SATFD

Lab. 3

March 22, 2024

Exercise 3: Time and Frequency

Introduction

Refer to the lecture. The purpose of the exercise is to compare the Short-Time Fourier Transform (STFT) with bilinear transforms - Wigner-Ville and other transforms from Cohen's class.

Realization of the exercise

Suggested environments: Matlab, Python.
Prepare a script that will perform the following tasks:

1. Generate a signal of length $N = 512$ and sampling frequency $f_s = 500Hz$: a linear chirp with initial frequency $f_1 = 20Hz$ and final frequency $f_2 = 100Hz$. The chirp is generated by the *chirp* function, which arguments include time (should be defined as a vector - e.g., $t_n = (1 : N)/f_s$), initial frequency, time at which the final frequency should be reached (i.e., N/f_s), and the final frequency. In Python, the *chirp* function is available in the *scipy.signal* library.

2. Generate a spectrogram - other speaking, the Fourier transform in a short window.

   Matlab: To generate a spectrogram, you can use the *spectrogram* method. The function's arguments are: signal, window length (default: Hamming window), overlapping length (defines whether adjacent windows overlap), FFT window length: $nfft$, sampling frequency.
   Python: To generate a spectrogram, you can use the spectrogram method available in the *scipy.signal* library. The function's arguments are: signal, sampling frequency, window type, window length, overlapping length.
   The standard value for the overlapping parameter is $nfft/2$. Compare the plots for different $nfft$ values: 16, 32, 64, 128, and 256 as well as 512 for a signal twice as long: of length $N = 1024$ or four times longer $N = 2048$. Discuss the obtained result from the perspective of the Heisenberg uncertainty principle (in the signal theory version - see the lecture).

3. Generate a Wigner-Ville transform of the signal. The VW transform requires an analytical function, so before the VW transform, a Hilbert transform of the signal should be performed. The function returns the transform as a real two-dimensional function and additionally a time vector and a frequency vector, which can be used to describe the axes of the plot. Since the W-V transform can give negative values, the absolute value should be taken before plotting. Display the modulus of the transform and the logarithm of the modulus.

   Matlab: The hilbert function is used for the Hilbert transform. Cohen class transforms are implemented in the *cohen.m* file. Additionally, the two-dimensional convolution function: *conv2.m* (should be standard in Matlab distribution) and auto-correlation *int_autocorr.m* are used. The *cohen* function takes as arguments the analytical function, sampling frequency, and an argument that specifies which kernel (core) (cf. lecture) is to be used. The VW argument gives the Wigner-Ville kernel. For 3D drawing, many functions are used, the simplest is `imagesc(t,f,<your 2-D function here>); axis xy;`
   Python: The *hilbert* function from the `scipy.signal` library is used for the Hilbert transform.

Cohen class transforms are implemented in the *cohen.py* file. The *cohen* function takes as arguments the analytical function and the sampling frequency. For 3D drawing, many functions are used, including `pcolormesh(t,f,<your 2-D function here>)`.

4. Then, select a specific FFT window length - for example, $nfft = 64$, and then show side by side the VW logarithm transform for a signal of length $N = 512$ and STFT. Under such prepared conditions, you should:

   (a) Change the final chirp frequency to $f_2 = 200Hz$ and then $f_2 = 500Hz$ (you can also display intermediate: 220, 250, 270, 300, 400, etc.). Then comment on the result (in the report). Comment = understand where it came from.

   (b) Determine the spectrum for the sum of two chirps changing in parallel: one changing from $f_1 = 20$ to $f_2 = 200$ and the other changing from $f_1' = 2 \cdot f_1$ to $f_2' = f_2 + f_1$.

   (c) Determine the spectrum for the sum of two chirps changing non-parallelly: one changing from $f_1 = 20$ to $f_2 = 200$ and the other changing from $f_1' = 2 \cdot f_1$ to $f_2' = 2 \cdot f_2$.

   Comment on all obtained results.