ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 22 | Modern Digital Communications

Final Exam | December 20, 2017

Name:

Note:

- You have 2 hours to work at the exam.
- The exam is closed book, but your are allowed one A4 page (double-sided) 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 code. Do so in a single archive.

Hints:

- Let $\mathcal{U}$ be a subspace of an inner product space $\mathcal{V}$. Let $\boldsymbol{v} \in \mathcal{V}$ and let $\boldsymbol{u}$ be the vector in $\mathcal{U}$ which is closest to $\boldsymbol{v}$ in a squared-norm sense. The projection theorem says that $\boldsymbol{v} - \boldsymbol{u}$ is orthogonal to every vector of $\mathcal{U}$.
- You may want to check out the MATLAB command `toeplitz(a,b)`, where `a` and `b` are vectors.

To get started with the exam, do the following:

1. Close all the windows and programs on your laptop.
2. Launch MATLAB and close all the tabs (previously written code).
3. From Moodle, download the file `mdc_final_2017.zip`. Unzip the file to create the directory `mdc_final_2017`. For the rest of the exam you are required to work inside that directory.
4. Turn your WiFi off until you are ready to upload your solutions.
5. Wait until you receive the go-ahead signal.

PROBLEM 1. 15 points (Paper and Pencil / MATLAB)

Let $\boldsymbol{x} = (x_0, \ldots, x_{N-1})^T$ be a symbol sequence transmitted over a symbol-level channel and let

$$y_n = \sum_{i=0}^{L-1} h_i x_{n-i} + z_n, \quad n = 0, \ldots, N+L-2,$$

be the noisy channel output (where $x_n = 0$ if $n \notin \{0, \ldots, N-1\}$.) Each z_n is a sample from an independent zero-mean Gaussian random variable of variance σ^2.

1. Let $\boldsymbol{y} = (y_0, \ldots, y_{N+L-2})^T$, $\boldsymbol{h} = (h_0, \ldots, h_{L-1})^T$ and $\boldsymbol{z} = (z_0, \ldots, z_{N+L-2})^T$. Specify the matrix C such that
$$\boldsymbol{y} = C\boldsymbol{h} + \boldsymbol{z}.$$

2. Suppose that $\boldsymbol{h} \in \mathbb{R}^L$ is the realization of a Gaussian random vector $\boldsymbol{H} \in \mathbb{R}^L$ which is independent of the noise. The receiver wants to estimate $\boldsymbol{h}$, using $\boldsymbol{x}$ as a training sequence. (Hence $\boldsymbol{x}$ is known to the receiver.) Derive the expression for the MMSE estimator $\hat{\boldsymbol{h}}(\boldsymbol{y})$. Your expression can only depend on the covariances $K_{\boldsymbol{Z}}$ and $K_{\boldsymbol{H}}$, as well as on C.

3. Implement and test your channel estimation with a MATLAB script. The variables $\boldsymbol{x}$, $\boldsymbol{y}$, σ, $K_{\boldsymbol{H}}$ can be loaded with the command `load prob1.mat`. The command will also load $\boldsymbol{h}$, so that you can test your code (e.g. by plotting the absolute value of $\boldsymbol{h}$ and that of the estimate $\hat{\boldsymbol{h}}$).

Problem 2. 17 points (MATLAB)

The file `rx_signal.mat` contains a matrix, the columns of which essentially consist of noisy 4-QAM symbols that form an OFDM block. You may think of it as the matrix obtained by an OFDM receiver, after it takes the FFT (column by column) and it divides each row by the channel frequency response evaluated at the carrier frequency of that row. However, due to the presence of a carrier-frequency offset (CFO), each element of the same column is rotated by some amount. (Same amount for each element of the column.)

1. Write a MATLAB script that, for each column (OFDM block), finds an estimate of the rotation. This can be done without the need of pilot symbols, because the rotation of the first block is small (hence, with high probability, the original symbols can be determined in spite of the rotation), and from one block to the next the rotation changes by a small amount. Your code should contain at most 1 loop.

2. Correct the rotation of the OFDM blocks and plot the phase-corrected constellation.

Problem 3. 12 points (Paper and Pencil / MATLAB)

Let $p(t)$ be a known signal and let $r(t) = p(t-\tau)e^{j2\pi dt}$. We view τ and d as a delay and a Doppler frequency, respectively, both of which are unknown. We are interested in estimating d. Let $p_{\mathcal{F}}(f)$ and $r_{\mathcal{F}}(f)$ be the Fourier transforms of $p(t)$ and $r(t)$ respectively.

1. Express $r_{\mathcal{F}}(f)$ as a function of $p_{\mathcal{F}}(f)$.
2. Describe a method for estimating d from $|r_{\mathcal{F}}(f)|$.
3. The files `p.mat` and `r.mat` contain the samples of $p(t)$ and $r(t)$, respectively. Write a MATLAB routine that estimates d according to the method of question 2. Choose the DFT size N so that the estimate is quantized in steps of 1 Hz or better. The sampling time is $T_s = 10^{-5}$ seconds.
4. Correct $r(t)$ for the Doppler and do a scatterplot of the result. If your routine works correctly, you should see a noisy BPSK constellation (with no rotation).

PROBLEM 4. 16 points (Paper and Pencil / MATLAB)

The file `phaseEstimate.mat` contains a sequence of N noisy phase estimates $y_1, \ldots, y_N$ made at regular intervals of time.

1. Plot the sequence y_i as a function of the index i. (Note: You should be able to see the presence of a carrier-frequency offset (CFO), i.e., that the points represent noisy measurements of points that should lie on a line.)

2. We want to derive the least squares (LS) estimate of the line $ai + b$ that best fits the y_i sequence, where best is meant in terms of minimizing the squared error

$$\sum_{i=1}^{N} [y_i - (ai + b)]^2. \tag{1}$$

   The following steps will guide you through the derivation of the LS estimate of the vector $\boldsymbol{v} = (a, b)^T$ that describes the parameters a and b to be estimated. First, write (1) in the following matrix form

$$\|\boldsymbol{y} - H\boldsymbol{v}\|^2, \tag{2}$$

   where $\boldsymbol{y} = (y_1, \ldots, y_N)^T$. Specify H.

3. Notice that $H\boldsymbol{v}$ is an element of some n-dimensional vector space $\mathcal{U}$. Specify the vectors $\boldsymbol{u}_k$, $k = 1, \ldots, n$ that span $\mathcal{U}$.

4. We are looking for the element $H\boldsymbol{v} \in \mathcal{U}$ which is closest to $\boldsymbol{y}$ in squared norm. Use the projection theorem to derive n expressions that need to be satisfied by $H\boldsymbol{v}$.

5. Use matrix notation to write the n expressions of the previous question in a single equation, and solve for $\boldsymbol{v}$. The result is the LS estimator. The expression that you have found should be

$$\boldsymbol{v}_{LS}(\boldsymbol{y}) = (H^T H)^{-1} H^T \boldsymbol{y}.$$

6. Implement the LS estimator in MATLAB, and to test it, plot on the same figure the line $ai + b$, $i = 1, \ldots, N$ and the original phase estimates.