Spectral estimation methods

Lab 12, SDP

Objective

Students should use some well-known spectral estimation methods and one of its applications.

Theoretical notions

Exercises

1. Find the average value and the autocorrelation function of the signal $x[n]$ obtained as the output of an ARMA(1,1) random process with the following difference equation:

   $$x[n] = \frac{1}{2}x[n-1] + w[n] + w[n-1],$$

   where $w[n]$ is white noise with variance σ_w^2 and average value 0.

2. The autocorrelation function of an AR random process $x[n]$ is:

   $\gamma_{xx}[m] = \frac{1}{4}^{|m|}$.

   Find the difference equation of the random process $x[n]$. Is this unique? If not, find more than one possible solution.

3. In Matlab, create a signal of length $N = 1000$ defined as follows:

   $x[n] = \cos(2\pi f_1 n) + 0.5 \cdot \cos(2\pi f_2 n) + A \cdot noise$

   where $f_1 = 1000/44100$ and $f_2 = 1800/44100$, and the noise is random white gaussian noise (`randn()`).

   Try different values of A (e.g. 0.1, 0.3).

4. Estimate the power spectral density of the signal `x` in three diferent ways:

   1. Compute the Fourier transform $X(f)$ (with `fft()`), and display $|X(f)|^2$
   2. Use the function `periodogram()`
   3. Use the Yule Walker method (`pyulear`), with order 30 (try different values, from e.g. 5 to about 70).

   Pay attention to:

   - are the frequency peaks correctly located at f_1 and f_2?
   - are the frequency peaks wide or narrow?
   - is the noise spectrum flat or not?

5. Check if my guitar is in tune or not:

   a. Load the signal `1st_String_E.ogg` with the function `audioread()`, and display its spectrum Use `[pxx, f] = periodogram(x, [], [], Fs)` to obtain both the spectrum and the frequency values, and plot `pxx` against `f`.
   b. Identify the fundamental frequency
   c. Compare the value with the frequency values of a standard guitar tuning (see Wikipedia page on Guitar Tunings)
   d. Repeat for all other strings

6. In Matlab, create a script file which implements a live spectrum analyzer.

   a. Load the signal `music.wav` with the function `audioread()`.
   b. Use the function `buffer()` to split the signal into windows of length 30ms.
   c. Use `periodogram()` to estimate and plot, successively,the spectrum of each window signal.
   d. Optional: localize and plot the dominant frequency from the spectrum of each window. Convert the frequency to the corresponding musical note and output it.

Final questions

1. TBD