PROJECT INSTRUCTIONS

1) This is an **team** project, teams can be composed of $2-4$ students.
2) All team members are accountable for all project parts.
3) Team reports (including source codes, figures or comments) are not to be shared with others, neither before nor after submission. However, in person discussions are encouraged.
4) Any copied reports, either fully or partially, will receive 0 points. This applies to both the original and the copy.
5) Submission is by the due date. **Late submission** is allowed for 10% deduction per day (or part of a day), for a maximum of 5 days.
6) In submission, you have to submit **.m files** separately. In addition, the figure should be submitted in **.fig format** and should be **included** in the **.pdf report**. Reports should be comprehensive and readable on their own.
7) The **.pdf report** is the main document to be evaluated, *i.e.* no credit is given for the source codes. However, source codes are to be checked against plagiarism.
8) Grading will depend on:
   - **60%**: Completeness and correctness of every deliverable (as per all the required files).
   - **20%**: Clarity of figures, and proper labeling (as per the .pdf report).
   - **20%**: Report writing and organization.

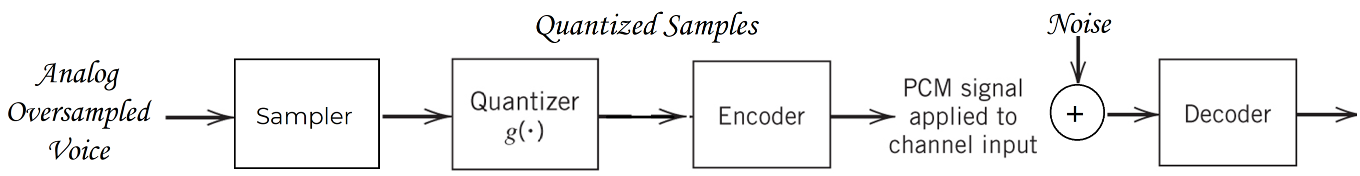
PROJECT PRESENTATION

Prepare a 15-minutes live presentation on the project and its results. The presentation has a separate evaluation/grade from the project and its report.
This presentation should include a Demo during which you enter various parameters, as detailed in the project description, and show the outputs as instructed by the evaluators.
The grading criteria of the **presentation** will be as follows:

- **25%:** Comprehensiveness and clarity of the presentation.
- **50%:** Answering questions and successfully demonstrating the outputs of the project.
- **25%:** Personal presentation skills.

PROJECT DESCRIPTION: VOICE CODECS

Consider the system shown below.



PCM-Based Vocoder/Decoder

You are required to **write a voice codec software** based on PCM of voice signals to send voice over data networks.

Codec Description

You need to write a function for each of the system blocks as follows:

1) The **Sampler** function, with the required sampling frequency as input.
2) The **Quantizer** function should have the option that **the user chooses between:**
   a) **Mid-rise Uniform quantizer**
   b) **Mid-tread Uniform quantizer**

   For each, the user specifies the number of levels, L, the peak quantization level, m_p.
   The function should **allow the user to input a signal** to be quantized. That signal will be in the form of two vectors, a time vector and an amplitude vector.

   This function should also result in the following:
   a) A figure showing the input signal and the quantized signal, on the same plot, with proper legend.
      Note: Display the input signal as a continuous signal, and display the quantized signal as a continuous staircase signal.
   b) The value of the mean square quantization error, i.e. $\mathcal{E}\{(m-\nu)^2\}$.
   c) A stream of bits representing the quantized signal.
3) The **Encoder** function is required to represent the bit stream resulting from the quantizer as a signal. This function should have the option that **the user chooses between:**
   a) **Unipolar NRZ Signaling**
   b) **Polar NRZ Signaling**

   For each, the user specifies the pulse amplitude and the bit duration.
   Note that the bit duration is related to the sampling rate and the number of bits of the **Quantizer**.
4) The **Decoder** function is required to transform received PCM coded pulses into a stream of bits, then transform each $\log_2(L)$ bits into a quantized sample.
   This function should result in an output plot of the quantized samples.
   This function should have parameters matching those of the **Quantizer** and**Encoder** functions.

Testing your Voice Codec: Test 1

You are required to test your voice codec as follows:

1) Import an audio file to your program. This represents the audio to be digitized and transmitted. The file should be at least a 5 seconds audio file with CD quality.
2) Show the output of each of the 4 functions implemented.
   Use the following parameters for your test:
   - Sampling frequency: $f_s = 20, 10$ and 5 KHz.
   - Quantizer peak level: Based on the input signal.
   - Quantizer number of levels: $L = 16, 64$ and 256.
   - Encoder: Both Unipolar & Polar NRZ signaling, showing only the first 10 bits of the bit stream.
3) Save the output of the **Decoder** as a new audio file. This represents the received digitized audio file.

Testing your Voice Codec: Test 2

Repeat the previous test in the presence of noise as follows:

1) The **Encoder** function results in pulses with amplitude $A = 1$ volt. Test both Unipolar and Polar signaling.
   For the other functions, use the following parameters for your test:
   - Sampling frequency: $f_s = 10$ KHz.
   - Quantizer peak level: Based on the input signal.
   - Quantizer number of levels: 64.
2) Generate channel noise as AWGN $\sim \mathcal{N}(0, N_0)$.
3) Implement a suitable **Regenerative** function that precedes the **Decoder**, with an input of noisy PCM coded pulses and an output of regenerated PCM pulses.
4) Save the output of the **Decoder** as a new audio file.
5) Perform this test for $N_0 = 0.5, 1, 3$ and 5.

DELIVERABLE

Deliver the following in a .zip file

1) Source codes (.m files) of functions and main files.
   This will be used to test your system with arbitrary parameters and for arbitrary input audio file.
2) The test audio file.
3) The output audio file of Test 1, for various combinations of the parameter values.
4) Comment on your observations of the audio quality based on the parameters used.
5) The output audio file of Test 2, for various values of the noise PSD, N_0.
6) Comment on your observations of the audio quality for each of the values of N_0.
7) A single .pdf project report with a cover page.