ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Electrical and Computer Engineering

Telecommunication Systems 2 Assignment (Part 1)

Course Instructor

Prof. Georgios Karystinos

Assignment Supervisor

Ioannis-Leonidas Steiakakis

`isteiakakis@tuc.gr`

Introductory Notes

For the programming parts of the assignment, the accepted programming languages are Python 3 (as either scripts or Jupyter notebooks) and MATLAB.

For tasks requiring code implementation, name the script file for each task using its sequential number. For example, if Task 9.z is a task requiring code implementation, use `task_9z.`⟨`py`|`ipynb`|`m`⟩ as the file name of the script file for that task.

For a task requiring code implementation, you may copy-paste a script from a previous task and use it as a starting point.

Try to follow good programming practices.

- You may refactor code into functions for modularity and logical abstraction for a cleaner code in the main script file of a task.
- Please use appropriate comments: clear and explanatory, but not overly verbose or unnecessary (e.g., avoid commenting on self-explanatory function calls and easily understandable lines of code).

Additionally, organize your project by creating subdirectories within the master directory for each section of the assignment. It is not a problem if two of those subdirectories contain identical function files.

In code simulation tasks, the indicative values given for parameters are just indicative. Do not hardcode them. Your code should be parameterized by those parameters and should work for other (appropriate) values, too.

You may use an immediate result (i.e., without proof) directly from the lecture notes, provided you appropriately reference[1] it, of course. Make sure to adjust the notation to match that of the assignment prompt and your deliverable report.

Important! Any non-trivial concepts, whether in theory or code, must be thoroughly explained. For theoretical aspects, provide clear reasoning and justification. For code, ensure that your implementation is well-commented and reflects the underlying theory. Theoretical ideas applied to your code should also be clearly justified and well-documented. Lack of proper explanation may lead to point deductions.

Finally, for any questions, I am at your disposal, by all means.

[1]That is, clearly indicate the source of the result — for example, which document it comes from, the page or equation number if applicable, and any additional context needed for clarity.

Contents

1 Preliminaries

Completing the tasks in this section is a requirement for proceeding to the rest of the assignment.

Important rules to follow. For the whole assignment (both part 1 and part 2):

- Produce the vectors of the x-axes for convolution and FFT manually; do not use any built-in functions for such cases (except `fftshift` which can be used as long as you understand what it does and why it does it).
  - Implement your own functions `convxaxis` and `fftxaxis` for these cases for reusability.
- Do not use utilities like the option `'same'` in the convolution function which changes the size of the result vector.
- Do not use any built-in functions for passing a signal through a system, do it via convolution.
- For the generation of random samples, use only the very basic functions, like `randn` for Gaussian, `rand` for continuous uniform, and `randint`/`randi` for discrete uniform.

Reminders:

$$\operatorname{sinc}(t) \triangleq \begin{cases} 1 & \text{if } \ t = 0 \\ \dfrac{\sin(\pi t)}{\pi t} & \text{otherwise} \end{cases}$$

$$\operatorname{RC}(f) \triangleq \begin{cases} T & \text{if } \ |f| \le \dfrac{1-\alpha}{2T} \\ \dfrac{T}{2}\left(1 + \cos\left[\dfrac{\pi T}{\alpha}\left(|f| - \dfrac{1-\alpha}{2T}\right)\right]\right) & \text{if } \ \dfrac{1-\alpha}{2T} < |f| \le \dfrac{1+\alpha}{2T} \\ 0 & \text{if } \ |f| > \dfrac{1+\alpha}{2T} \end{cases}$$

$$\operatorname{srrc}(t) \triangleq \mathcal{F}^{-1}\left\{\sqrt{\operatorname{RC}(f)}\right\}$$

Important things to remember. When a continuous signal x is simulated through its samples x_k, for $k \in \mathbb{Z}$, with a sampling period $T_\text{s} > 0$, i.e., $x_k = x(kT_\text{s})$, do not forget:

- The factor T_s when performing convolution and FFT.
- The factor $1/T_\text{s}$ when creating the FFT x-axis.
- The factor $1/T_\text{s}$ in the continuous variance to get the discrete variance if x is a white noise stochastic process.
  - In particular, since the Dirac delta function can be approximated by a rectangular pulse with unit area and height $1/T_\text{s}$, i.e., $\delta(t) \approx \frac{1}{T_\text{s}}\Pi\left(\frac{t}{T_\text{s}}\right)$, then the

discretized version of $R_{xx}(\tau) = \sigma^2_{\text{cont}}\delta(\tau)$ is[2]

$$R_{x_k x_k}[m] = \underbrace{\sigma^2_{\text{cont}} \cdot \frac{1}{T_{\text{s}}}}_{\sigma^2_{\text{disc}}} \delta[m].$$

Task 1.a
(0%)

$$\boxed{\text{Implement a code simulation of a simple Linear Time-Invariant (LTI) system that takes an input signal and produces an output signal.}}$$

Further instructions:

- For the input signal, produce an instance of a white Gaussian stochastic process $x(t)$ with zero mean and $R_{xx}(\tau) = N_0\delta(\tau)$. Indicatively, $N_0 = 2$.
- Input signal time interval: $[0, 1]\,\text{s}$
- The spectrum of the LTI system impulse response is $H(f) = \Pi\left(\frac{f}{2W}\right)$. Indicatively, $W = 20\,\text{Hz}$.
- Derive theoretically $h(t)$ and use it with finite duration within $[-T_h/2, T_h/2]$. Indicatively, $T_h = 0.25\,\text{s}$.
- Sampling frequency: $1000\,\text{Hz}$
- In a 2×3 subplot, in the first row plot the input signal, the LTI system impulse response, the output signal (in this order), and in the second row plot the amplitude of their corresponding Fourier transforms.

Task 1.b
(0%)

Implement a code simulation for the exercise 4.7.2 from the course notes. For many implementations of the stochastic process, estimate the power spectral density by averaging the periodograms. Derive the theoretical PSD (power spectral density). Plot both the theoretical and the experimental PSD.

Further instructions:

- The $R_{ss}(t)$ has been derived in class. You may use the immediate result formula by adding a reference to the solution.
- Indicative values:
  - Number of s_i symbols: 100 (which implies that 102 random b_i bits have to be generated)
  - Number of instances of the stochastic process $s(t)$: 1000

[2]This is just the intuition, not a real proof.

  - Symbol period: $10^{-2}\,\mathrm{s}$
  - Oversampling factor (i.e., $\frac{\text{symbol period}}{\text{minimum time step}}$): 100
- In the figure, limit the x-axis in $[-500, 500]\,\mathrm{Hz}$.

2 Frequency Conversion

Task 2.a
(15%)

Implement a code simulation for the upconversion of a baseband (complex) signal, then the unfiltered downconversion, and then the filtered result.

Further instructions:

- Carrier frequency: 2 Hz
- Symbol period: 1 s
- Oversampling factor (i.e., $\frac{\text{symbol period}}{\text{minimum time step}}$): 10
- Shaping pulse and matched filter: SRRC (the code generating it is given).
  - Half duration of the pulse: 5 symbol periods
  - Roll-off factor: 0.4
- Put these 4-QAM symbols in a shaped signal in this order:

$$
\begin{aligned}
s_0 &= -1 - j1 \\
s_1 &= +1 - j1 \\
s_2 &= +1 + j1 \\
s_3 &= -1 - j1 \\
s_4 &= +1 + j1
\end{aligned}
$$

- In a 2×4 subplot, in the first row plot the baseband signal, the passband signal, the unfiltered downconverted signal, the final filtered signal, and in the second row plot the amplitude of their corresponding Fourier transforms.
- For *complex (non-real) signals*, plot *both* the real (in blue) and imaginary (in red) components within the same panel; for *real signals*, include *only a single* blue plot. Whether a signal is complex (non-real) or real is not meant programmatically (i.e., the variable type in the code) but theoretically. It is important to follow this instruction.
- Avoid using legends, as they may clutter the plots.

Make sure that in the final signal, if sampled at $t_k = kT$, for $k = 0, \ldots, 4$, and T being the symbol period, we get the initial symbols s_k.

3 Frequency-Shift Keying (FSK) Modulation

For any $n \in \mathbb{Z}$, define

$$\mathbb{Z}_{\geq n} \triangleq \{m \in \mathbb{Z} \mid m \geq n\} = \{n, n+1, \dots\}.$$

For any $n \in \mathbb{Z}_{\geq 1}$, define

$$\mathbb{Z}_n \triangleq \{m \in \mathbb{Z}_{\geq 0} \mid m < n\} = \{0, \dots, n-1\}.$$

Define the operator

$$[\cdot]_2 : \mathbb{Z} \to \mathbb{Z}_2$$

that returns the remainder of its input modulo 2. E.g., $[-11]_2 = [5]_2 = 1, [-8]_2 = [6]_2 = 0$.

For any $f : \mathbb{R} \to \mathbb{C}$, define the operator $\overleftarrow{\cdot}$ as

$$\overleftarrow{f}(t) \triangleq f(-t).$$

For any $f, g : \mathbb{R} \to \mathbb{C}$, define the operator $\langle \cdot, \cdot \rangle$ as

$$\langle f, g \rangle \triangleq \int_{-\infty}^{\infty} f(\tau) g^*(\tau)\, d\tau = f(t) * \overleftarrow{g}^*(t)\Big|_{t=0} = \bar{R}_{fg}(0).$$

For a symbol period $T > 0$, let $\varphi : \mathbb{R} \to \mathbb{R}$ be a function such that

$$\langle \varphi, \varphi \rangle = 1.$$

M-FSK uses a different shaping pulse g_m for each symbol s_m, defined as

$$g_m(t) \triangleq \varphi(t) e^{j2\pi s_m \Delta F t},$$

for $m \in \mathbb{Z}_M$ and some $\Delta F > 0$, where $m \mapsto s_m \in \mathbb{R}$ is just an invertible mapping with integer differences,[3] i.e., $\forall m, m' \in \mathbb{Z}_M,\ s_{m'} - s_m \in \mathbb{Z}$.

For some $K \in \mathbb{Z}_{\geq 1}$, let the sequence $m_0, \dots, m_{K-1} \in \mathbb{Z}_M$ to be transmitted. The transmitter generates the baseband signal

$$s(t) = \sum_{k=0}^{K-1} A g_{m_k}(t - kT),$$

[3]This restriction ensures that the minimum distance between the symbols remains (at least) the same as before the mapping $m \mapsto s_m$.

for some amplifying quantity $A > 0$.

Assuming a flat-fading channel, the receiver receives the (baseband equivalent) signal

$$r(t) = hs(t) + n(t),$$

where $h \in \mathbb{C}$ and $n(t)$ is a complex stochastic process representing noise (its statistics are omitted since we will not use them).

Afterward, the $h^* r(t)$ signal is passed through the M matched filters $\overleftarrow{g}_0^*, \dots, \overleftarrow{g}_{M-1}^*$, and then it is sampled at $t = kT$ for $k \in \mathbb{Z}_K$, i.e.,

$$\mathbf{r}_k \triangleq \begin{bmatrix} r_k^{(0)} \\ \vdots \\ r_k^{(M-1)} \end{bmatrix},$$

where

$$r_k^{(m)} \triangleq \Big(h^* r(t)\Big) * \overleftarrow{g}_m^*(t)\Big|_{t=kT}.$$

For any $m \in \mathbb{Z}_M$ and $k \in \mathbb{Z}_K$, define φ_k and $g_{m,k}$ as

$$\varphi_k(t) \triangleq \varphi(t - kT),$$
$$g_{m,k}(t) \triangleq g_m(t - kT).$$

Task 3.a
(0%)

Prove that

$$r_k^{(m)} = A|h|^2 \sum_{k'=0}^{K-1} \Big\langle g_{m_{k'},k'}, g_{m,k} \Big\rangle + n_k^{(m)},$$

where

$$n_k^{(m)} \triangleq h^* \Big(n(t) * \overleftarrow{g}_m^*(t)\Big)\Big|_{t=kT}.$$

Here, the objective is to select the parameters so that we will be able to perform the decision rule on the k-th received sample.

We introduce the following intermediate result that will help to simplify the analysis (for the described family of φ functions), by focusing just on $\langle g_{m'}, g_m \rangle$.

Task 3.b
(1%)

For

- $m, m' \in \mathbb{Z}_M$ and $k, k' \in \mathbb{Z}_K$,
- φ being zero outside $[0, T)$,

prove that

$$\Big\langle g_{m',k'}, g_{m,k} \Big\rangle = \Big\langle g_{m'}, g_m \Big\rangle \delta_{k,k'}.$$

For the above scenario, we can say that $\langle g_{m',k'}, g_{m,k}\rangle$ is orthogonal with respect to (k, k'), if $\langle g_{m'}, g_m\rangle \neq 0$. (See how is orthogonality defined right after the next task.)

Task 3.c
(2%)

For

- $m, m' \in \mathbb{Z}_M$,
- φ being constant in $[0, T)$ and zero elsewhere,

prove that

$$\Big\langle g_{m'}, g_m \Big\rangle = e^{j\pi\Delta F\Delta sT}\,\mathrm{sinc}(\Delta F\Delta sT),$$

where $\Delta s \triangleq s_{m'} - s_m$.

- What is the minimum value of ΔF that guaranties the orthogonality of $\langle g_{m'}, g_m\rangle$ with respect to (m, m')?
- For that value of ΔF and the aforementioned φ, derive the formula for $r_k^{(m)}$ from the Task 3.a.

Orthogonality of $\langle g_{m'}, g_m\rangle$ with respect to (m, m') means that $\langle g_{m'}, g_m\rangle = c\delta_{m,m'}$ for some constant $c \in \mathbb{C}^*$.

Let us study the case where we may try to make ΔF even smaller.

Task 3.d
(2%)

For

- $m, m' \in \mathbb{Z}_M$,
- φ being constant in $[0, T)$ and zero elsewhere,
- $\Delta F = \frac{1}{2T}$,

prove that

$$\Big\langle g_{m'}, g_m \Big\rangle = \delta^{\mathrm{FSK}}_{m,m'},$$

where

$$\delta^{\mathrm{FSK}}_{m,m'} \triangleq \delta_{m,m'} + j\frac{2/\pi}{\Delta s + \delta_{m,m'}}[\Delta s]_2, \quad \text{for } \Delta s \triangleq s_{m'} - s_m.$$

For the aforementioned φ and ΔF, derive the formula for $r_k^{(m)}$ from the Task 3.a.

Although orthogonality may be lost, it can be restored by discarding the imaginary part.

Notice that the above results keep the mapping $m \mapsto s_m$ general. To understand the usefulness of keeping the mapping $m \mapsto s_m$ general, let us study another case in the following two tasks. For that case, we will use an SRRC pulse for φ.

Task 3.e
(3%)

For

- a single symbol per shaped signal (i.e., $K = 1$),
- φ being an SRRC pulse with symbol period T, roll-off factor β,
- $m, m' \in \mathbb{Z}_M$ and $k, k' \in \mathbb{Z}_K$,

prove that

$$\left\langle g_{m',k'}, g_{m,k} \right\rangle = \Phi(f) * \Phi(f)\Big|_{f=\Delta F \Delta s},$$

where:

- $\Delta s \triangleq s_{m'} - s_m$,
- $\varphi \overset{\mathcal{F}}{\rightleftharpoons} \Phi$.

What is the minimum value of ΔF that guaranties the orthogonality of $\langle g_{m',k'}, g_{m,k} \rangle$ with respect to both (m, m') and (k, k')?

Task 3.f
(2%)

- For the specifications from the Task 3.e, including the ΔF that you found, and the mapping $s_m = m$, what is the frequency range on the baseband (at the transmitter)?
- What change would you suggest in order to make the baseband frequency range symmetric around 0 while maintaining the bandwidth?

Now, we will implement a simulation for each FSK symbol to visualize how signals look. *Completing the previous tasks in this section is a requirement for proceeding to this one.*

Task 3.g
(25%)

Implement a code simulation for FSK. For each distinct symbol of the modulation, create a figure plotting its corresponding signal as a function of time in the states described below (see the further instructions).

- For each value of ΔF:
  - What do you observe on the transmitted signals between different symbols?
  - What do you observe on the filtered signals for each sent symbol?
- For a consecutive transmission of many symbols, which value of ΔF would you choose if you want the transmitter to send a continuous signal all the time? Why?
- Which value of ΔF would you choose if this continuous signal requirement is not a concern? Why?

Further instructions:

- Carrier frequency: 6 Hz
- Symbol period: 1 s
- Oversampling factor (i.e., $\frac{\text{symbol period}}{\text{smallest time unit}}$): 100
- Assume 5-FSK, using the mapping that you found in the Task 3.f.
- φ is constant in $[0, T)$ and zero elsewhere.
- $\Delta F \in \left\{\frac{1}{T}, \frac{1}{2T}\right\}$
- Assume ideal channel.
- Assume no noise.
- Each figure represents a distinct symbol, meaning the total number of figures will match the number of symbols in the modulation.
- For *complex (non-real) signals*, plot *both* the real (in blue) and imaginary (in red) components within the same panel; for *real signals*, include *only a single* blue plot. Whether a signal is complex (non-real) or real is not meant programmatically (i.e., the variable type in the code) but theoretically. It is important to follow this instruction.
- All figures should follow the format shown below. Avoid using legends, as they may clutter the plots.
- Ensure that the filtered signal plots have identical axis scales.

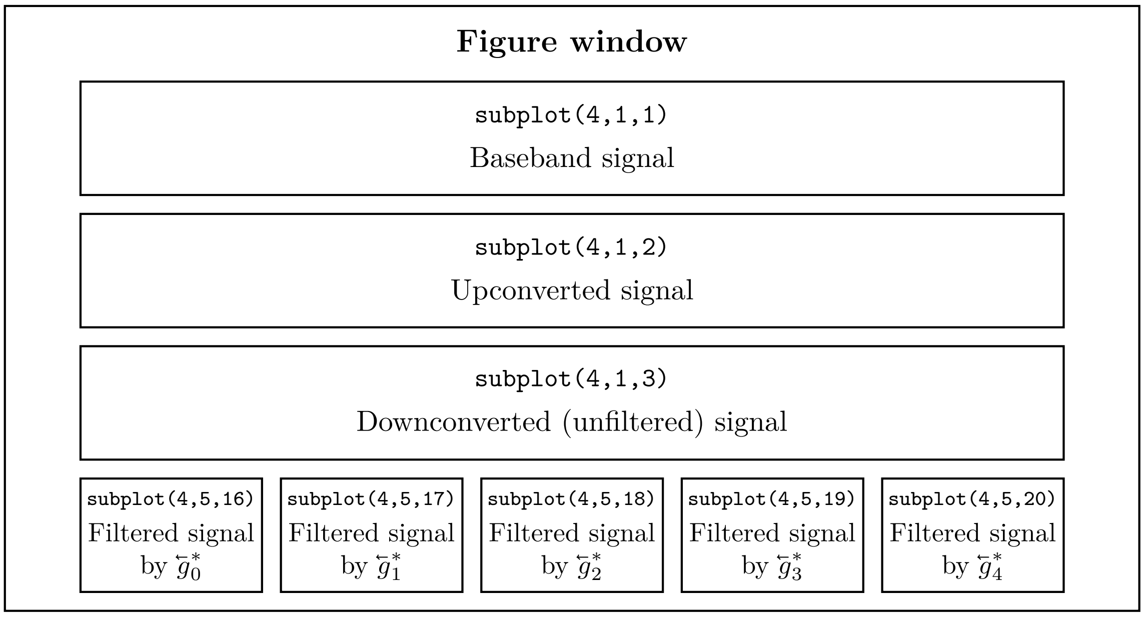
Figure window
subplot(4,1,1)
Baseband signal
subplot(4,1,2)
Upconverted signal
subplot(4,1,3)
Downconverted (unfiltered) signal
subplot(4,5,16) Filtered signal by $\overleftarrow{g}_0^*$
subplot(4,5,17) Filtered signal by $\overleftarrow{g}_1^*$
subplot(4,5,18) Filtered signal by $\overleftarrow{g}_2^*$
subplot(4,5,19) Filtered signal by $\overleftarrow{g}_3^*$
subplot(4,5,20) Filtered signal by $\overleftarrow{g}_4^*$

4 Non-Ideal Channel

When we began learning about telecommunication systems, the AWGN (*Additive White Gaussian Noise*) ideal channel,

$$y_{\mathrm{BP}}(t) = x_{\mathrm{BP}}(t) + n(t),$$

was introduced, which, for simplicity, ignores a lot of things that happen. One of those things is that the channel may scale the sending signal and interfere it with itself, like this,

$$y_{\mathrm{BP}}(t) = a_0 x_{\mathrm{BP}}(t) + a_1 x_{\mathrm{BP}}(t - t_1) + n(t). \tag{4.1}$$

If the baseband equivalent of the channel can be described by an LTI (linear time invariant) system with impulse response $h\delta(t)$, then the channel is called *flat-fading* (since its Fourier transform is flat, i.e., constant) and the relationship

$$Y = hX + N \tag{4.2}$$

describes the receiver's sampling procedure for a single sent symbol X.

Y, X, N are random variables, and, of course, X, N are independent. $h \in \mathbb{C}$ is deterministic.

The ideal channel is a special case of the above where $h = 1$.

The flat-fading channel will be our assumed model for this section.

The SNR on the transmitter is defined as

$$\mathsf{SNR}_{\mathrm{Tx}} \triangleq \frac{\mathbb{E}\left[|X|^2\right]}{\mathbb{E}\left[|N|^2\right]}.$$

The SNR on the receiver is defined as

$$\mathsf{SNR}_{\mathrm{Rx}} \triangleq \frac{\mathbb{E}\left[|hX|^2\right]}{\mathbb{E}\left[|N|^2\right]} = |h|^2 \cdot \mathsf{SNR}_{\mathrm{Tx}}.$$

Let $N \sim \mathcal{CN}(0, N_0)$, and X be uniformly distributed over the constellation symbol set.

Initially, assume that the system uses the BPSK modulation, which implies that $X \sim \mathcal{U}\{\pm a\}$ for some $a > 0$.

Task 4.a (0%)

Derive the theoretical transmitter SNR, $\mathsf{SNR}_{\text{Tx,BPSK}}$.

Task 4.b (0%)

Derive the decision rule for the optimal detector. Assume that h is known to the receiver.

Task 4.c (0%)

Derive the probability of error as a function of $\mathsf{SNR}_{\text{Tx,BPSK}}$ and h.

Now, assume that the system uses the 4-QAM modulation (with Gray coding), which implies that $X \sim \mathcal{U}\{\pm a \pm ja\}$ for some $a > 0$.

Task 4.d (0%)

Repeat Task 4.a for the new modulation (i.e, $\mathsf{SNR}_{\text{Tx,4QAM}}$).

Task 4.e (3%)

Repeat Task 4.b for the new modulation.

Think smartly, and the BPSK detector will pop up.

Task 4.f (1%)

Repeat Task 4.c for the new modulation (i.e., function of $\mathsf{SNR}_{\text{Tx,4QAM}}$ and h).

Completing the previous tasks in this section is a requirement for proceeding to this one.

Task 4.g (46%)

Implement a code simulation for the full system (transmitter, receiver, channel with 2 rays), which includes: the shaping pulse and upconversion in the transmitter; the effect of the channel; downconversion, the matched filter, the sampling, and the detection in the receiver. Use 4-QAM. Plot the Fourier transform of the channel impulse response. Keep track of the experimental BER for each $\mathsf{SNR}^{\text{dB}}_{\text{Tx,4QAM}}$ value. Plot the experimental BER as a function of $\mathsf{SNR}^{\text{dB}}_{\text{Tx,4QAM}}$. In the same panel, plot the theoretical BER as reference. Write your observations for different channels as instructed below.

Further instructions:

- $a = 1$ (the 4-QAM symbol parameter)
- Carrier frequency: 800 kHz
- Passband bandwidth: 80 kHZ
- Oversampling factor (i.e., $\frac{\text{symbol period}}{\text{minimum time step}}$): 200
- Shaping pulse and matched filter: SRRC (the code generating it is given).
  - Half duration of the pulse: 4 symbol periods
  - Roll-off factor: 0.35
- Generate a big number of random bits for the experiment (indicatively, 10^5 bits).
- The channel, described by the parameters (a_0, a_1, t_1), remains the same for the transmission of all the generated bits.
- Put m symbols in a shaped signal (i.e., the signal after the shaping pulse). (Indicatively, $m = 10$)
- The receiver receives the signal (4.1).
- The detector uses the approximation channel coefficient $h = H(0)$ to apply the decision rule on (4.2).
  - Assume that the receiver knows the channel parameters (a_0, a_1, t_1).
- Run for $\mathsf{SNR}^{\text{dB}}_{\text{Tx,4QAM}} = 0, 2, \ldots, 12$.
- Use logarithmic y-axis for the BER plots.
- The theoretical BER plot should be the same color with its corresponding experimental one, but the theoretical will be as a dashed line and the experimental as a solid line.
- To plot the Fourier transform of the channel impulse response, use the $H(f)$ formula as derived in theory.
  - In a 2×1 subplot, plot $|H(f)|$ in the upper subplot, and $\angle H(f)$ in the lower one.
  - Use y-axis limits: $[0, 1]$ for the upper and $[-1, 1]$ for the lower subplot.
  - If the baseband bandwidth range is $[-W, W]$, plot for $f \in [-10W, 10W]$ with step $W/50$ in both subplots.
  - Make the baseband bandwidth area stand out (yellow color). (Do it before the plotting of H so as it does not cover it.)
  - In both subplots, the color to be used will be the same as in the corresponding BER plots.

- Make the experiment described as of now repeat itself internally in the simulation 3 times, each time for different channel parameters (a_0, a_1, t_1), transmitting the same bits each time.
  - So, the plots produced at each time will be plotted in the same panels, respectively.
  - Change the plot colors between these 3 times. Legends will be helpful here unless you use the following recommendation: use blue, red, green (in this order).
- Answer to the following questions. (T is the symbol period.)
  - Run the simulation for the channel parameters triplet

    $$(0.5,\, 0.3,\, 0.2222\,T)$$
    $$(0.5,\, 0.3,\, 0.4444\,T)$$
    $$(0.5,\, 0.3,\, 0.963\,T)$$

    * What do you observe between the experimental BER plots? Explain it (using the channel spectrum).
    * Explain the relationship between the corresponding theoretical and experimental BERs.
  - Rerun for

    $$(0.5,\, 0.3,\, 0.037\,T)$$
    $$(0.5,\, 0.3,\, 0.1261\,T)$$
    $$(0.5,\, 0.3,\, 0.2222\,T)$$

    * Why are BERs getting better in this scenario, even though the time lag is increasing? Explain it (using the channel spectrum).
    * Explain intuitively the relation between the two rays (how does one affect the other) in the first case (the worst one).
    * Explain the relationship between the corresponding theoretical and experimental BERs.
  - Now, let us try the scenario where one of the cases has only the primary ray (no second ray). Rerun for

    $$(0.5,\, 0,\, 0\,T)$$
    $$(0.5,\, 0.3,\, 0.07\,T)$$
    $$(0.5,\, 0.3,\, 0.96\,T)$$

  - $*$ Can the presence of a second ray be advantageous? Can it be damaging? Explain (using the channel spectrum).
  - $*$ Explain the relationship between the corresponding theoretical and experimental BERs.

Reminder: $\mathsf{SNR}^{\text{dB}}_{\text{Tx,4QAM}} = 10\log_{10}(\mathsf{SNR}_{\text{Tx,4QAM}})$.

Advice:

- Try to solve it in stages, breaking the main problem into smaller, more manageable parts.
- Try to implement it efficiently (using vectors and matrices is way faster than for-loops).

The next part of the assignment builds upon this task.

Code Segments

```python
import numpy as np

def gen_srrc(T, over, A, a):
    """
    phi, t = gen_srrc(T, over, A, a)

    OUTPUT
         phi: truncated SRRC pulse, with parameter T, roll-off
                factor a, and duration 2*A*T
         t:   time axis of the truncated pulse

    INPUT
         T:  Nyquist parameter or symbol period (positive real
               number)
         over: positive integer equal to T/T_s (oversampling
               factor)
         A:  half duration of the pulse in symbol periods
               periods (positive integer)
         a:  roll-off factor (real number between 0 and 1)

       Created using the MATLAB original source code from
       A. P. Liavas, Oct. 2020
    """

    Ts = T / over

    # Create time axis, avoiding division by zero at t = 0
    t = np.arange(-A * T, A * T + Ts/2, Ts) + 1e-8

    if 0 < a <= 1:
        num = np.cos((1 + a) * np.pi * t / T) \
                + np.sin((1 - a) * np.pi * t / T) / (4 * a * t /
                   T)
        denom = 1 - (4 * a * t / T) ** 2
        phi = 4 * a / (np.pi * np.sqrt(T)) * num / denom
    elif a == 0:
        phi = 1 / np.sqrt(T) * np.sin(np.pi * t / T) \
                / (np.pi * t / T)
    else:
        phi = np.zeros(len(t))
        print("Illegal value of roll-off factor")
        return

    return phi, t
```

Code Segment 1: (Python 3) Generate an SRRC pulse.

```
function [phi, t] = gen_srrc(T, over, A, a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [phi, t] = gen_srrc(T, over, A, a)                            %
% OUTPUT                                                        %
%      phi: truncated SRRC pulse, with parameter T, roll-off    %
%             factor a, and duration 2*A*T                      %
%      t:   time axis of the truncated pulse                    %
%                                                               %
% INPUT                                                         %
%      T:  Nyquist parameter or symbol period (positive real    %
%            number)                                            %
%      over: positive integer equal to T/T_s (oversampling      %
%            factor)                                            %
%      A:  half duration of the pulse in symbol periods         %
%            periods (positive integer)                         %
%      a:  roll-off factor (real number between 0 and 1)        %
%                                                               %
%    A. P. Liavas, Oct. 2020                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ts = T/over;

% Create time axis
t = [-A*T:Ts:A*T] + 10^(-8); % in order to avoid
                             % division by zero problems at t=0.

if (a>0 && a<=1)
   num = cos((1+a)*pi*t/T) + sin((1-a)*pi*t/T) ./ (4*a*t/T);
   denom = 1-(4*a*t./T).^2;
   phi = 4*a/(pi*sqrt(T)) * num ./ denom;
elseif (a==0)
   phi = 1/(sqrt(T)) * sin(pi*t/T)./(pi*t/T);
else
    phi = zeros(length(t),1);
    disp('Illegal value of roll-off factor')
    return
end
```

Code Segment 2: (MATLAB) Generate an SRRC pulse.