8.1.2023/PA

ELEC-E5522 Speech Processing Project (5 cr)

Project outline

1. Introduction

The goal of ELEC-E5522 Speech Processing Project is to familiarise the student with the design, implementation, evaluation and documentation of a selected speech processing topic. Together these parts constitute *a speech processing project*. As the project topic, the course addresses *speech coding based on linear prediction (LP)*. Speech coding is a justified project topic because it represents a ubiquitous speech technology application that has been used widely for several decades in ICT. In addition, by working on this topic the student learns essential skills that are needed in speech technology in planning and implementing signal processing algorithms as well as in conducting subjective evaluation tests in speech technology.

The course is arranged as follows. First, an introduction lecture is given to recap the necessary background methodologies in LP that are needed to work in the project. Second, by working in teams of 2-3 members, the student conducts the project through the following steps:

(1) The design and implementation of a speech codec using MATLAB (or Python) and given instructions.
(2) The analysis and objective evaluation of the implemented codec.
(3) The informal, small-scale subjective evaluation of the implemented speech coding method.
(4) The documentation of the entire project.

The teams work in the project according to a pre-selected schedule consisting of four milestones. At each milestone, the team's progress is reported to the lecturer in regular zoom / face-to-face meetings between the team members and the lecturer.

NOTE: You do not need to program everything from scratch, but you can use MATLAB (or Python) functions. Useful MATLAB functions: audioread, audiowrite, hamming, filter, fft, real, impz, lpc, levinson, xcorr, quantiz, freqz

Schedule:

- Introduction lecture (January 9, 2023, 14:16-16:00)
- Milestone 1: Sections 3.1-3.3 (completed by February 3, 2023)
- Milestone 2: Section 3.4 (completed by March 3, 2023)
- Milestone 3: Section 4 (completed by March 17, 2023)
- Milestone 4: Section 5 (completed by March 31, 2023)

Pre-requisites: The course focus is on **signal processing** methodologies in the selected topic of speech technology. Therefore, the student should have basic knowledge in speech processing (i.e. ELEC-E550 Speech Processing or a similar course done) and in signal processing (i.e. ELEC-C5231 Introduction to Signal Processing or a similar course done).

2. Speech data

The core idea of the project is the hands-on implementation of the topic using real speech. The speech data to be used consists of 20 sentences from the TIMIT database (Garofolo et al., 1993). These signals have been down-sampled from the original sampling rate of 16 kHz to 8 kHz and are expressed using a resolution of 16 bits, which means that the bit-rate of the input is 128 kbit/s. The sentences have been produced by 20 different speakers (10 female and 10 male speakers, all native speakers of US English). This data is saved as wav files on the course's MyCourses webpage. The names of the wav files and their orthographic transcriptions are as follows:

The speech files spoken by female speakers:
sa1_8kHz.wav ”She had your dark suit in greasy wash water all year.”
sa2_8kHz.wav ”Don't ask me to carry an oily rag like that.”
si614_8kHz.wav ”This big, flexible voice with uncommon range has been superbly disciplined.”
si696_8kHz.wav ”Some have walked through pain and sorrow to bring you their message of hope.”
si747_8kHz.wav ”Add remaining ingredients and bring to a boil.”
si788_8kHz.wav ”We have become amateur insurance experts and fine-feathered yard birds.”
si978_8kHz.wav ”For girls, the overprotection is far more pervasive.”
si1149_8kHz.wav ”The response of reaction is dominated by a concern for what is vanishing.”
si1434_8kHz.wav ”Conceivably the submarine defense problem can be solved by sufficient forces.”
si1621_8kHz.wav ”Now he'll choke for sure.”

The speech files spoken by male speakers:
si745_8kHz.wav ”Broil or toast as usual.”
si839_8kHz.wav ”Practically all forecasts mention new and exciting products on the horizon.”
si923_8kHz.wav ”To many experts, this trend was inevitable.”
si1010_8kHz.wav ”The so-called vegetable ivory is the hard endosperm of the egg-sized seed.”
si1024_8kHz.wav ”At the base of the rocky hillside, they left their horses and climbed on foot.”
si1039_8kHz.wav ”He has never, himself, done anything for which to be hated -- which of us has?”
si1175_8kHz.wav ”Diversity of perception, yes; diversity of fact, yes.”
si1230_8kHz.wav ”Princes and factions clashed in the open street and died on the open scaffold.”
si1364_8kHz.wav ”But within that framework he allowed for as much flexibility as possible.”
si1899_8kHz.wav ”You took me by surprise.”

3. Design and implementation of the codec

3.1 Selection of the parameter values

Use the following parameters in all LP computations (see the lecture slides):

- LP order: p=10
- Autocorrelation method with the Hamming window
- Frame length: 25 ms (no overlapping)
- FFT size in the spectrum computation: 1024

In the following, the transfer function of the LP analysis (which is a FIR filter) is referred to as A(z) and its inverse (which is an IIR filter) is referred to as 1/A(z).

3.2 LP analysis and synthesis without quantization

The goal of this sub-task is to make the student familiar with two basic concepts in LP-based speech coding and in LP-based processing of speech signals in general: (1) the LP analysis part, that is, FIR

filtering of the input speech signal with the LP analysis filter which results in the residual and (2) the LP synthesis part, that is, IIR filtering in which the residual signal is converted back to the input speech signal by filtering the residual with 1/A(z).

Steps:

- Using two input speech files ("si745_8kHz.wav" for a male sample, and "si747_8kHz.wav" for a female sample), compute LP-analysis using the parameters given in section 3.1. Note that you need to compute both LP analysis and LP synthesis in several frames because the length of the input signal is much longer than the LP frame. Therefore, you should update the delay line of A(z) correctly in the beginning of every frame to include the last *p* samples of the previous frame.
- Draw the original speech signal and the corresponding residual using the same amplitude scale over a time-window which spans the entire signal.
- Using the residual computed in the analysis stage, reconstruct the original signal by filtering the residual with 1/A(z) (i.e. compute LP synthesis). Note the following: (1) Be sure to update the delay line of 1/A(z) correctly in the beginning of every frame to include the last *p* samples of the previous frame. (2) Observe that since LP analysis computed with the autocorrelation criterion guarantees filter stability, IIR filtering should not result in oscillation (…provided that you have computed everything correctly).
- Draw the original speech signal and the computed output of the LP synthesis using the same amplitude scale over a time-window which spans the entire signal.

3.3 Quantization of the LP filter

In section 3.2, no information was quantized which means that no compression of speech information took place. As the first step towards implementing the codec, let us study the quantization of the LP synthesis filter. The effect of quantization on the spectrum of the LP synthesis filter can be objectively assessed using the measure called the spectral distortion (SD) (Paliwal and Atal, 1993). By denoting the power spectrum of the original, non-quantized LP synthesis filter as $Po(i)$ (where i is the discrete frequency variable of a N-size FFT) and the power spectrum of the corresponding quantized LP synthesis filter as $Pq(i)$, SD can be computed (in dB) using the FFT spectra as follows:

$$SD^2 = 1/(\frac{N}{2} + 1) \sum_{i=0}^{N/2} [10lg10Po(i) - 10lg10Pq(i)]^2$$

Steps:

- Convert the LP synthesis filters computed in section 3.2 to the lattice form. Quantize the lattice filter using uniform scalar quantization with 5, 4 and 3 bits per filter coefficient. Note: remember to design a quantization scale that keeps the quantized LP synthesis filter stable.
- Compute the SD measure for the three quantization schemes (i.e. 5, 4 and 3 bits) and report the average SD for wav files "si745_8kHz.wav" and "si747_8kHz.wav".
- By studying one frame from "si745_8kHz.wav" (select the frame in a place where the signal waveform is sustained and of high amplitude), draw the original lattice filter coefficients and those quantized with 3 bits in the same figure.
- Repeat the analysis/synthesis procedure described in section 3.2. using in LP analysis and synthesis direct form filters that have been converted from the 3-bit quantized lattice filters that you computed in above. What can you observe?

3.4 Implementation of the codec

In order to compress the bit-rate of the input speech signal, that is, in order to implement a real speech coding method, both the LP filter and the residual signal need to be quantized. In section 3.3, only the quantization of the former was studied, so let's quantise in this section also the residual. Note: Use the 3-bit quantization scheme computed in section 3.3. for the LP filter throughout this section.

Steps:

- In order to quantize the residual, let's take advantage of a simplified version of the approach which is used in the regular-pulse excited codec (Sluyter et al., 1988). In this scheme, the reduction of the bit-rate is achieved by *decimating* the original residual by factor r=3, that is, by reducing the sampling rate by keeping only every 3^{th} sample and by replacing the other samples by zeros (see Figure 1). The compression is achieved because zero-valued samples carry no information and they do not need to be quantized. Note that this simple idea cannot be used for the speech signal but it can be used for the noise-like residual. Note also that in order to avoid aliasing in downsampling (Oppenheim and Schafer, 1991), the residual needs to be low-pass filtered before the decimation. Based on this idea, design a residual quantization method which includes the following steps:
  (1) Design a short low-pass zero-phase FIR filter (cut-off frequency ca. 1300 Hz) and low-pass filter the residual. As a low-pass filter, you can use the 11-tap FIR filter whose impulse response is given in Table 1.
  (2) In each frame, normalise first the low-pass filtered residual by its maximum amplitude value. Information about this maximum amplitude needs to be known by the decoder, and therefore you need to quantize this information with a selected number of bits (e.g. 6 bits).
  (3) Build the four decimated residual sequences shown in Fig. 1 and select the one with the largest energy to represent the original residual. Note: The decoder needs to know which of the four sequences is selected in each frame. Therefore, you need to reserve a few bits to this information.
  (4) Quantize the non-zero samples of the selected decimation sequence using uniform scalar quantization (e.g. 3 bits per non-zero sample).
- Compute the exact bit-rate of the codec. Explain each part of the bit budget. Your aim is to select such number of bits in the above parts that you end up with a bit-rate of approximately 11 kbits/s**.**
- The above parts deal with the *encoder* (i.e. the transmitter) where the input speech signal of 128 kbit/s is expressed in a compressed form using the bit-rate of ca. 11 kbit/s. To generate the coded speech signal, you still need to build the *decoder* (i.e. the receiver). The decoder multiplexes the bit stream generated by the encoder by performing LP synthesis using the quantized information. In order to build the decoder, you need to do the following for every frame: (1) Reconstruct the LP residual using the information about the decimation sequence and the maximum amplitude. Add the missing zeros. (2) Reconstruct the LP synthesis filter using the corresponding bits. (3) Synthesise the speech signal frame-wise from the above parts.
- Using the designed implementation, encode all the 20 speech signals listed in section 2.1.

4. Subjective evaluation of the codec

Speech codecs need to be evaluated in order, for example, to compare different codecs. Speech codecs can be evaluated using several subjective test types (ITU-T P.800, 1996). A widely used test type in the evaluation of the quality given by a tested speech codec is the absolute category rating (ACR) test. The ACR test results in the scores known as the mean opinion score (MOS) (ITU-T P.800,

1996). In the ACR test, listeners evaluate individual spoken sentences encoded with the speech codec under evaluation. Listeners evaluate the quality on a discrete five-point scale from bad (1) to excellent (5). Small quality degradations can be better evaluated with a *pair comparison* test in which listeners compare processed speech samples to their reference counterparts (typically the original, input speech signal). The degradation category rating (DCR) test (ITUT P.800, 1996) is a comparison test which has been used in the evaluation of speech codecs. In the DCR test, the listener hears two sentences separated by approximately 0.5 s of silence. The first sentence is the original input speech signal and the second sentence is the coded speech signal. The listener is asked to evaluate the *quality degradation* of the second sample compared to the first one. The quality degradation is evaluated using a five-point degradation scale given below where the number on left is the score. The DCR test type offers higher sensitivity to minor degradations than the ACR test type.

5: Degradation is inaudible.

4: Degradation is audible but not annoying.

3: Degradation is slightly annoying.

2: Degradation is annoying.

1: Degradation is very annoying

Steps:

- Using the 20 input sentences and their coded counterparts produced in section 3.4, construct 20 test samples to run the DCR test for your codec.
- Conduct the above experiment using the members of your team as test subjects. Report the results as mean DCR scores for your codec. Note: In formal evaluations of speech codecs, the designers of the codec are not allowed to take part in the evaluations. In addition, codecs are tested using a much larger number of listeners.

5. Documentation

Combine all the tasks done in sections 3 and 4 in a document and add the code that you wrote as an appendix.

References:

3GPP, TS 26.090 (2001). Adaptive multi-rate (AMR) speech codec; Transcoding functions, 3GPP.

Garofolo, J. S., et al. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium.

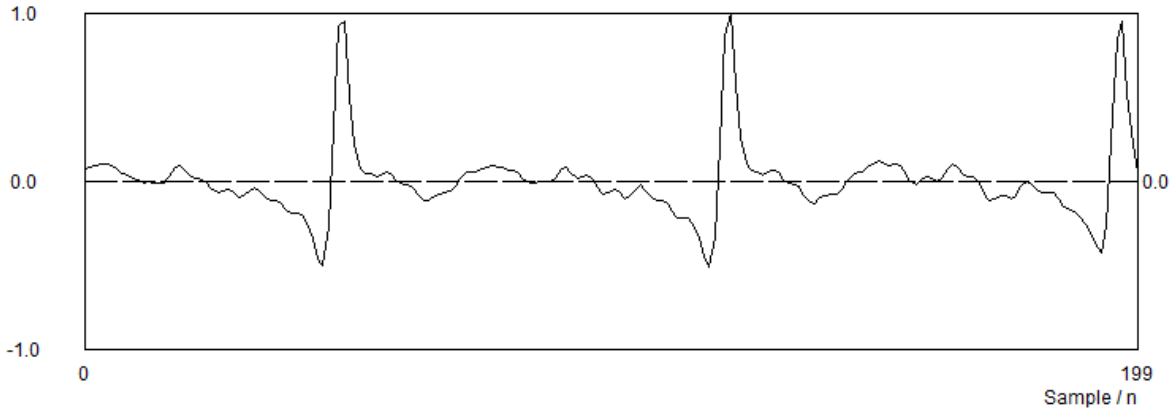
ITU-T P.800 (1996). Methods for subjective determination of transmission quality, International Telecommunications Union, Aug. 1996.

Oppenheim, A., Schafer, R. (1991). Discrete-time Signal Processing, Prentice Hall.

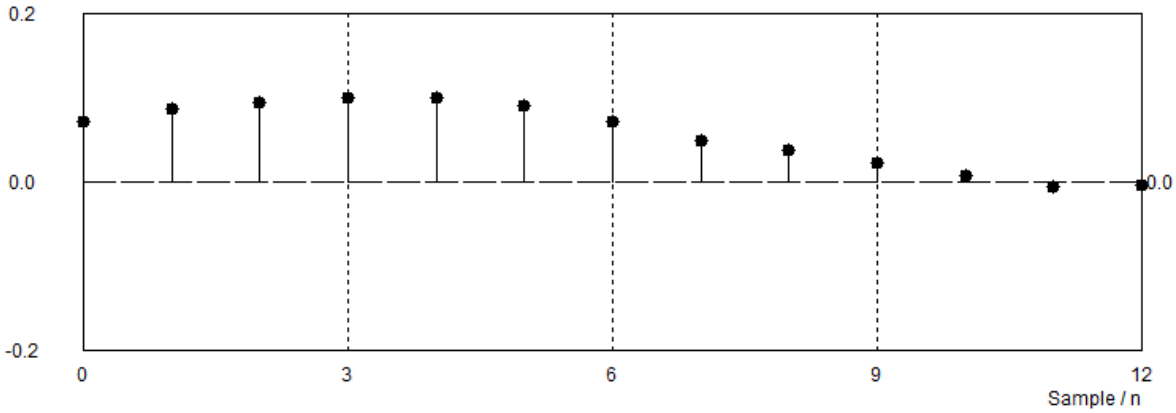
Paliwal, K., Atal, B. (1993). Efficient vector quantization of LPC parameters at 24 bits/frame. IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 1, pp. 3-14.

Sluyter, E., Vary, P., Hofmann, R., Hellwig, K. (1988). A regular-pulse excited linear predictive codec. Speech Communication, Vol. 7, No. 2, pp. 209-215.

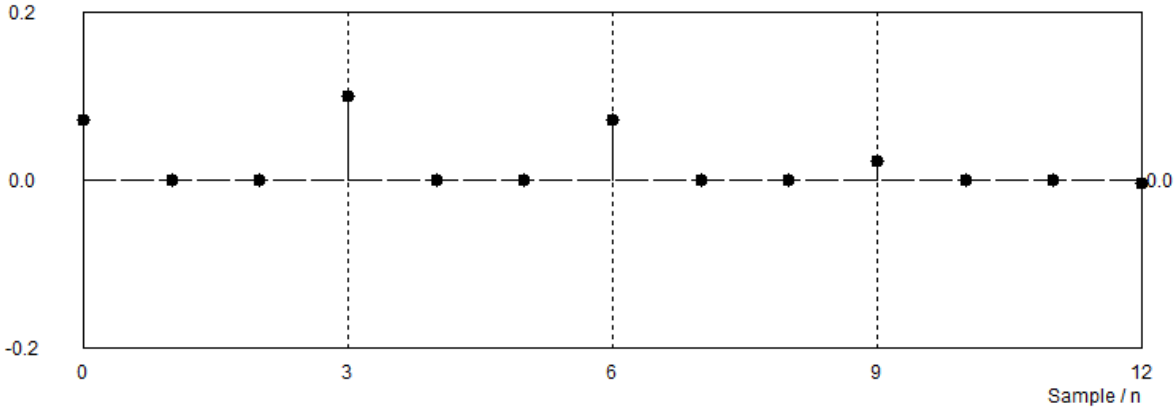
Vary, P., Martin, R. (2006). Digital Speech Transmission. Enhancement, Coding and Error Concealment. Wiley.



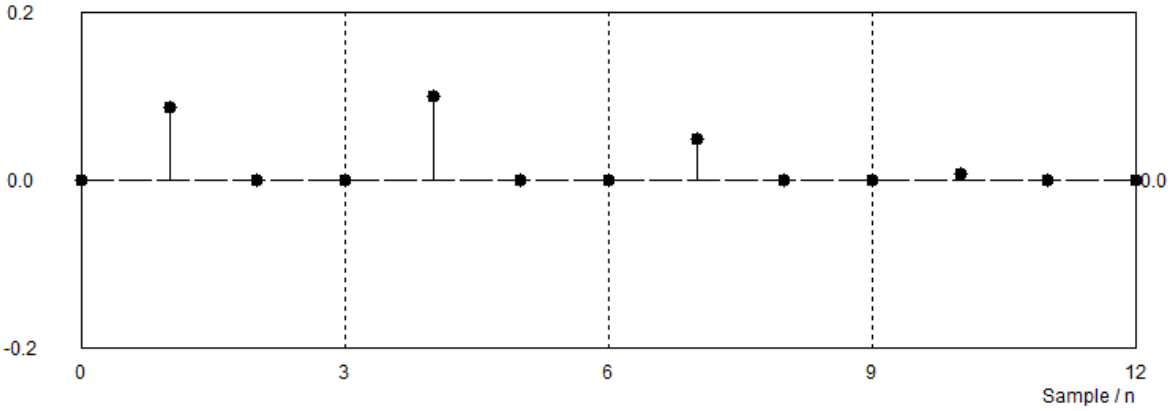
Residual (entire frame)



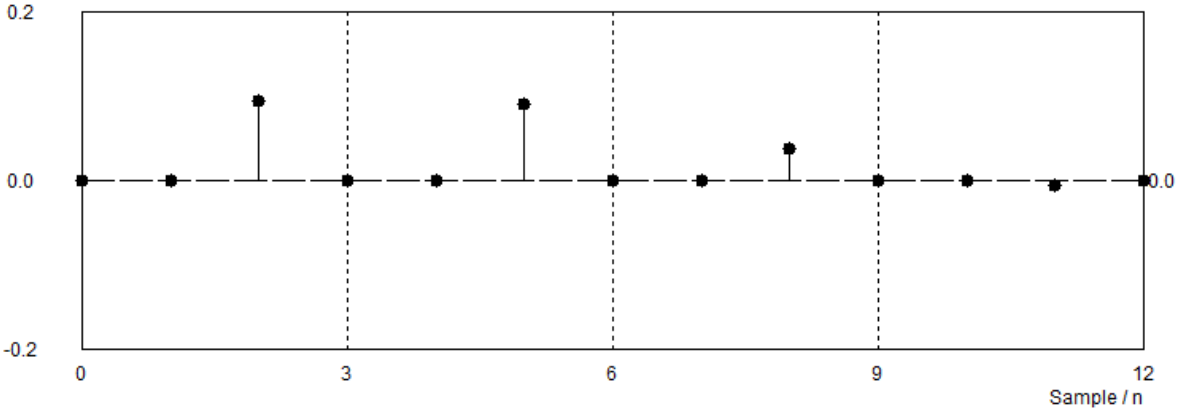
Residual (in the beginning of the frame)



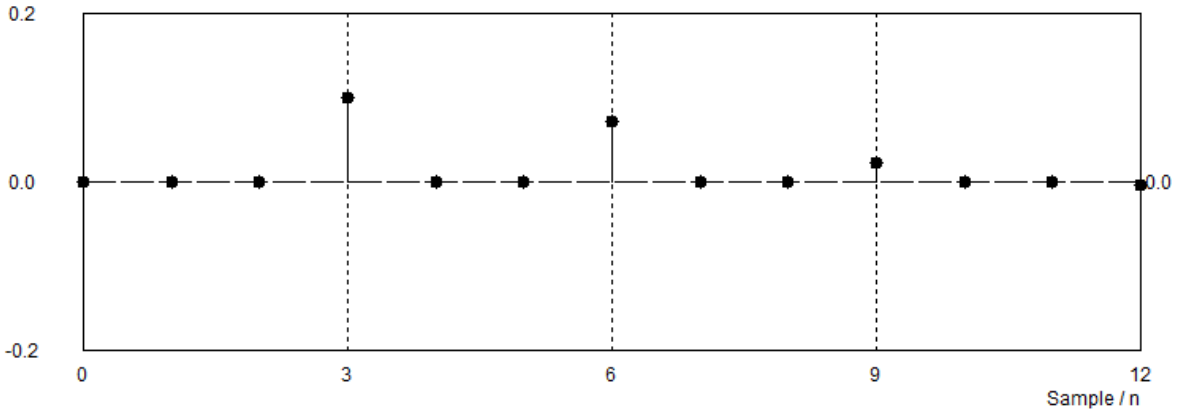
0^{th} decimated residual (in the beginning of the frame)



1st decimated residual (in the beginning of the frame)



2nd decimated residual (in the beginning of the frame)



3rd decimated residual (in the beginning of the frame)

Figure 1. Demonstration of the decimation of the residual. The first panel shows a residual signal (after antialising filtering) over a frame (index range from 0 to 199). The beginning of the same frame is show in the second panel (index range from 0 to 12). The four decimated sequences that are obtained by inserting zeros into the original residual are shown by the last four panels. Note: each decimated sequence has 66 non-zero samples during the frame (the last non-zero samples will be at indices 195, 196, 197 and 198, for the 0th, 1st, 2nd and 3rd sequence respectively).

Table 1. The impulse response of the antialising low-pass FIR to be used in decimation of the residual.

n	h(n)
-5	-0.016357422
-4	-0.045654297
-3	0.0
-2	0.25073242
-1	0.70080566
0	1.0
1	0.70080566
2	0.25073242
3	0.0
4	-0.045654297
5	-0.016357422