Lecture "Digital Signal Processing"

Prof. Dr. Dietrich Klakow, Summer Term 2021

Assignment 3

Submission deadline: 10 May 2021, 23:59

Submission Instructions:

You have one week to solve the assignments.
The code should be well structured and commented. Do not use any Matlab-Toolbox or Python external libraries if it is not mentioned that you can use them.

- You are required to hand in your solutions in a group of two students.
- There are two parts in this assignment: theoretical part and practical part.
- The practical part that is solved with Python should be submitted as an ipynb file (Jupyter Notebook or Google Colab), where every function is written in a separate block.
- The theoretical part can either be written by hand and scanned, or typed with LaTeX in the text / markdown area of ipynb file.
- Submission of both parts should be done via Microsoft Teams by one of the group members.
- Submission should be named as: Ex03_matriculationnumber1_matriculationnumber2.zip

The submission should contain the following files:

- file "README" that contains an information on all team members:
  name, matriculation number, and email address.
- code files
- file "answers.pdf" which contains answers to the questions appearing in the exercise sheet. *Note: If you use ipynb file, you don't have to submit "answers.pdf". You can embed your scanned copy or write your answers in the text / markdown area.*

1 (2P) Linear PCM & Quantization

Linear Pulse Code Modulation (PCM) is a waveform coding technique that quantizes each sample using scalar quantization. With B bits, it is possible to represent 2^B separate quantization levels. Linear PCM is based on the assumption that the input signal $x[n]$ is bounded, i.e. $|x[n]| \leq X_{max}$, and we use uniform quantization with quantization step size Δ which is constant for all levels x_i, i.e. $\Delta = x_i - x_{i-1}$.

Using mid-tread quantizer the output $\hat{x}[n]$ can be obtained from the code symbol $c[n]$ given as follows:

$$c[n] = \lfloor \frac{|x[n]|}{\Delta} + 0.5 \rfloor \, sign(x[n]) \tag{1}$$

$$\hat{x}[n] = c[n]\Delta \tag{2}$$

1. Assuming that the quantization error $e[n]$ follows a uniform distribution and it is white and additive, derive the range of the quantization error $e[n]$ and its variance σ_e^2.

2. Assuming that the input signal $x[n]$ also follows a uniform distribution and is uncorrelated to the error, derive its variance σ_x^2, and show that each quantization bit contributes to $\sim 6dB$ of SNR.

2 (2P) Cholesky Decomposition

Yule-Walker equation can be solved using Cholesky decomposition. In general, Cholesky decomposition of a real matrix A can be written as follows:

$$A = LL^T \tag{3}$$

List the requirements on matrix A and the properties of matrix L.
State the formulae for computing matrix L.
Decompose the following matrix manually:

$$\begin{bmatrix} 4 & 2 & 6 \\ 2 & 5 & 5 \\ 6 & 5 & 19 \end{bmatrix}$$

Optional: You can verify your answer by using the built-in function *chol* in Matlab or *cholesky* from scipy.linalg library of Python.

3 LPC Algorithm

Yule-Walker equation that solves the objective function of Linear Predictive Coding (LPC) can be expressed in matrix form of:

$$\Phi a = \varphi \tag{4}$$

3.1 (2P) Functions to compute Φ and φ

Implement two functions that compute and return Φ and φ, respectively, based on the derivation that has been described in the lecture. Each of the functions takes two arguments, signal vector and order (P) of the LPC model. Name the functions *cap_phi(signal, order)* and *sml_phi(signal, order)*, respectively.

3.2 (1P) Function to compute a

Write a function that returns the prediction coefficients (a) vector of LPC. The input arguments of the function are signal vector and order of LPC. Inside this function you will use the two functions created in task 3.1, and your favorite linear systems of equations solver (you can use any built-in method from Matlab or Python numpy) to find the coefficients. Name the function *solve_coeff(signal, order)*.

3.3 (1P) LPC on recorded audio

1. Record audio of "*dsp*" and read them as vector (use sampling rate 48000, single channel, and duration 2 seconds).
2. Store only 72000 samples from the middle of the signal and store it as wav file.
3. Plot the signal to visualize the stored samples.
4. Play the wav file to check the audio recording.
5. Calculate its LPC coefficients (use order 16), and print the coefficients.

Note: Include the wav file in the submission for checking.

3.4 (1P) Prediction

Write a function that takes a portion of signal, predicts the next value of that signal, and returns the value. *Hint: use portion of signal and coefficient vector as input argument.*

3.5 (1P) Error computation

1. Use the saved signal and the calculated prediction coefficients to find the error signal. You can write this step as a function.
2. Apply the function on the recorded audio.
3. Plot the predicted signal overlaid on the original signal, as well as plot the prediction error.