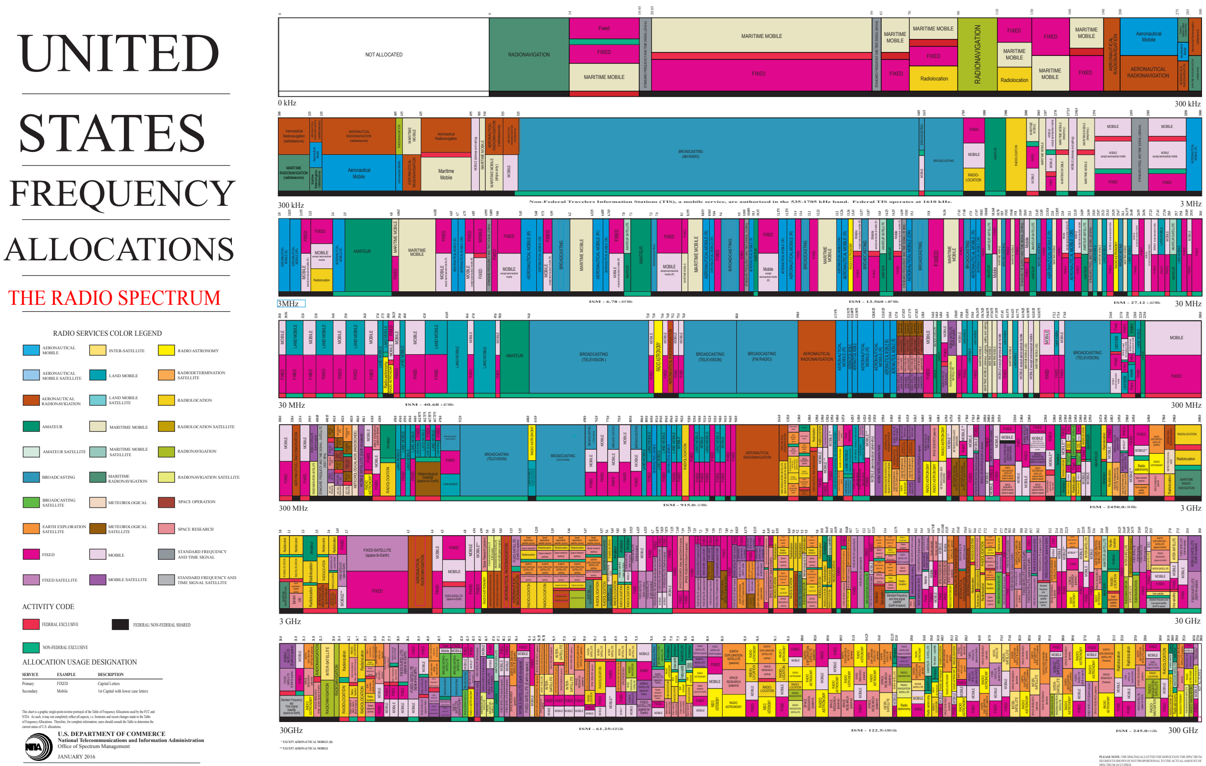


Figure 12: Frequency allocation chart of 2016 in the United States.

Figure 12 shows the United States frequency allocation chart from 2016 in the range of 1 kHz (kilo-Hertz, which stands for 1000 Hz) to 300 GHz. As you can see, nearly all frequencies are occupied by communication standards. Hence, when designing a communication system, one has to ensure to only occupy the frequency range allocated for your transmission system. In fact, if you somehow cause interference with neighboring systems, you will get into legal trouble with the FCC (and you have to pay ridiculously large fines!). To this end, when designing communication systems, one regularly uses the analysis tools we have introduced before: the spectrum and the spectrogram.

One of the key goals of almost every communication system is to transmit as many bits per second (this is known as the throughput) as possible through the provided bandwidth. The *bandwidth* of a communication system specifies how many contiguous frequencies one can use; a bandwidth of 1 MHz means, for example, that one can use the frequency spectrum from 20 MHz to 21 MHz. For this example, the center frequency would be at 20.5 MHz (which is right in the middle of the two frequencies). Measuring the amount of bits per second that can be transmitted per bandwidth is known as the *spectral efficiency*. Engineers who design communication systems are interested in improving spectral efficiency, rather than achieving the highest possible throughput (also known as the data rate; the throughput is only a measure of how many bits per second we can reliably transmit). In contrast, marketing people are mainly interested in advertising the highest possible throughput—this, however, does not tell you how many frequencies you are occupying (and cannot be used by other communication standards anymore). What most of us do not know: It is very easy to design communication systems that achieve high throughputs (for example, one can simply occupy all frequencies and not let anyone else transmit). It is, however, extremely difficult to achieve high throughput while only occupying a tiny fraction of the available frequency spectrum—that's why so many engineers are working on the design of wireless communication systems!

Activity 17: Compare spectral efficiencies of real-world communication systems

IEEE 802.11 is the name of many wireless LAN communication standards. If, for example, you are using Wi-Fi on your cell-phone, you are using one of the IEEE 802.11 standards. Since 1997, many wireless LAN standards have been proposed. The following table lists the key design parameter and performance metrics of some of them:

IEEE Standard	Year	Center Frequency [GHz]	Bandwidth [MHz]	Data rate [Mbit/s]
802.11g	2003	2.4	20	54
802.11n	2009	2.4	40	600
802.11ac	2013	5	160	3466.8
802.11ay	2020	60	8,000	20,000

IEEE 802.11g, for example, has a spectral efficiency of $54/20 = 2.7\,\text{bit/s/Hz}$ (simply take the ratio of the data rate and the bandwidth; be careful with the units). Compute the spectral efficiency for the other three standards. What do you observe? Which of these standards is the most efficient (in terms of spectral efficiency)? Which one achieves the highest data rate? Also, do you have an explanation why more recent standards are operating at higher center frequencies (up to 60 GHz for IEEE 802.11ay)? ***Let us know what you think!***