ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 11 Modern Digital Communications
Midterm Exam November 7, 2018

Name:

Note:

- You have 2 hours to work at the exam.
- The exam is closed book, but your are allowed one sheet (one single-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 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 MATLAB and close all the tabs (previously written code).
3. From Moodle, download the file `mdc_midterm_2018.zip`. Unzip the file to create the directory `mdc_midterm_2018`. For the rest of the exam you are required to work inside that directory. The MATLAB files for Problem n, $n = 3, 4$, 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. 10 points (Paper and Pencil)

Consider a baseband signal $s(t)$ of support contained in $[0, T_p]$, where $T_p = NT_s$ for some positive even integer N, and suppose that the support of its Fourier transform $s_{\mathcal{F}}(f)$ is essentially contained in $(-\frac{1}{2T_s}, \frac{1}{2T_s})$, so that $s(t)$ can be reconstructed from its samples taken every T_s seconds (or less). Let $\boldsymbol{s}$ be the vector of samples

$$\boldsymbol{s} = \big(s(0), s(T_s), \ldots, s(T_s(N-1))\big). \tag{1}$$

1. Let $\boldsymbol{s}_{\mathcal{F}}$ be the DFT of $\boldsymbol{s}$. Write the expression that describes the kth component of $\boldsymbol{s}_{\mathcal{F}}$ as a function of $s_{\mathcal{F}}(f)$.

2. Do the same with samples taken every $\frac{T_s}{2}$ seconds. Specifically, let

   $$\boldsymbol{u} = \big(s(0), s(T_s/2), s(T_s), \ldots, s((2N-1)T_s/2)\big) \tag{2}$$

   and let $\boldsymbol{u}_{\mathcal{F}}$ be the DFT of $\boldsymbol{u}$. Write the expression that describes the kth component of $\boldsymbol{u}_{\mathcal{F}}$ as a function of $s_{\mathcal{F}}(f)$. Obviously $\boldsymbol{u}_{\mathcal{F}}$ can be obtained from $\boldsymbol{s}_{\mathcal{F}}$ since from $\boldsymbol{s}_{\mathcal{F}}$ one can reconstruct $s(t)$, then sample it to obtain $\boldsymbol{u}$ and take the DFT. Is there a more straightforward way to obtain $\boldsymbol{u}_{\mathcal{F}}$ from $\boldsymbol{s}_{\mathcal{F}}$? Describe in words. Hint: You may want to start by sketching a sample $s_{\mathcal{F}}(f)$, e.g., a centered triangular function, and sketch the corresponding DFTs $\boldsymbol{s}_{\mathcal{F}}$ and $\boldsymbol{u}_{\mathcal{F}}$.

3. Write a MATLAB function (paper and pencil) that computes $\boldsymbol{u}_{\mathcal{F}}$ from $\boldsymbol{s}_{\mathcal{F}}$.

PROBLEM 2. 6 points (MATLAB)
Consider an object at position $p_s = (1000, 200, 0)$[meter], having velocity vector $v = (-10, 3, 0)$ [meter/sec], transmitting at carrier frequency $f_c = 10^9$ Hz and bitrate $r = 10^6$ bits/s. Consider a receiver at position $p_r = (0, 0, 0)$. Considering the speed of light to be $c = 3 \times 10^8$ [meter/sec], write a script `p2.m` that computes the following:

1. The Doppler frequency `fd` .
2. The bitrate `bitRateAtRx` seen by the receiver.
   Hint: print it with the command `fprintf('bitRateAtRx = %.5f\n',bitRateAtRx)`.

PROBLEM 3. 12 points (MATLAB)
Consider symbol-by-symbol on a pulse train simulated at the sample level. The symbols take values in $\{\pm 1\}$ and the pulses are shifted replicas of `pulse`. Subsequent pulses are non-overlapping and without space in between. Somewhere inside the symbol sequence there is the `preamble`. The signal is transmitted over a channel that multiplies the samples by $e^{j\phi_0}$ and adds white Gaussian noise. The phase $\phi_0 \in (-\pi, \pi]$ is unknown to the receiver. The channel output sequence is stored in `rxSamples`, but an initial and a final part of the channel output have been missed.

1. Estimate the phase ϕ_0 and the position where the first full pulse is contained in the received sequence. Note that the first full pulse might start before the start of the preamble.

2. From `rxSamples`, remove the excess samples before the start of the first pulse, and remove the samples after the last full pulse.

3. Using your estimate of ϕ_0, remove the rotation introduced by the channel.

4. Treat the result `r` as the output of the AWGN channel (without rotation) and, by means of a matched filter, construct a vector of sufficient statistics for the symbol sequence contained in `r`. Plot the resulting constellation.

5. From your sufficient statistics, estimate the symbol sequence. If your receiver was properly implemented, you should obtain the sequence stored in `txSymbols`.

PROBLEM 4. 12 points (MATLAB)
Suppose that we want to measure the distance to an object as well as the speed at which the object is moving away from us: at time $t = 0$ seconds we send a pseudorandom (PRN) `code`, and at time $t = 1$ seconds we send it again. These signals hit the target and come back to us, traveling at the speed of light. From the two roundtrip delays we determine the distances at two specific times, hence we can determine the (radial) speed, which is assumed to be constant. We can assume that the receiver clock is aligned with that of the transmitter.

Complete the MATLAB script `p4.m` as follows. Start by loading the file `signals.mat`. This file contains the PRN `code` and the received signal (`signalRx`). It also contains the transmitted signal (`signalTx`) which is not needed but can be useful for debugging purposes. Finally, it contains the vector `tof` that you can use if you cannot solve the first question.

1. Determine the travel times (receiver-target-receiver) for the two PRN sequences. (If you cannot solve this part, you can use `tof`, expressed in seconds).
2. Determine the corresponding receiver-target distances.
3. Since the object might be moving, the distances that you have found correspond to two specific times, say `t1` and `t2`. Compute them.
4. Print the target's (radial) speed, with positive sign if the target moves away from the receiver.
5. Print the distance to the object at $t = 0$.