2 Representation of Signals II: Speech and Audio Signals

2.1 Overview

In this exercise, we are going to work with recorded speech and audio signals and apply basic estimation methods. In the lecture script, we refer to section 1.2.

Why analysis of speech and audio signals?

- Speech and audio coding: extraction for efficient information transmission
- Detection, Speech recognition and speaker identification: extraction of features carrying the information used for signal classification
- Human-machine interfaces: Algorithm design strongly dependent on the signal characteristics

2.2 Sample Sequences of Speech and Audio Signals

In the directory `./SHARED_FILES/spsa/Exercise2/` exists a subdirectory `wav` including two speech files and two audio files (sampling rate $f_s = 44.1\text{kHz}$). Copy the subdirectory `wav` into your home directory `spsaX`.

a, First, load the two speech files `female_german.wav` and `male_german.wav` into two MATLAB variables x_1 and x_2, respectively, using the command `audioread`. Don't forget the semicolon '`;`' after the command. M

– x_1: `female_german.wav`

– x_2: `male_german.wav`

Note that the two speech files are stereo and need to be converted to mono, by averaging over both channels, before proceeding. Now, plot both time-domain signals (Note: `figure` opens a new window for a separate plot.). The time axis should be scaled in seconds rather than samples (`plot(k,x)`). For the construction of the vector k, the sampling frequency has to be taken into account.

What time interval does the distance between two samples correspond to? T

▷ ______________________________

b, Speech is not stationary. In what time interval are speech signals nearly stationary (so-called *short-time stationarity*)? T

▷ ______

c, Using your plots, how can one determine whether the signal is speech of a female or male speaker? (Hint: Use the zoom button in the plot window). M

▷ ______

2.3 Relation to the Physical World

With MATLAB, one can send signals to the audio interface of the computer by using the command `sound` (or `soundsc`). See `help sound` for details.

a, At a certain sampling rate f_s, it is possible to exactly reconstruct audio signals only if a certain bandwidth is not exceeded. How large is that bandwidth? T

▷ ______

b, In order to convert a signal to a lower sampling rate, it has to be bandlimited (*anti-aliasing filter*). A very useful MATLAB function taking care of this job and the downsampling by a rational factor is `resample`. Using a headphone, it is easy to compare the audio qualities of the original signal (44.1kHz, HiFi) and the signal in telephone quality (8kHz): M

```
x1_mod = resample(x1, 8e3, 44.1e3);
soundsc ( x1, 44.1e3 );       % CD quality
soundsc ( x1_mod, 8e3 );      % telephone quality
```

2.4 Long-Term and Short-Time Analysis

According to 2.2 (b), it is appropriate for speech (and audio) analysis to distinguish between short-time analysis (short-time stationarity assumption) and long-term analysis (stationarity assumption). Note that in real systems (real-time processing, short delay using short blocks of data), only short-time analysis is possible.

a, Compare the histograms of x_2, measured over a short time interval and the entire signal, respectively. How can the sharp peak be interpreted? M

▷ ______

b, Load the audio files `castanets.wav` and `orchestra.wav` into the variables x_3 and x_4, respectively, and compare the corresponding long-term pdfs.

- x_3: `castanets.wav`
- x_4: `orchestra.wav`

▷ ______________________________

Analysis in the Time Domain

c, **Auto-Correlation Sequence (ACF) and Cross-Correlation Sequence (CCF).** In the applications, the ACF and the CCF are very important for the characterization of speech and audio signals. The known definition of the ACF is

$$R_{xx}[m] = E\{x[k]x[k+m]\}. \tag{5}$$

Here, it is assumed that the sample sequence is stationary, and can be observed for an unlimited time-interval. In real-world applications, only a limited observation interval is available. Two different estimates for the ACF are considered in the following, as they can be obtained from a signal segment of length N:

Estimate 1: Average over time for given $x[k]$, $k = 0, \ldots, N-1$, normalized to the *number of samples taken into account for the respective estimate*:

$$\hat{R}'_{xx}[m] = \begin{cases} \frac{1}{N-m} \sum\limits_{k=0}^{N-m-1} x[k]x[k+m] & \text{for } m \geq 0, \\ \frac{1}{N-|m|} \sum\limits_{k=|m|}^{N-1} x[k]x[k+m] & \text{for } m < 0. \end{cases} \tag{6}$$

Estimate 2: Average over time for given $x[k]$, $k = 0, \ldots, N-1$, normalized to the *total number of available samples*:

$$\hat{R}_{xx}[m] = \begin{cases} \frac{1}{N} \sum\limits_{k=0}^{N-m-1} x[k]x[k+m] & \text{for } m \geq 0, \\ \frac{1}{N} \sum\limits_{k=|m|}^{N-1} x[k]x[k+m] & \text{for } m < 0. \end{cases} \tag{7}$$

Both of these estimates are implemented in the MATLAB function `xcorr` (see `help xcorr`). Estimate 1 is called the *unbiased* estimate, estimate 2 is called *biased*.

To study the properties of these estimates, we first create the following signals: M

- v_1: `v1 = randn(1,1000);`
- v_2: `v2 = randn(1,10000);`

Calculate both the biased and unbiased estimates of $R_{v_1v_1}[m]$ and $R_{v_2v_2}[m]$ using `xcorr`. For comparison, the 'biased' and 'unbiased' curves should be shown in one figure using the commands `subplot`, followed by `plot` for each sequence.

Which version (biased/unbiased) is closer to the expected result (what is the expected result?) in terms of variance? M

▷ ____

How do the different lengths of v_1 and v_2 affect the results?

▷ ____

e, The function `randn` as used for v_2 creates a zero mean signal. Create a new signal v_3 by adding 1 to the signal v_2:

- v_3: `v3 = v2+1;`

What does the ACF of v_3 *ideally* look like? T

▷ ____
▷ ____

Now, calculate the biased and unbiased estimates of the ACF of sequence v_3. What typical shape does the biased estimate have? M

▷ ____

What are the advantages and disadvantages of the biased and unbiased estimates?

▷ ____
▷ ____
▷ ____
▷ ____

f, Plot the ACFs of the audio signals x_1, x_2, x_3 and x_4 for the observation intervals 6s (biased) and 20ms (unbiased). Can the pitch periods of x_1 and x_2 be determined using these plots? How are the formant frequencies represented? M

▷ ____
▷ ____
▷ ____

Analysis in the Frequency Domain

g, As an estimate for the power spectral density (PSD)

$$S_{xx}(e^{j\Omega}) = \sum_{m=-\infty}^{\infty} R_{xx}[m]e^{-j\Omega m} = \lim_{M\to\infty} \frac{1}{2M+1}\left|X_N(e^{j\Omega})\right|^2, \tag{8}$$

we consider the so-called *periodogram* $I_{xx}(e^{j\Omega})$, which is straightforwardly obtained using the Discrete Time Fourier Transform of estimate 2 for the ACF:

$$I_{xx}(e^{j\Omega}) := \hat{S}_{xx}(e^{j\Omega}) = \sum_{m=-N+1}^{N-1} \hat{R}_{xx}[m]e^{-j\Omega m} = \frac{1}{N}\left|X_N(e^{j\Omega})\right|^2 \tag{9}$$

In general, the limitation in the time domain (i.e., rectangular windowing) leads to an undesired slow decay of the spectral envelope. Therefore, other window functions are desirable.

For our simulations, we first use a new artificial signal v_4, derived from v_2, but with variance $\sigma_{v_4}^2 = 0.25$: M

– v_4: `v4 = a*v2;`

Calculate a periodogram for length $N = 10000$ for a rectangular window, a Hamming window (function `hamming(length)`), and for a Blackman window (function `blackman(length)`). For the calculation of the Discrete Fourier Transform, the function `fft` can be used. Determine and discuss for each window function the variance of the error between the periodogram and true PSD (quadratic mean of the deviations of the frequency bins).

▷ ____________________

▷ ____________________

i, We now analyze the speech signal x_2. How are the formant frequencies and the pitch frequencies represented in the long-term PSD?

▷ ____________________

▷ ____________________

Use the Welch method (in MATLAB using the function `pwelch`) and data windowing to estimate the PSD of x_2. Plot it on the log scale (in decibel). (*Hint:* You can plot the PSD estimate using `pwelch` without having to call `plot`).