Hand-in Assignment 1 (HA1):
Fundamental of Acoustics

Department of the Built Environment, Building Acoustics Group
7LS8M0 Architectural Acoustics

version 1.0

February 2022

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Contents

Instructions

The guideline of the Hand-in Assignment 1 (HA1) includes the concepts as well as the tasks needed to be submitted. Two MATLAB examples are provided as part of this assignment. Your report is needed to be submitted on Canvas **before** the deadline. **Late submissions are not accepted, and graded with zero!** Your submitted report is needed to be in a **pdf** format and **named** such GroupNumber_Assignment.pdf (e.g., Group01_HA1.pdf) and your MATLAB scripts such GroupName_Assignment.m (e.g., Group01_HA1.m). In addition, both the MATLAB functions and scripts are needed to be separately submitted in a **zip** file. All the MATLAB scripts need to be commented very clearly, explaining what you are doing in every line of the code using comments. The figures must have the same format as the figure generated. All the generated figures need include title, labels in axes, and legends. In your report, the figures need to have a proper size, so that all the information to be well observable. At the appendix of your report, you need to include all your MATLAB codes. All the tasks are needed to be accomplished and included into the report. For collecting the full points per task, the answers should be fully motivated, correct and clear. Support your evidence with the proposed literature or other literature from your side. Finally, the distribution of points in HA1 is presented in the Table 1.

Table 1: Distribution of Points in HA1.

	Tasks Sec. 2	Tasks Sec. 3	Tasks Sec. 4	Structure/Lang.	References/Cites	Total
Points (pts)	2.50	4.00	0.50	0.50	0.50	8.00

1 Introduction

This assignment focuses on fundamental of acoustics topics related to harmonic and recorded time signals and their analysis in the frequency domain. Considering time signals, their characteristics related to their amplitude and/or phase can be extracted, transforming the signals from time-domain to frequency-domain via the Fourier Transform. It is going to be seen that the distortion of the time-pattern of the signals influences the representation of their components in the frequency domain as well. Finally, various representations of the signals components (i.e., continuous vs discrete) in the frequency domain is explored.

2 Discrete (Time) Signals

The first tasks of the assignment are mainly associated to the generation of discrete harmonic time-based signals, distortion effects, and representation of signals in time-domain.

2.1 Pure Sinusoidal Signals

Pure (i.e., harmonic) sinusoidal signals are expressed such,

$$y[t] = A\sin(2\pi ft + \phi), \quad t \in [0, T] \tag{1}$$

where, A is the amplitude (-), f is the frequency (Hz), ϕ is the phase (rad), t is the time-vector (s), and T is the total duration (s) of the signal y. Recall that the time vector (t) depends on both the total duration (T) and the sampling frequency (f_s). For further information, see the Chapter 1.4.1 in reference [2].

Tasks [0.50 pts]

2.1.A. Open and complete the MATLAB function `SinGen`. By calling `SinGen` in your script and defining the values of parameters A, f, ϕ, T, and f_s, the sinusoidal signal, $y[t]$, and its time vector, t, are generated.

2.1.B. Generate two sinusoidal signals $s1$ and $s2$ with $f_1 = 200Hz$, $f_2 = 205Hz$, $A_1 = A_2 = 1$, $\phi_1 = \phi_2 = 0$ $rads$, $T_1 = T_2 = 2s$, and $f_s = 4000Hz$. Plot the signals s_1 and s_2 over time into the same figure. Keep only the first 100 samples in your plot.

2.1.C. Compute and plot the signal $s3 = s1 + s2$ over time.

2.1.D. Describe the effect on **Task 2.1.C.**, its cause and provide its value.

2.2 Distorted Sinusoidal Signals

Suppose that the pure sinusoidal signals, generated by the `SinGen` function, are distorted by some factors. Here, two possible distortion effects are considered; the "clip" and the "click" effect.

Tasks [0.50 pts]

2.2.A. Generate a sinusoidal signal $(sTr1)$ of $A = 1$, $f = 1Hz$, $T = 1s$, and $f_s = 4kHz$. Then, zero pad the last 1000 samples so that a "click" effect to be present. Plot the resulted signal. Hint: use `zeros` function.

2.2.B. Generate a sinusoidal signal with the same characteristics as the $sTr1$ and name it $sTr2$. Then, pad the samples ranging from 335 to 1667 with the value 0.5 so that a "clip" effect to be present. Plot the resulted signal $(sTr2)$ together with the previous signal $(sTr1)$ over time. Hint: use the `ones` function.

2.2.C. In which circumstances these effects may be present?

2.3 Sinusoidal-based Complex Signals

Complex signals can be composed by pure sinusoidal signals. The square-wave function can be composed via a sum of odd harmonic sinusoidal functions. This is known as the Fourier series expansion for a square-wave. This can be expressed such,

$$y[t] = \sum_{f \in odd}^{f_{max}} \frac{\sin(2\pi f t)}{f}. \tag{2}$$

As $f_{max} \to +\infty$, the "pure" solution could be written such,

$$y[t] = \frac{\pi}{4}\mathrm{sgn}(\sin(2\pi t)). \tag{3}$$

Tasks [1.25 pts]

2.3.A. For the first 10 odd harmonic frequencies f, calculate the expression (2). Set, $T = 1s$ and $f_s = 44100Hz$. Hints: store each odd harmonic in the columns of a matrix, follow the same procedure for expression (2), and use the **`for`**-loop function.

2.3.B. Plot each odd harmonic signal (i.e., term into the sum of the expression (2)) in the same plot over time. Hint: Use a separate **`for`**-loop function.

2.3.C. Plot the expression (2) in an hierarchical way (i.e., plot your sum every time when a new harmonic is added) over time. In the same plot, include the "pure" solution expressed by equation (3). Hints: use and check the **`sum`** function and use the same **`for`**-loop as in the **Task 2.3.B.**.

2.3.D. Discuss your findings as the $f \to f_{max}$. What are you observing? Extra reading: see 4.2.4 in referece [2].

2.4 Audio Signals

Recorded signals, related to audio files or audio signals, can be read in MATLAB, using the `audioread` function (i.e., usage: `[s,fs]=audioread('AudioFile.wav')`). This function returns the values of the audio signal (`s`) as well as its sampling frequency (`fs`).

Tasks [0.25 pts]

2.4.A. Read the audio files Viola1.wav, Viola2.wav, ImpNarrow.wav and ImpWide.wav. The first two signals[1] are recordings of two viola musical instruments, whereas the other two of a narrow and a wide impulse.

2.4.B. Plot the two viola signals over time into the same plot.

2.4.C. Plot the two impulse signals over time into the same plot.

3 From Time Domain to Frequency Domain

Fourier transform is used for the transformation of the signals, $s[t]$, from time domain to frequency domain, $S[f]$. This is known as the frequency response $s[t]$, from which the components of the time-based signals can be revealed. More specifically, the amplitude and phase components over frequency are extracted. These responses are known as magnitude, $|S[f]|$, and phase response, $\angle S[f]$, respectively. Further information can be found in the Chapter 2.9 of the reference [1] and/or in the Chapter 4 of the reference [2].

[1] **`sound`** command can be used to listen to the viola signals.

In MATLAB, the Fourier Transform is computed via the build-in **`fft`** function. This function computes the discrete Fourier transform (DFT) of a time-signal with respect to an Fast Fourier Transform (FFT) algorithm. The DFT is the representation of the Fourier transform in the discrete domain.

Although the magnitude response is mathematically symmetrical with respect to zero $[-f_s/2, +f_s/2]$, it is common to *plot* only the positive magnitude response $[0, +f_s/2]$. To retain the energy in the positive magnitude response, it is needed to be multiplied by a factor of two. Note that in MATLAB, a DC component is introduced, indicating that the two-sided spectrum is in $[0, f_s]$, and the one-sided in $[0, +f_s/2]$. See **`FFTs.m`** example.

Tasks [1.75 pts]

3.0.A. Transfer the example **`FFTs.m`** in the **`SpFFT`** function. The input arguments are the time signal, and its sampling frequency, whereas the output arguments are the two-sided frequency-based signal, (i.e., the non-shifted FFT) and its frequency vector.

3.0.B. Plot the one-sided magnitude response of the $s1$ signal, generated in the **Task 2.1.B**.

3.0.C. By setting that the $T = 4s$ in the $s1$ signal, re-plot its one-sided magnitude response and describe the possible difference with the plot in **Task 3.0.B.**?

3.0.D. Plot the one-sided magnitude responses of the signals generated in the **Task 2.2.A.** and **Task 2.2.B**. What do you observe?

3.0.E. Plot the one-sided magnitude response of the full signal generated in the **Task 2.3.C.** What do you observe?

3.0.F. Plot the one-sided magnitude responses of the signals generated in the **Task 2.4.A.** Discuss the differences between the viola and impulse signals.

3.1 Continuous Spectrum

In acoustics, the term spectrum is associated to the representation of the transformed pressure signals, $p(t)$, to the frequency domain, $p(f)$, with respect to meaningful quantities such as the Sound Pressure Level (SPL-dB). Further information can be found in Chapter 2.1 and 3.5 of [1]. The SPL-based spectrum is written such,

$$L_p(f) = 10 \log_{10} \frac{|\tilde{p}(f)|^2}{p_{ref}^2}, \tag{4}$$

where, $\tilde{p}(f) = p(f)/\sqrt{2}$ is the root-mean-square (RMS) pressure and $p_{ref} = 20 \cdot 10^{-6}\ Pa$ the reference pressure.

Tasks [0.25 pts]

3.1.A. Compute and plot the continuous one-sided spectrum of the viola signals in the **Task 2.4.A.**

3.2 Discrete Spectrum

A different representation of the signals in the frequency domain is associated to the discretization and summation of spectrum in frequency bands. The most common frequency bands are the octave and one-third octave bands (i.e., 1/1-octave and 1/3-octave bands, respectively). The SPL in nth-octave frequency bands is expressed such,

$$L_{n^{th}-oct}(f_c) = 10 \log_{10} \left(\sum_{f_l}^{f_u} \frac{|\tilde{p}(f)|^2}{p_{ref}^2} \right), \tag{5}$$

where, f_c, f_l, and f_u are the central, lower and upper frequency for nth-octave frequency band, respectively.

Usually, when the spectrum is considered in frequency bands, the whole frequency range is reduced until $f_{max} = 10kHz$. Each 1/1- and 1/3- octave frequency band is characterized by its central, lower, and upper frequency, which are based on the band number (m). In Chapter B.4 of reference [3], these are defined such,

$$f_{l,1/3} = f_{c,m_{1/3}} \cdot 2^{-\frac{1}{6}} \qquad f_{c,m_{1/3}} \approx f_{c,30} \cdot 2^{-10+\frac{m_{1/3}}{3}} \qquad f_{u,1/3} = f_{c,m_{1/3}} \cdot 2^{+\frac{1}{6}} \tag{6}$$

$$f_{l,1/1} = f_{c,m_{1/1}} \cdot 2^{-\frac{1}{2}} \qquad f_{c,m_{1/1}} \approx f_{c,30} \cdot 2^{-10+\frac{m_{1/1}}{3}} \qquad f_{u,1/1} = f_{c,m_{1/1}} \cdot 2^{+\frac{1}{2}} \tag{7}$$

where, $m_{1/3} = [10 : 40]$, corresponding to the frequency range from $10Hz$ to $10kHz$, $m_{1/1} = [12 : 3 : 39]$, corresponding to the frequency range from $16Hz$ to $8kHz$, and $f_{c,30} = 1kHz$.

Tasks [2.00 pts]

3.2.A. Compute the f_c, f_l, and f_u for 1/1- and 1/3-octave frequency bands. Store the expressions in two functions with the names, **`OctBnd11`** and **`OctBnd13`**.

3.2.B. Calculate and plot the SPL inside each of the 1/1- and 1/3-octave frequency band for Viola1 and Viola2 using the expression (5). Plot the results in bar chart. Hint: see **`ForSumMinMax.m`** example.

3.2.C. Discuss the differences between SPLs in 1/3- and 1/1-octave frequency bands. Hint: focus on joint frequency bands.

3.2.D. Focusing on the different plots (i.e., **Task 3.0.F.**, **Task 3.1.A.**, and **Task 3.2.B.**), which representation would you prefer for the viola signals and why?

3.2.E. Suppose that you have three different signals; (i) a road-traffic signal, (ii) a music signal, and (iii) a Gaussian signal. Which representation in the frequency domain would you choose for each signal? Support your ideas. No computation is needed.

4 From Frequency Domain (Back) to the Time Domain

Similarly to Fourier Transform, the Inverse (Fast) Fourier Transform (IFFT) is used for transforming the frequency-based signals, $S[f]$, to time-based signals, $s[t]$. For this purpose, the build-in function **`ifft`** is used.

Tasks [0.50 pts]

4.0.A. Compute the inverse Fourier Transform of the signal $s1$ generated in the **Task 2.1.B**.

4.0.B. Plot the transformed and the original signal over time.

4.0.C. What is the difference between the original and the transformed signal?

References

[1] Heinrich Kuttruff. *Acoustics : an introduction.* Taylor & Francis., first edition, 2007.

[2] John G. Proakis and Dimitris G Manolakis. *Digital Signal Processing: Pearson New International Edition.* Pearson, fourth edition, 2014.

[3] Erik M. Salomons. *Computational Atmospheric Acoustics.* Springer, first edition, 2001.