Lecture "Digital Signal Processing"

Prof. Dr. Dietrich Klakow, Summer Term 2021

Assignment 4

Submission deadline: 17 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: Ex04_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 Theoretical Part

1.1 (2P) Toeplitz Matrix

Given a matrix $R \in \mathbb{R}^{(n+1)\times(n+1)}$, which is positive semi-definite and whose elements are as follows:

$$r_{ij} = \frac{1}{n+1} \sum_{k=0}^{n} x_{k+i}\, x_{k+j} \tag{1}$$

It is an approximation to an autocorrelation matrix of a (stochastic) process, whose signal has a length of $(n+1)$. Show that:

1. the matrix Φ of Yule-Walker equation, and

2. the autocorrelation matrix of processes with periode $(n+1)$

can have a Toeplitz-structure, which is as follows:

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_n \\ r_{n+1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_{2n} & \cdots & r_{n+1} & r_0 \end{pmatrix}$$

1.2 (2.5P) LPC with Regularizer

1. Derive the Yule-Walker equation that minimizes the error in Linear Predictive Coding (LPC) added with L_2 regularizer. The necessary equations can be found in the Lecture Notes of DSP Chapter 4, and L_2 regularizer is basically the sum of the squares of all the prediction coefficients.

2. Can Cholesky Decomposition and Levinson-Durbin Recursion methods still be used to find the prediction coefficients of the above equation? Why / Why not?

2 Practical / Programming Part

Allowed Python libraries: numpy, wave, and matplotlib.pyplot. Should you need to record audio, you can use sounddevice, and to play the recording, you can use playsound. Matlab libraries would be the equivalent ones.

2.1 (2P) Auto-correlation

1. Implement a function that does auto-correlation on a given signal. The function should take as arguments: input signal and maxlags (i.e. number of lags to show), and return the lag vector and auto-correlation vector (as defined in Python *matplotlib.pyplot.acorr*).

2. Apply the function to the audio "dsp" file that you have recorded in Tutorial 3, and store the first 16 lags.

3. Plot the result in comparison to the built-in plot of matplotlib.pyplot.acorr (i.e. show two plots: one with your own function, another with Python built-in *acorr*).

2.2 (3.5P) Levinson-Durbin Recursion Algorithm

1. Write a function to solve the Yule-Walker equation using Levinson-Durbin recursion. *Hint: the function should take as argument the non-negative part of the auto-correlation vector that you have produced in task 2.1, and return the prediction coefficients* a.

2. Apply the function to your output of task 2.1, and print the prediction coefficients. Please also submit the audio file that you used.

3. What is the computation time of the Levinson-Durbin algorithm explored in the lecture?