ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 21 Modern Digital Communications
Final Exam December 21, 2022

Name:

Note:

- You have 2 hours to work on the exam.
- The exam is closed book, but you are allowed one double-sided A4 page of handwritten notes. Resources from the internet as well as code written outside this exam are not allowed (unless the code is written on the sheet of handwritten notes).
- The code will be evaluated according to the usual criteria, namely correctness, speed, form, and readability. Short comments that allow us to follow what you are doing will improve readability.
- The problems can be solved in any order.
- You will upload (to Moodle) your solution to the problems that require writing MATLAB or Python code. Do so in a single archive.

To get started with the exam, do the following:

1. Close all the windows and programs on your laptop.
2. Launch the MATLAB/Python editor and close all the tabs (previously written code).
3. From Moodle, download the file `mdc_final_2022.zip`. Unzip the file to create the directory `mdc_final_2022`. For the rest of the exam you are required to work inside that directory. The MATLAB/Python files for Problem n, are found in subfolder pn.
4. Turn your WiFi off until you are ready to upload your solutions.
5. Wait until you receive the go-ahead signal.

PROBLEM 1. 18 points (Paper and Pencil)

Consider the communication system of Figure 1.

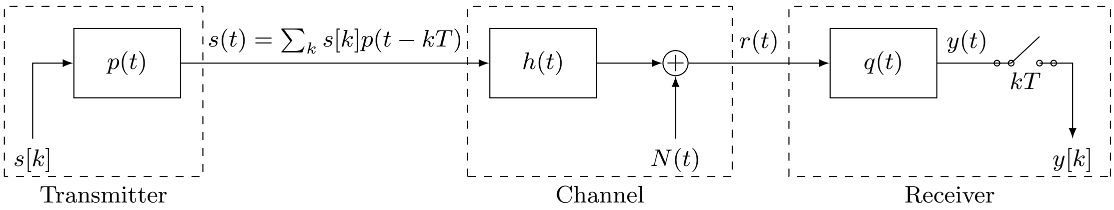

Figure 1: Communication System of Interest

$N(t)$ is assumed to be complex-valued AWGN with zero mean and variance $\frac{N_0}{2}$ per real-valued dimension.

1. Compute $r(t)$ and express it as a function of the sequence $\{s[k]\}$, some filter $g(t)$, and $N(t)$. Specify $g(t)$ as a function of $p(t)$ and $h(t)$.

2. Compute $y(t)$ and express it as a function of the sequence $\{s[k]\}$, some filter $f(t)$, and some noise process $Z(t)$. Specify $f(t)$ as a function of $p(t)$, $h(t)$ and $q(t)$. Specify $Z(t)$.

3. Compute $y[k] = y(kT)$, and express it in the form $y[k] = \sum_n b_n v[n] + z[k]$. Specify the sequence $\{b_n\}$, the filter $v[n]$ and the noise $z[k]$.

4. Compute the mean and the covariance of the random process $z[k]$.

5. Assume that $p(t)$ is a Nyquist pulse, $q(t) = p^*(-t)$ and $h(t) = \delta(t)$. Compute $y[k]$ of question 3 under these new assumptions, as well as the covariance of the process $z[k]$.

6. Is $p(t) = \frac{1}{\sqrt{T}} \operatorname{sinc}(\frac{t}{T})$ a Nyquist pulse? Argue your answer using $p_{\mathcal{F}}(f)$, which is the Fourier transform of $p(t)$.

Problem 2. 24 points (MATLAB/Python)

In this problem you are asked to implement an OFDM transmitter and receiver, where all the carriers are used for transmission.

Please follow the instructions in the script `p2.m[py]`.

PROBLEM 3. 18 points (Paper and Pencil + MATLAB/Python)

Consider an OFDM system with N carriers, described by the following equation, as seen in class:

$$\boldsymbol{Y} = D\boldsymbol{A} + \boldsymbol{Z},$$

where $\boldsymbol{Y}$ is the column vector of received symbols, $\boldsymbol{A}$ is the column vector of transmitted symbols, $\boldsymbol{Z}$ is the column vector of the noise, and D is the matrix, defined as in class, which contains the strengths of the parallel channels created by OFDM. Note that $\boldsymbol{Y}$, $\boldsymbol{A}$ and $\boldsymbol{Z}$ can be matrices as well, if we consider the transmission/reception of several consecutive OFDM blocks.

Assume that we know the strengths of the parallel channels (the lambdas). They are given in the file `lambdas.mat`. Also, you are given the matrix $\boldsymbol{Y}$ of the received symbols, in the file `received_OFDM_symbols.mat`, as well as the first column of the matrix $\boldsymbol{A}$, which contains the training symbols, the corresponding file being `training_symbols.mat`. We would like to estimate the transmitted data symbols (the remaining columns of the matrix $\boldsymbol{A}$).

Let $\boldsymbol{y}$, $\boldsymbol{a}$ and $\boldsymbol{z}$ be the k-th columns of $\boldsymbol{Y}$, $\boldsymbol{A}$ and $\boldsymbol{Z}$, respectively. For any k, we can write:

$$\boldsymbol{y} = D\boldsymbol{a} + \boldsymbol{z}.$$

1. Specify the matrix B, such that $B\boldsymbol{y}$ is the LS (Least Squares) estimate of $\boldsymbol{a}$.
2. Find the matrix C, such that $C\boldsymbol{y}$ is the LMMSE (Linear Minimum Mean Square Error) estimate of $\boldsymbol{a}$. Assume that $\boldsymbol{z} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2 I_N)$, $E\left[\boldsymbol{a}\boldsymbol{a}^\dagger\right] = I_N$, and $\boldsymbol{a}$ and $\boldsymbol{z}$ are statistically independent.

After finding the matrices above, please proceed and follow the instructions in the script `p3.m[py]`.

You might find useful the commands:
`m\n` (for MATLAB) and
`numpy.linalg.lstsq(m, n, rcond=None)[0]` (for Python).
Of course, you need to figure out what to use for the matrices `m` and `n`.