Lab 1: Signals and Systems

November 20, 2015

Introduction

The course *Signals and Systems* consists of lectures, tutorials and lab exercises. There are three lab assignments (sessions). The assignments are made in pairs (or individually if you really insist). Both partners receive the same grades for the lab sessions.
The lab assignments are made with Matlab or Octave. Octave is a public domain clone/lookalike of Matlab and is available for virtually any operating system. On most Linux distributions, Octave is available in the software repository. You can also download it, including its documentation, from the Octave homepage (`http://www.gnu.org/software/octave`). Matlab is a commercial product and is quite expensive. For this reason, all supplied codes are made and tested with Octave. Therefore, you are advised to use Octave as well. If you prefer to use Matlab, however, that is perfectly fine. Most codes will run out of the box or need very little tweaking. If you are not used to programming in Matlab/Octave, then you are referred to a short crash course in Matlab which is available via Nestor.
For the first session, you are requested to solve the exercises and answer the questions below. The main objective of the first exercise is to get to know the Octave/Matlab environment and to explore some fundamental properties of sinusoidal waves. **A full report is not needed (answer the questions and provide explanation when asked, but do not write a full problem analysis, design of algorithm etc.)**. For the remaining sessions you will have to hand in a (brief) report. You must mail your reports/answers digitally (as a pdf) to `sigsys1516@gmail.com`. Please include your Matlab/Octave source codes as well, so that the teaching assistant can test them.
[Tip: take a headset with you to the lab rooms. Do not make sound using the speakers of the PCs, since it will surely disturb others.]

1 Summing Sinusoids

(a) Write a function `gensinusoid(A,f,phi,t1,t2,Fs)` that returns a discrete (i.e. sampled) sinusoidal wave that is defined by its six parameters:

- `A`: amplitude of the wave
- `f`: frequency in Hertz (Hz)
- `phi`: phase at $t = 0$ s.
- `t1` and `t2`: start and end time of the sampled wave (in s.)
- `Fs`: sample rate, i.e. number of samples per second

(b) Generate the signal that corresponds to $x1(t) = 5\cos(800\pi t)$, for $t \in [0, 0.5]$. Make a plot of the first 200 samples. Use a sample rate of 8000 samples per second. How many samples correspond to one oscillation?

(c) Write a function `[A,f,phi]=sumsinusoid(A1,f,phi1, A2,phi2)` that returns the resulting amplitude, frequency and phase of the signal that is obtained by adding two signals that are given by the parameters of the function (note that both signals have the same frequency).

(d) The signal $x2$ is a time-shifted version of the signal $x1$ given by $x2(t) = x1(t + 0.013125)$. Generate the signal $y(t) = x1(t) + x2(t)$ for $t \in [0, 0.5]$. What is the formula for this signal if we write it in the form $y(t) = A\cos(2\pi f t + \phi)$? Write a script to verify your answer.

(e) Given is the signal $x3(t) = \cos(1600\pi t)$. What do you think the signal $z1(t) = x1(t) + x3(t)$ will look like? Make a plot to verify your answer. Do the same for the signal $z2(t) = x1(t) + x4(t)$ where

$x4(t) = \cos(3200\pi t)$. Finally, try the same for $x5(t) = \cos(32000\pi t)$. What do you hear if you listen to this signal?

2 A microphone and some sources

(a) The speed of sound is approximately 343.2 m/s. Write a script that determines the signal that a microphone at location `(micx,micy)` would perceive given a number of sound sources (each source produces a perfect sinusoidal wave). Make the script as general as possible, i.e. you must be able to vary the sampling rate, the number of sources, their frequencies, amplitudes and phases.
You may assume that the signals start at $t = -\infty$ s. Hence, there will always be an audible signal at $t = 0$ s. You can use $\texttt{t1} = 0$ s and $\texttt{t2} = 0.5$ s. You should not simulate energy loss in the signal, so i.e. the amplitude will should depend on the location of the microphone.

(b) Using the script you made in part (a), show that two microphones at different locations may actually perceive different signals (not only a phase shift).

(c) Figure out a constellation of two microphones and two sources such that one microphone does not perceive any signal at all, while the other microphone does perceive sound.

(d) Consider the following constellation.

- A source that produces sound with a frequency of 400 Hz and amplitude 5 at location (0,100).
- Three microphones: one at location (0,0), one at location (50,0) and one at location (100,0).

Determine for each microphone the signal that it perceives. Predict what the signals will look like. Verify your answer by plotting (parts of) the signals.

(e) Write a function that, given two periodic signals `x[n]` and `y[n]`, computes the value `delta` such that `x[n]` and `y[n+delta]` are most similar. Define yourself a suitable similarity measure. Apply this function to the signals that you obtained in part (d). Explain the results.

(f) In this exercise we use the following constellation of three microphones. Microphone 1 is at location `(0,0)`, microphone 2 is at location `(dmic,0)`, and microphone 3 is at location `(2*dmic,0)`, where `dmic` is the distance between two neighbouring microphones (in meters).
Assume that there is a sound source at some location `(p,q)`, where `q` is a positive number. Each of the microphones will record a time shifted (delayed) version of the emitted signal. Consider the delay between the signals perceived at microphone 1 and microphone 2. This delay, which is the difference in arrival time at the microphones, contains information about the possible locations of the emitter. Using two microphones, we cannot detect the location of the emitter, but we can make an equation for all possible emitter locations `(x,y)`. Explain why this equation defines a *hyperbola*. Now, if we also construct such an equation using the difference in time of arrivals between microphone 2 and microphone 3, then it is possible to compute `(p,q)`, since we only need to intersect the two corresponding hyperbolas. On Nestor, you can find a matlab file `dtoa.m`, that does this computation for you. For example, the call `dtoa(1, 0.0020062, 0.0018863)` returns the approximate location of the emitter, if we place the microphones 1 meter apart (i.e. `dmic=1`), and we observe delays of 0.0020062 seconds, and 0.0018863 seconds (this example corresponds with `(p,q)=(10,10)`.
Write a Matlab function that has as its input `dmic` and the three signals perceived by the microphones. The output of this function must be the (approximated) location of the emitter. Test your function thoroughly, and explain your results.

3 Doppler effect

Consider a moving sound source that produces sound with a frequency of 400 Hz. The sound source moves with a speed of 180km/h from (0,10) to (250,10), where the coordinates are in meters. A microphone (with sampling rate 8000 samples per second) is located at location (125,0). Compute the signal that this microphone will perceive and play the sound. Explain what you hear.