Communications and Information Engineering Program
Communications Theory and Systems
CIE 337 - Spring 2021
Term Project - Part II

General Instructions & Grading Criteria

Instructions

1) This is a **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) No late submissions are allowed.
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.

Grading Criteria

Grading of each part will depend on:

- **40%:** Completeness and correctness of every deliverable (as per the .pdf report).
- **40%:** Clarity of figures, and proper labeling (as per the .pdf report).
- **20%:** Report writing and organization.

PART A: PCM QUANTIZATION

Consider the system shown in Fig. 1

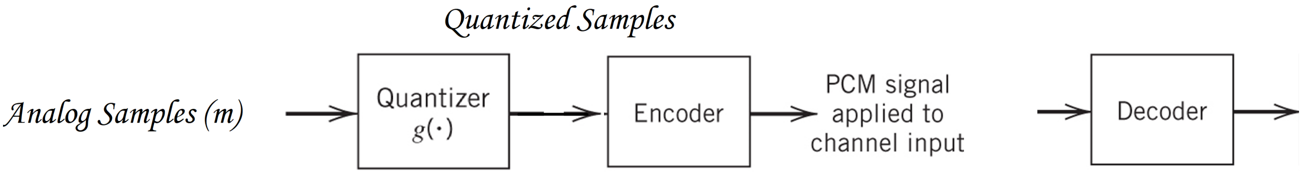

Fig. 1: PCM System with Discrete Input

You are required to **write software programs** and **implement a GUI** to help users **quantize**[1] an analog discrete signal.

GUI Description

Your **Quantizer** function and GUI should have the option that **the user chooses between:**

1) **Uniform quantizer**, where the user specifies the number of levels, L, the peak quantization level, m_p, and whether the quantizer in mid-rise or mid-tread
2) **Non-Uniform μ-Law quantizer**, where the user specifies μ, L and m_p

The GUI 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.

The GUI should also display the following:

1) 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.
2) The value of the mean square quantization error, i.e. $\mathcal{E}\{(m-\nu)^2\}$

[1] In this part, you will implement the Quantizer only

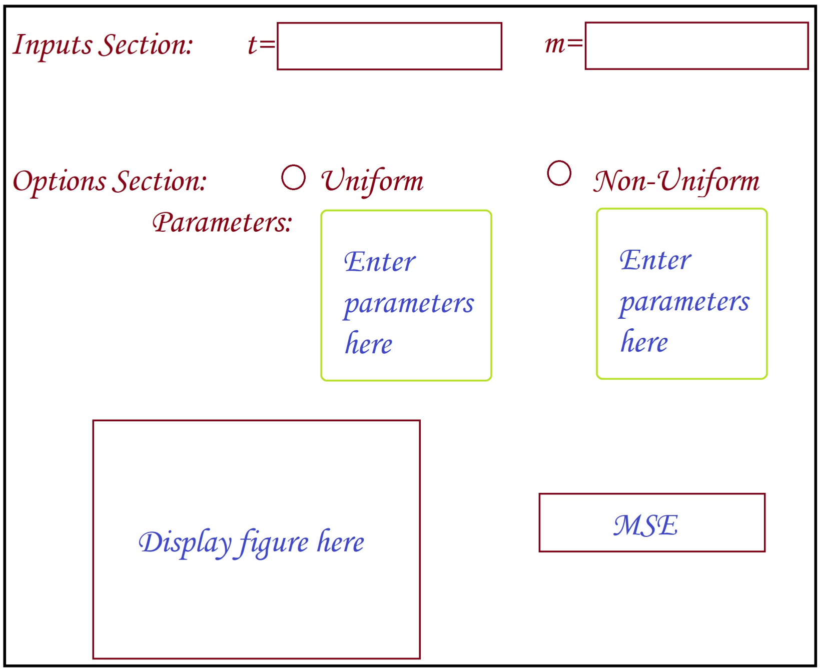

Fig. 2: Sample GUI

Testing your Simulator and GUI

Test your quantizer GUI for the input signal $m[k]$ for the following cases

$$m[k] = 5\cos(2\pi f_m k)\,, \qquad \text{where } f_m = 10\ Hz, \quad \text{for one complete cycle of the signal}$$

	Case 1	Case 2	Case 3	Case 4
Sampling Frequency	$f_s = 40\ Hz$	$f_s = 20\ Hz$	$f_s = 15\ Hz$	$f_s = 20\ Hz$
Quantizer	$\mu = 0, L = 8, m_p = 5$	$\mu = 0, L = 32, m_p = 5$	$\mu = 0, L = 16, m_p = 5$	$\mu = 100, L = 32, m_p = 5$

Deliverable - Part A

Deliver the following in a .zip file

1) The GUI files.
   This will be used to test your system with arbitrary parameters and for arbitrary input signals
2) Source codes (.m files) of functions and main files.
3) Source code of main script used for the 4 test cases.
4) Screenshots of the GUI's output for the 4 test cases.
5) For each of the 4 cases, make a brief comment on your findings
6) A single .pdf project report with a cover page.

PART B: LINE CODES ENCODER

You are required to **write software programs** and **implement a GUI** to help users **encode** a quantized discrete signal.

GUI Description

Your **Encoder** function and GUI should have the option that **the user chooses between:**

1) **Unipolar NRZ** signaling
2) **Polar RZ** signaling
3) **Alternate mark inversion (AMI)** signaling
4) **Manchester** signaling

The GUI should **allow the user to input** a stream of bits to be encoded. The GUI can also give the user the option to generate N random bits, where N is determined by the user.

The GUI should also:

1) Give the user **the option** to determine T_b, as well as the signal amplitude, A.
2) **Display** the output line code, with proper labeling according to the user choices.

Testing your Simulator and GUI

Test your Encoder GUI for an input stream of bits, generated as follows:

$$\nu = \texttt{randi([0~~1],1,10)}$$

	Case 1	**Case 2**	**Case 3**	**Case 4**
Encoder	Unipolar NRZ	Polar RZ	AMI	Manchester

Deliverable - Part B

Deliver the following in a .zip file

1) The GUI files.
   This will be used to test your system with arbitrary parameters and for arbitrary input bit streams
2) Source codes (.m files) of functions and main files.
3) Source code of main script used for the 4 test cases.
4) Screenshots of the GUI's output for the 4 test cases.
5) A single .pdf project report with a cover page.