Advanced Digital Signal Processing Coursework

Prof. Danilo P. Mandic
TAs: Giuseppe Calvi, Ilia Kisil, Ahmad Moniri, Takashi Nakamura

March 13, 2018

Contents

Guidelines

The coursework comprises five assignments, whose individual scores yield 90% of the final mark. The remaining 10% accounts for presentation and organisation. Students are allowed to discuss the coursework but must code their own MATLAB scripts, produce their own figures and tables, and provide their own discussion of the coursework assignments.

General directions and notation:

- The simulations should be coded in MATLAB, a *de facto* standard in the implementation and validation of signal processing algorithms.
- The report should be clear, well-presented, and include the answers to the assignments in this handout with appropriate numbering. Students are encouraged to submit through Blackboard **(in PDF format only)**, although a hardcopy submission will also be accepted at the undergraduate office.
- The report should document the results and the analysis in the assignments, in the form of figures (plots) and tables, and not by listing MATLAB code as a proof of implementation.
- We adopt the following notation: boldface lowercase letters (e.g. $\mathbf{x}$) for vectors, lowercase letters with a (time) argument ($x[n]$) for scalar realisations of random variables and elements of a vector, and uppercase letters (X) for random variables. Column vectors will be assumed unless otherwise stated, that is, $\mathbf{x} \in \mathbf{R}^{N\times 1}$.
- The typewriter font, e.g. `mean`, is used for MATLAB functions.

Presentation:

- The length limit for the report (all parts) is 42 pages. This corresponds to eight pages per assignment in addition to one page for front cover and one page for the table of contents, however, there are no page restrictions per assignment but only for the full-report (42 pages).
- The final mark also considers the presentation of the report, this includes: legible and correct figures, tables, and captions, appropriate titles, table of contents, and front cover with student information.
- The figures and code snippets (only if necessary) included in the report must be carefully chosen, for clarity and to meet the page limit.
- Do not insert unnecessary MATLAB code or the assignment questions in the report.
- For figures, (i) decide which type of plot is the most appropriate for each signal (e.g. solid line, non-connected points, stems), (ii) export figures in a correct format: without grey borders and with legible captions and lines, and (iii) avoid the use of screentshots when providing plots and data, use figures and tables instead.
- Avoid terms like *good estimate, is (very) close, somewhat similar*, etc - use formal language and quantify your statements (e.g. in dB, seconds, samples, etc).
- Note that you should submit two files to Blackboard: the report in PDF format and all the MATLAB code files compressed in ZIP/RAR format. Name the MATLAB script files according to the part they correspond to (e.g. ASP_Part_X-Y-Z.m).

Honour code:

> Students are strictly required to adhere to the College policies on students rights and responsibilities.The College has zero tolerance to plagiarism. Any suspected plagiarism or cheating (or prohibited collaboration on the coursework, see above) will lead to a formal academic dishonesty investigation. Being found responsible for an academic dishonesty violation results in a discipline file for the student and penalties, ranging from severe reduction in marks to expulsion from College.

1 Random signals and stochastic processes

Aims:

- To become acquainted with the generation of random signals in MATLAB.
- To investigate the effect of a linear system upon a random signal.
- To calculate and understand auto- and cross-correlation functions.

In most real-world signal processing applications the observed signals cannot be described by an analytical expression, however, by the Wold decomposition theorem, a stationary signal can be written as a sum of a deterministic signal and a stochastic signal. Our focus is on the statistical properties of such signals and their study paves the way for the design of estimation algorithms.

1.1 Statistical estimation

Using the MATLAB command `rand`, generate a 1000-sample vector $\mathbf{x} = [x[1], x[2], \ldots, x[1000]]^T$ where each sample $x[n]$ is a realisation of a uniform random variable $X \sim \mathcal{U}(0, 1)$ at time instant n. Plot the result and observe that despite its stochastic nature, $\mathbf{x}$ exhibits a degree of uniformity due to its time-invariant statistical properties, since the different samples $x[n], x[m]$ have been drawn from the same distribution. Such signals are referred to as **statistically stationary**. The vector $\mathbf{x}$ can be considered as a 1000-sample realisation of a stationary stochastic process X_n, whereby $X_n \sim \mathcal{U}(0, 1), \forall n$.

1. Calculate the expected value of X, denoted by $m = \mathbb{E}\{X\}$, also known as the *theoretical mean*. Using your 1000-sample realisation $\mathbf{x}$, also compute the *sample mean* using the MATLAB function `mean`, calculated as [5]

$$\widehat{m} = \frac{1}{N} \sum_{n=1}^{N} x[n],$$

where the circumflex denotes an estimate. Comment on the accuracy of the sample mean as an estimator.

2. Repeat the analysis for the standard deviation: calculate the theoretical value $\sigma = \sqrt{\mathbb{E}\{X - \mathbb{E}\{X\}\}^2}$ and also its sample estimate from data $\mathbf{x}$ using the MATLAB function `std` which computes the *sample standard deviation* as[1] [5]

$$\widehat{\sigma} = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (x[n] - \widehat{m})^2}. \qquad (1)$$

Comment on the accuracy of $\widehat{\sigma}$.

3. The *bias* of the sample mean estimation is given by $B = \mathbb{E}\{X\} - \widehat{m}$. Generate an ensemble of ten 1000-sample realisations of X, denoted by $\mathbf{x}_{1:10}$, and calculate the sample means $\widehat{m}_{1:10}$ and standard deviations $\widehat{\sigma}_{1:10}$ for each realisation. Plot these estimates of mean and standard deviation and comment on their bias, by showing how they cluster about their theoretical values. [5]
Note: In general, when plotting quantities that are not indexed by time stamps (as in this case, where the index is the 'realisation') there is no reason to connect the plotted points.

4. The mean and standard deviation describe second order statistical properties of a random variable, however, to obtain a complete statistical description it is necessary to examine the probability density function (pdf) from which the samples are drawn. Approximate the pdf of X by showing in the same plot the histogram of $\mathbf{x}$ (see `hist`), normalised by the number of samples considered, and the theoretical pdf. Comment upon the result, in particular, on whether the estimate appears to converge as the number of generated samples increases, and how the number of histogram bins considered affects the analysis. [5]

Note: As mentioned above, the theoretical pdf of X is $\mathcal{U}(0, 1)$.

5. Repeat Part 1–Part 4 using the MATLAB function `randn` to generate zero-mean, unit standard deviation, Gaussian random variables. [20]

[1] Note that the use of $(N - 1)$, instead of N, in Eq. (1) is known as *Bessel's correction*, which aims to remove the bias in the estimation of population variance, and some (but not all) bias in the estimation of the population standard deviation. This way, Eq. (1) gives an *unbiased* estimator of σ.

1.2 Stochastic processes

In real-world applications, time-varying quantities are usually modelled by a stochastic process, that is, an ordered collection of random variables. For **ergodic** processes, the theoretical mean can be approximated by time averages, however, for **non-ergodic** processes time averages do not necessarily match averages in the probability space and therefore the theoretical statistics are not always well approximated using observed samples.

To illustrate this concept, consider the **deterministic** sinusoidal signal $x[n] = \sin(2\pi 5 \times 10^{-3} n)$ corrupted by independent and identically distributed (i.i.d.) Gaussian noise $\eta[n] \sim \mathcal{N}(0, 1)$, to give $y[n] = x[n] + \eta[n]$. By averaging multiple realisations of the process $\mathbf{y} = [y[1], y[2], \ldots, y[N]]^T$, we aim to obtain reduced-noise estimates of the process $\mathbf{x}$, in terms of the Signal-to-Noise (SNR) ratio. If M independent realisations of the process $\mathbf{y}$, denoted by $\mathbf{y}_{1:M}$, are considered to compute an ensemble estimate, the variance of such an estimate is given by

$$\sigma_M^2 = \mathbb{E}\left\{\left(\mathbb{E}\{\mathbf{y}\} - \frac{1}{M}\sum_{i=1}^{M}\mathbf{y}_i\right)^2\right\} = \mathbb{E}\left\{\left(\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\eta}_i\right)^2\right\} \tag{2}$$

$$= \frac{1}{M^2}\mathbb{E}\left\{\left(\sum_{i=1}^{M}\sum_{j=1}^{M}\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right)\right\} = \frac{1}{M^2}\left(\sum_{i=1}^{M}\sum_{j=1}^{M}\mathbb{E}\left\{\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right\}\right), \tag{3}$$

where every noise sequence $\boldsymbol{\eta}_j$ comprises realisations of zero-mean and **uncorrelated** random variables $\eta[n]$. We know that $\mathbb{E}\left\{\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right\} = \sigma_{\boldsymbol{\eta}}^2$ iff $i = j$, and zero otherwise, hence

$$\sigma_M^2 = \frac{1}{M^2}\left(M\sigma_{\boldsymbol{\eta}}^2\right) = \frac{\sigma_{\boldsymbol{\eta}}^2}{M}. \tag{4}$$

Therefore, the SNR of an M-member ensemble estimate increases linearly with the number of members of the ensemble

$$SNR = \frac{\sigma_{\mathbf{y}}^2}{\sigma_M^2} = \frac{\sigma_{\mathbf{y}}^2}{\sigma_{\boldsymbol{\eta}}^2}M, \text{ and in dB: } SNR_{dB} = \log_{10}\left(\frac{\sigma_{\mathbf{y}}^2}{\sigma_{\boldsymbol{\eta}}^2}M\right)[dB]. \tag{5}$$

Figure 1 shows a realisation of $\mathbf{y}$, together with ensemble averages for $M = 10, 50, 200, 1000$ and the original deterministic signal $\mathbf{x}$. Additionally, the bottom plot shows the SNR computed from the ensemble averages and its theoretical value in Eq. (5). Observe that for nonstationary signals, time-average will not provide meaningful approximations of the process statistics (e.g. sample mean).

We now study three stochastic processes generated by the following MATLAB codes, which give an ensemble of M realisations of N samples for each stochastic process.

a)
```
function v=rp1(M,N);
a=0.02;
b=5;
Mc=ones(M,1)*b*sin((1:N)*pi/N);
Ac=a*ones(M,1)*[1:N];
v=(rand(M,N)-0.5).*Mc+Ac;
```

b)
```
function v=rp2(M,N)
Ar=rand(M,1)*ones(1,N);
Mr=rand(M,1)*ones(1,N);
v=(rand(M,N)-0.5).*Mr+Ar;
```

c)
```
function v=rp3(M,N)
a=0.5;
m=3;
v=(rand(M,N)-0.5)*m + a;
```

Run the above MATLAB codes and explain the differences between the time averages and ensemble averages, together with the stationarity and ergodicity of the process generated by the following steps:

1. Compute the ensemble mean and standard deviation for each process and plot them as a function of time. For all the above random processes, use $M = 100$ members of the ensemble, each of length $N = 100$. Comment on the stationarity of each process. [10]

2. Generate $M = 4$ realisations of length $N = 1000$ for each process, and calculate the mean and standard deviation for each realisation. Comment on the ergodicity of each process. [10]

3. Write a mathematical description of each of the three stochastic processes. Calculate the theoretical mean and variance for each case and compare the theoretical results with those obtained by sample averaging. [10]

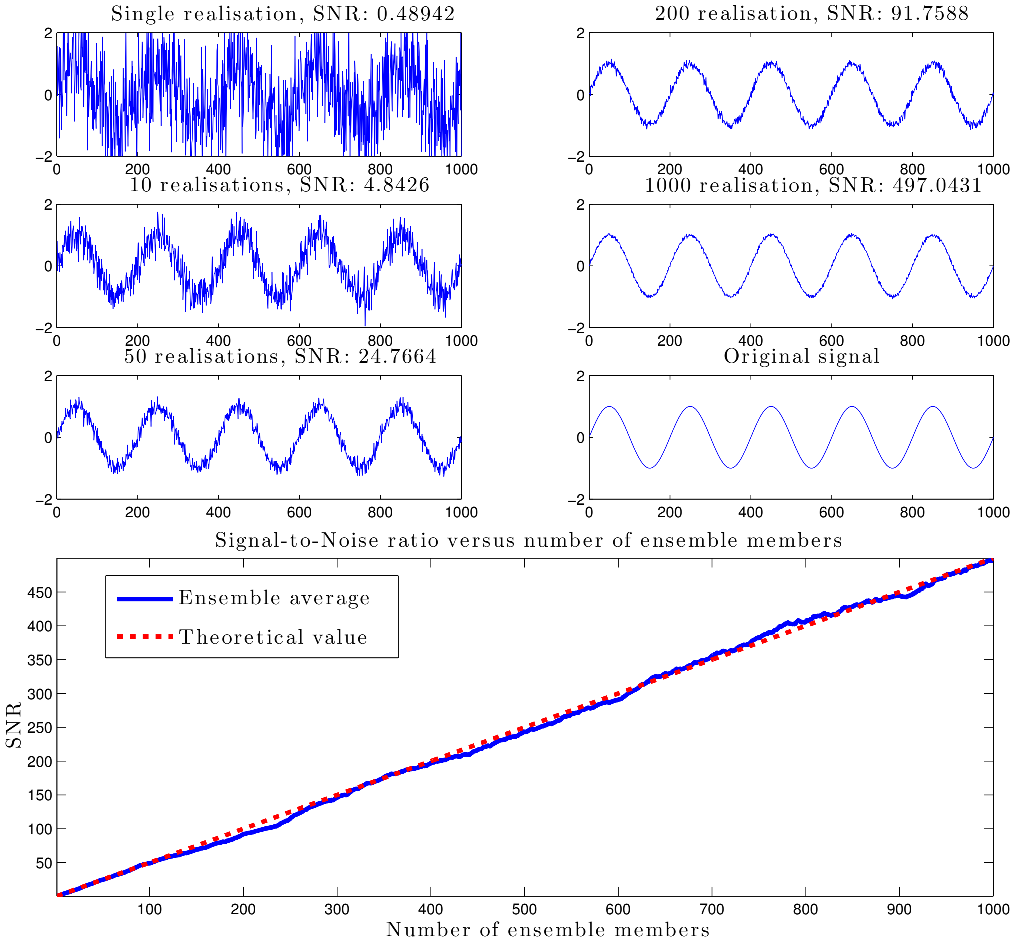

Figure 1: Ensemble estimates of a deterministic sequence corrupted by zero-mean uncorrelated white noise.

1.3 Estimation of probability distributions

Stochastic processes can be uniquely represented by their probability density functions (pdf), since the pdf contains all the statistical information about a random variable. When the process at hand is stationary, the statistical properties are time-invariant, and the process is uniquely defined by a time-invariant pdf. Furthermore, for ergodic processes such distributions can be estimated via time averages.

Design and implement a pdf estimator and test it on the three stochastic processes studied in Part 1.2.

1. Write an `m`-file named `pdf`, which gives an estimate of the pdf of a collection of samples based on the MATLAB function `hist`. Test your code for the case of a stationary process with a Gaussian pdf, `v=randn(1,N)`. The data length `N` should be at least 100. [10]

2. For those processes in Part 1.2 (a,b,c) that are stationary and ergodic, run your MATLAB code to approximate the pdf for $N \in \{100, 1000, 10000\}$. Compare the estimated densities with the theoretical pdf using the MATLAB `subplot` function, and comment on the results as data length N increases. [10]

3. Is it possible to estimate the pdf of a nonstationary process with your function `pdf`? Comment on the difficulties encountered for the case when the mean of a 1000-sample-long signal changes from 0 to 1, after sample point $N = 500$. Explain how you would compute the correct pdf in this case? [10]