Assignment 1, due: Mar. 2, 2026 (in class)

Note: Late submissions will not be accepted.

1. Read relevant chapters/sections of the textbook (or other recommended books), study carefully all examples, attempt some end of chapter problems. Extra references are given below.

2. Find the energy and energy spectral density (ESD) of the following pulse $p(t) = A\Pi((t-1/2)/T_s)$, where $\Pi(t)$ is the standard rectangular pulse, see Lec. 3. Plot $p(t)$ and its ESD for $A = 1$ and $T_s = 1\mu s$. Find the bandwidth of $p(t)$ (use the definition of bandwidth appropriate for this signal). What is the impact of A and T_s on the energy/ESD and bandwidth, e.g. what happens if one of these quantities is doubled?

3. In this question, we use numerical simulations to validate analytical results of Question 2. In particular, we use fast Fourier transform (FFT)[1] to find the ESD numerically in e.g. Matlab (it has FFT function already implemented). FFT replaces continuous-time Fourier transform with discrete Fourier transform (DFT) [1][3][5]. Specifically, to use FFT, we replace the continuous-time signal $p(t)$ with discrete-time sequence $\{p_n\}$:

$$p_n = p(t_n),\ t_n = n\Delta t,\ n = 0...N-1 \tag{1}$$

and the Fourier integral with the Riemann sum:

$$S(f) = \int_{-\infty}^{\infty} p(t)e^{-j\omega t}dt \ \rightarrow\ S_N(f) = \sum_{n=0}^{N} p_n e^{-j\omega t_n}\Delta t \tag{2}$$

and evaluate it at discrete frequencies $f_m = m\Delta f$, $m = 0...N-1$, so that $S(f)$ is replaced by $s_m = S_N(f_m)$:

$$s_m = \Delta t \sum_n p_n e^{-j2\pi f_m t_n} = \Delta t \sum_{n=0}^{N} p_n e^{-j\frac{2\pi}{N}mn} \triangleq \Delta t \cdot \mathrm{FFT}\{p_n\} \tag{3}$$

Note that Δt is usually not included in numerical implementations of the FFT algorithm, e.g. as in Matlab `fft` function, so you have to add it manually. It is strongly recommended that you study carefully the first example ("noisy signal") in [2] before implementing FFT to avoid possible mistakes (there is a very detailed discussion there).

To get accurate results, we have to select properly the time resolution (sampling interval) Δt and frequency resolution $\Delta f = 1/T$, where $T > T_s$ is the duration of the simulated time interval $[0, T]$, so that the numerical ESD is accurate and agrees well with the analytical one from Question 2. As a rule of thumb, $T \geq 10T_s$ and $\Delta t \leq T_s/10$ so that the total number of samples $N = T/\Delta t \geq 100$ has to be large enough.

[1] FFT algorithm was included in 10 most important algorithms discovered in 20th century [7][8].

To validate your numerical implementation of Fourier transform, use inverse FFT (IFFT) to recover p_n from s_m via $\widehat{x}_n = \text{IFFT}\{s_m\}$. Plot $\widehat{x}_n$ and x_n in the same figure. Do they agree with each other?

Play with N, T and determine their appropriate values, as well as the respective values of $\Delta t, \Delta f$ so that the analytical and numerical ESD agree well with each other. Since FFT is implemented in a very efficient way, you can easily use large N on a modern computer. Clearly indicate the values of $\Delta t, \Delta f, N, T$ you are using as appropriate. Plot the analytical and numerical ESD in the same figure for $A = 1$ and $T_s = 1\mu s$. What is the bandwidth of $p(t)$ from numerical ESD? Does it agree with analytical one?

Using numerical ESD, evaluate the (total) energy in frequency domain and compare it to that in the time domain, thus validating numerically Rayleigh energy theorem.

While comparing numerical and analytical ESD, keep in mind that sampling in time domain introduces periodicity in frequency domain, with the period = sampling frequency = $1/\Delta t$ so that negative frequencies are shifted to positive ones exceeding $1/(2\Delta t)$, see e.g. [1], and therefore $s_{-m} = s_{N-m}$ (this can be also verified from (3)).

Good references on FFT and its numerical implementation/use are [1]-[6]. Note that [1][2] include Matlab examples, which can be helpful.

4. Now consider 2-PAM with random bits encoded into amplitudes $a = \pm A$, all iid and of equal probability:

$$x(t) = \sum_{k=0}^{K-1} a_k p(t - kT_s), \tag{4}$$

where $p(t)$ is from Question 2. In this Question, we will use the average power spectral density (PSD) defined over the transmission time interval $[0, KT_s]$ via ESD:

$$\text{PSD} = \frac{1}{KT_s}\text{ESD} \tag{5}$$

(a) Find analytically the PSD of this random 2-PAM signal (where $a_k = \pm A$ are iid random variables and $\pm A$ have equal probabilities) and plot it for $A = 1$, $T_s = 1\mu s, K = 5$. What is the bandwidth of $x(t)$? Comment on the impact of A, T_s, K on the PSD and bandwidth (i.e. what happens if each quantity is doubled?).

(b) Generate 3 random realizations of $x(t)$ (corresponding to random realizations of a_k) and plot them. Comment on the impact of randomness.

(c) For each of 3 random realizations of the previous part, find numerically its PSD via FFT as in Question 3 (this is "instantaneous" PSD) and plot them along with the analytical PSD for comparison (you should have 3 figures, each having a numerical and analytical PSD). For this part, select appropriate simulation time interval $[0, T]$ so that the whole PAM sequence fits in and there is some margin on the right side, and also select appropriate N so that $\Delta t = T/N$ is sufficiently small. Do the numerical and analytical PSD agree with each other? Why? Using the numerically-found PSD, compute the (total) power in frequency domain and compare it to that in the time domain. Do they agree with each other?

(d) Repeat previous part by increasing K to $K = 10$. What is the impact of this increase on the agreement between numerical and analytical PSDs?

(e) Finally, generate many random realizations of $x(t)$, find numerically PSD for each, and then find the average PSD (across all realizations; this is called "sample average"). Compare it with the analytical PSD by plotting them together and find the bandwidth for each. Do they agree with each other? Why? How many random realizations of $x(t)$ do you need for the numerical and analytical PSD to agree with each other?

For all relevant questions, please include Matlab codes in your submission.

References

[1] B.P.Lathi, Z. Ding, Modern Digital and Analog Communication Systems, Oxford University Press, 2009. Ch. 3.9, 3.10.

[2] https://www.mathworks.com/help/matlab/ref/fft.html

[3] https://www.mathworks.com/help/matlab/fourier-analysis-and-filtering.html

[4] https://en.wikipedia.org/wiki/Riemann_sum

[5] https://en.wikipedia.org/wiki/Discrete_Fourier_transform

[6] https://en.wikipedia.org/wiki/Fast_Fourier_transform

[7] J. Dongarra, F. Sullivan, Guest Editors' Introduction to the Top 10 Algorithms, Computing in Science & Engineering, Jan. 2000.

[8] A. Townsend, The Top 10 Algorithms from The 20th Century, Cornell University (available online).

Note on plagiarism

Plagiarism (presenting somebody's else solution/report as your own, where "somebody" means any source, including Internet (Google, ChatGPT, etc.), "cut-and-paste" from a student to a student, other forms of "borrowing" the material) is absolutely unacceptable and will be penalized. Each student must submit his own solutions. If two (or more) identical or almost identical sets of solutions are found (including those from Internet), each student involved receives 0 (zero) for that particular assignment. If this happens twice, the students involved receive 0 (zero) for the entire assignment component of the course in the marking scheme and the case will be send to the Dean's office for further investigation and further penalties.