

# Speech coding

V 2.3, March 28, 2012

Martin Hagmüller, hagmueller@tugraz.at

Signal Processing and Speech Communication Laboratory

<http://www.spsc.tugraz.at/>

Inffeldgasse 16c/EG

Graz University of Technology

## Abstract

This part of the laboratory is dedicated to Speech Coding. We will study principles of speech coding as well as speech coding quality assessment. The experiments are performed in Matlab as well as in Praat.

### Equipment:

- PC with a sound card installed
- Matlab and Praat software, a copy of laboratory files (these you can download from our website  
<http://www.spsc.tugraz.at/courses/scl/download/coding.zip>)
- Headsets and necessary cables.

### Before you start...

1. You are expected to write a detailed report about your work. The report should be prepared on the fly, as you proceed with the tasks and handed over at the end of the laboratory.
2. Do not forget to answer the questions asked in experiments and those asked by the lab assistant. Your answers should be provided in full, with all the necessary explanations and, if required, plots and tables. Try to explain what you observe! If you can't, do not hesitate to ask your lab assistant!

## Experiment 1: Linear models and quantization

In speech coding, the main goal is to represent the speech signal as compactly as possible, preserving the audible quality. Speech modeling and proper quantization are the key elements in achieving this goal.

This part of the laboratory is to be performed in pairs such that one of the group prepares the speech material and another makes the measurement.

### (1.1) LPC residual signal

1. Start Praat and load any speech file from `speech_coding/data` directory into the Praat workspace. The given files are in RAW format, i.e. without any header information. To load the file follow `Read` → `Read from special sound file` → `Read from raw 16-bit Little Endian file...`. **Note:** `m38.raw` and `f68.raw` files are in German language while the others are in English.
2. For the raw sound file it is necessary to specify the sampling rate manually. Select `Modify -` → `Override Sampling frequency` and set *New Sampling rate* to 8000Hz.
3. Press `Formats & LPC` → `To LPC (autocorrelation)` → `Standards`. This will create a new LPC-object constructed from the selected signal.
4. Now, in the object browser select simultaneously the created LPC object and the corresponding sound signal. Press `Filter (inverse)`. The newly created object is the LPC residual signal (in Speech Analysis laboratory we used it as an approximation to the excitation signal).
5. Let the other person from the group listen to the created residual. Is it possible to understand what was said? Compare it with the original sound.
6. Now, exchange your roles and repeat the above experiment.
7. Can you tell why the LPC residual signal contains so much information?

### (1.2) Quantization

1. In this task we continue to use the same experiment strategy, i.e. one person loads the file and the other listens to the speech sample. Then the roles are exchanged.
2. Load any speech file from the `speech_coding/data` directory into the Praat workspace.
3. Select the corresponding sound object and press `Modify -` → `Formula...`. In the created dialog window specify the following expression: `if (self)>0 then 1 else -1 fi`. This operation results in a 1-bit quantization of the original sequence.
4. Now, let the other person listen to the created sound. Is it possible to understand what was said? Compare it with the original sound.
5. Repeat the experiment, but now exchange your roles.
6. What is your explanation of why the speech information is still audible?

## Experiment 2: ADPCM

This experiment is dedicated to the ADPCM (Adaptive Differential Pulse Code Modulation) coding scheme. ADPCM is an effective combination of the adaptive LPC modeling and quantization.

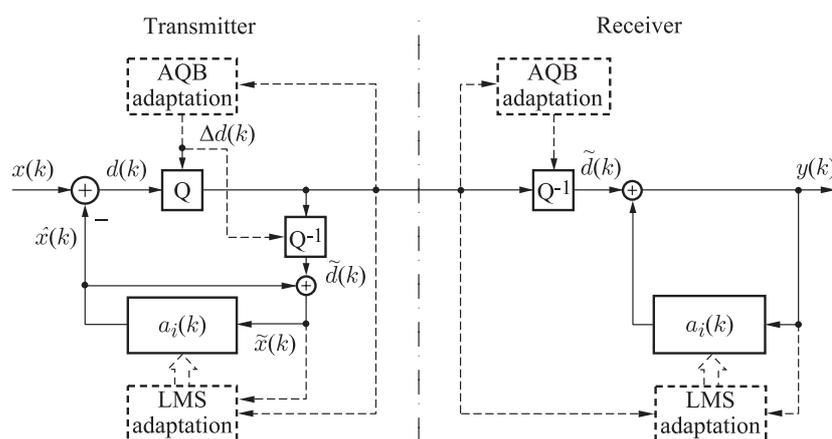


Figure 1: ADPCM speech coder

### (2.1) ADPCM coder

1. Start Matlab. Change the working directory to `speech_coding/adpcm` where you will find `in_data` directory with input speech files and `doadpcm.m` script that will be primarily used during the experiment.
2. Load `doadpcm.m` code into the Matlab editor. This script simply loads the specified speech files and invokes the coder `adpcm.m` with the proper settings. There are two important parameters that control the behavior of the script. The first one is the coder bitrate, specified in the variable `N` (possible bitrates are 16,24,32, and 40 kbps), and second parameter `ExtendedOutput` controls the number of output arguments in `adpcm.m` function. In case `ExtendedOutput` is different from 0 the `adpcm` returns most of the internal variables. Otherwise, only the coded index and decoded output signals are returned.
3. Set `ExtendedOutput` to 1 and run the script. For every loaded file listen to the original and the decoded signals. The corresponding speech samples are stored in the variables `x` and `y` respectively. Using the built-in Matlab function `sound` play the decoded and original samples. **Note:** in the Matlab implementation of `sound` function it is required for the signal to be scaled between  $\pm 1$ .
4. For each played sample plot the corresponding original sequence `x`, difference signal `d`, output of the coder `dQ`, and the difference between the original `x` and the reconstructed signal `y`.
5. Perform the above analysis for bitrates `N=16,24,32`, and 40kbps.

### Experiment 3: Objective coding quality criteria.

In this experiment we will study an objective quality measures to evaluate speech coding algorithms. The difficulty in defining objective quality measure lies in non-stationarity of the speech and strong speaker dependency.

#### (3.1)Segmental SNR

1. Using `segsnr.m` script from `speech_coding/quality` directory estimate the segmental SNR for the ADPCM coder from the previous experiment.
2. Run the `doadpcm.m` script and wait until the program enters the debug mode. The original signal `x` and the decoded signal `y` are stored in the workspace.
3. Fill in the following table:

L msec / bitrate	16kbps	24kbps	32kbps	40kbps
5 msec				
15 msec				
30 msec				
60 msec				

**Note:** Before computing the SNR make sure that both the decoded signal `y` and the original signal `x` have the same length and time aligned.

### Experiment 4: Subjective coding quality criteria.

Subjective criteria, on the other hand, are measured by a group of listeners that give their personal evaluations of the encoder quality. We will consider two of them – Modulated Noise Reference Unit (MNRU) and Mean Opinion Score (MOS). We will also study some speech coders that are used nowadays in digital communication systems.

#### (4.1)Modulated Noise Reference Unit

1. Just as in the previous experiment, run ADPCM encoding for the given speech signals.
2. For the original signal `x` generate the corresponding equivalent noise signal `x_eq` using `mnrnru.m` script from `speech_coding/quality` directory. Set noise parameter `Q` between 10 and 40dB.
3. Compare perceptually the ADPCM decoded signal with the equivalent noisy signal `x_eq`. The value of `Q` that results in a similar speech quality is used as a quality measure.
4. Fill in the following table:

	16kbps	24kbps	32kbps	40kbps
Q, dB				

**(4.2)Computing MOS**

1. If necessary, start Matlab and load `encode_decode.m` script from `speech_coding/quality/mos` directory. This script automatically encodes and decodes the specified speech signal with the specified encoder. It requires a single parameter – the index  $N$  of the coder as shown in Table 1.

N	Coder	Listener 1	Listener 2	MOS	Total MOS
1	AMR 4.75 kbps				
2	AMR 12.2 kbps				
3	G722 6.6 kbps				
4	G722 23.8 kbps				
5	G729 6.4 kbps				
6	G729 11.8 kbps				

Table 1: MOS

2. Now, let one of the group choose a coder at random by specifying the corresponding coder index. Run the script and, when prompted, enter the file name of a speech file from `speech_coding/quality/mos/samples/` directory. When the script enters the debug mode, play both the original and the decoded speech signals to you partner. **Note:** the partner shouldn't know what coder is used and what sound file is chosen. He only has to judge to quality of the speech sample using the scale shown in Table 2. Once

Score	Description
5 – excellent	Imperceptible difference
4 – good	Just perceptible but not annoying
3 – fair	Perceptible and slightly annoying
2 – poor	Annoying but not objectable
1 – unsatisfactory	Very annoying and objectable

Table 2: MOS grade system

you finished, exchange you roles.

3. Repeat the above steps until the Table 1 is filed. The MOS column of the table is computed as the mean value of the obtained scores for each coder.
4. The total MOS is computed as a mean score for the corresponding coders among the groups, i.e. individual MOS scores are averaged together.