Assignment 4

EC5.201 Signal Processing

Sampling & Quantisation

Deadline: Saturday, 13th Sep '25, 11.55PM IST

Instructions:

- All the questions are compulsory
- The questions contain both theory (analytical) and practical (MATLAB coding) parts
- The submission format is as follows:
  - **Theory (on Moodle)** – PDF file containing the handwritten theory assignment solutions
  - **Lab (on GitHub)**
    - **Code** – folder containing all the codes
    - **Images** – folder containing all the .png images
    - **Report** – PDF/text file containing observations for the MATLAB section

    The naming convention for the code and image files is q<qn no.> _ <sub-part no.>.
- **Late submission:** For the theory component, a 10% penalty per day will be applicable (accepted up to at most 3 days after the deadline). No late submissions will be accepted for the lab component.

Question 1: Sampling for non-band-limited signals

MATLAB:

We know that sampling can be applied only for band-limited signals. Let us consider a non-band-limited signal and investigate its reconstruction as sampling interval `Ts` is changed.

Consider the continuous-time triangular pulse signal of height 1, base in the interval $[-1, 1]$, and zero otherwise. Use an appropriate `t_samples` vector (for given `Ts`) such that all the samples lie within the base of the triangle (assuming there is a sample starting at `t = -1`).

(a) Generate and plot the corresponding samples `x[n]` (using the `stem()` command).

(b) Perform band-limited interpolation for this signal using the samples generated for the four cases (i) `Ts = 0.5s`, (ii) `Ts = 0.2s`, (iii) `Ts = 0.1s`, and (iv) `Ts = 0.05s`. For reconstruction, use `t_fine = -10:0.001:10`. You may utilise the MATLAB function defined in the previous assignment.

(c) Create a 2×2 plot, one panel for each value of `Ts`. In each panel, plot the samples and the corresponding reconstructed signal. What do you observe as `Ts` is changed? Write clearly in your report.

Question 2: Quantisation

MATLAB:

Quantisation involves discretisation and encoding of the samples of a continuous-valued signal. The quantisation function $Q(\cdot)$ maps any real input to a point from a discrete set of values. For a sampled signal $x[n]$, quantisation and quantisation error are given by:

$$x_q[n] = Q(x[n])$$
$$e_q[n] = x[n] - x_q[n]$$

There are numerous quantisation functions used in practice. Here, we look at the *quadratic non-uniform quantiser*, whose working is described below. Define a function `quadratic_quant` as follows,

```
function y = quadratic_quant(x, B, a)
    % quantises the samples of a given signal
    %
    % x - input signal to be sampled
    % B - number of bits to decide quantisation levels (positive integer)
    % a - positive real number for the range of quantisation [-a, a)
    %
    % y - quantised version of the input signal

    %% code goes here
end
```

→ For the range $[0, a)$: Divide the interval $[0, 1]$ into $L = 2^{B-1}$ equal-sized intervals. Let $0 = r_0 < r_1 < \cdots < r_{L-1} < r_L = 1$ be the edge points of these intervals. Then the quantiser maps values in the interval $[ar_i^2, ar_{i+1}^2)$ to its mid-point.

→ Repeat the same symmetrically for the range $[-a, 0)$. Make sure the quantiser has a total of 2^B levels in the interval $[-a, a)$.

→ For inputs outside the interval $[-a, a)$, quantise them to the end points of your quantized set of values.

(a) Consider the analog signal $x(t) = \sin(2\pi f_0 t)$ with $f_0 = 10Hz$. Take the signal in the time interval $t \in [0, 1]$ and sample at a sampling frequency $F_s = 5kHz$ to obtain $x[n]$. Use the above function to obtain the quantised signal $x_q[n]$ (use $a = 1$ and $B = 4$).

(b) Make a 3×1 grid and plot the sampled signal and the quantised signal in the first two panels respectively. Compute and plot the quantisation error in the third panel.

(c) Use the `histogram()` function from MATLAB to plot a histogram of the quantisation error with 15 bins.

(d) Repeat the same for `B = 3` and generate the corresponding histogram. Compare with the histogram in (c) and note your observations in the report.

(e) Repeat the quantisation process for `B = 1:8` (do not generate the histograms) and compute the maximum absolute quantization error (MAQE) (over the complete signal duration) for each case. Plot a graph with `B` on the x-axis and MAQE on the y-axis. Comment on your observations in the report.

(f) The signal to quantisation noise ratio (SQNR) is defined as the ratio of the signal power to the quantisation noise power:

$$SQNR = \frac{\sum_n |x[n]|^2}{\sum_n |e_q[n]|^2}$$

Plot a graph with `B` on the x-axis and SQNR (for quantisation performed over `B = 1:8` above) on the y-axis. Comment on your observations in the report.

Question 3: Digital storage

An audio compact disc (CD) stores audio signals sampled at 44.1 kHz and quantized using 32 bits. Assume there are two audio channels. Answer the following with explanations:

(a) If the CD contains 7 songs of duration 5 minutes each, what is the amount of memory used up in storing the files in the CD?

(b) If it is known that the stored audio signals have a maximum frequency of 12 kHz, could you suggest a way to reduce the amount of memory used in the CD without any loss in audio quality? How much memory can you save?

Question 4: Continuous-time to discrete-time

An LTI system with impulse response $h(t)$ produces output $y(t) = h(t) \star x(t)$ when the input is $x(t)$, here $\star$ denotes continuous-time convolution. The bandlimited signal $x(t)$ is sampled at rate $f_s = \frac{1}{T_s}$ (which is above its Nyquist rate) to obtain the discrete-time signal $x[n] = x(nT_s)$.

(a) Show that $y(t)$ can also be sampled at rate f_s without any loss of information.

(b) Let the samples be $y[n] = y(nT_s)$. Find the impulse response $h[n]$ of a discrete-time LTI system such that $y[n] = h[n] \star x[n]$, here $\star$ denotes discrete-time convolution. Justify whether this impulse response is unique.