ELEE4130 Project 1

Random Variable and Random Process

Due September 30, 2022

1 Main Topics

- Generation of random variables
- Autocorrelation of WSS random processes
- Linear filtering of random processes

2 Computer Assignment

1. (5pts) Generate a set of 1000 i.i.d. uniform random numbers in the interval [-2, 2] by using the MATLAB function `rand(1,N)`. Plot the histogram and the CDF using the generated sequence of numbers.
   Note: i) MATLAB function `hist(...)` can be used to plot histogram, use
   `>> help hist`
   in MATLAB command window to check the usage.
   ii) How to plot CDF using the generated sequence of numbers: plotting CDF using the sample sequence generated is called *empirical* CDF. There are multiple ways to do that. Two examples are given here:

   (a) Using the histogram you created: the histogram (counting the occurrence of values in a bin) can be viewed as empirical PDF, if y-axis is divided by the total number of samples (e.g. it shows the frequency of occurrence of values in a bin which is empirical PDF for x in that bin). Then, you can sum over the y-values over the bins from left to right, to"integrate" PDF over x to obtain empirical CDF.

   (b) Another simple way to plot empirical CDF is to sort your generated numbers in vector in ascending order using `sort` function into ${}_{\text{sort}}$. Then, you can directly plot empirical CDF as
   `>> plot(`${}_{\text{sort}}$`, [1:length(`${}_{\text{sort}}$`)]/length(`${}_{\text{sort}}$`) )`
   (think about why this gives you the empirical CDF)

2. (10pts) Generate a set of 1000 i.i.d. Gaussian random numbers with zero mean and unit variance using MATLAB function `randn(1,N)`. Plot the histogram and the CDF of the generated numbers.
   Note: In determining the histogram, the range of the random numbers may be divided into subintervals (bins) of width $\sigma/5$, with bin boundaries as ..., $-5\sigma/10, -3\sigma/10, -\sigma/10, \sigma/10, 3\sigma/10,$

$5\sigma/10$..., etc., and the center bin covers the range $x \in [-\sigma/10, \sigma/10]$, where σ^2 is the variance of the random numbers (in this problem $\sigma^2 = 1$).

3. (10pts) Let the sequence of 1000 uniform random numbers independently generated in Task 1 be $\{X_1, \cdots, X_{1000}\}$. It forms a realization of the random process.
   i) Determine numerically the autocorrelation function $R_X(m)$ of this random process $\{X_1, \cdots, X_{1000}\}$, for $m = 0, \cdots, 50$. Plot $R_X(m)$, $m = 0, \cdots, 50$, where x-axis is m, and y-axis is $R_X(m)$.
   ii) Determine the power spectrum density (PSD) $S_X(f)$ of $\{X_1, \cdots, X_{1000}\}$ by computing DFT of vector $\chi = [R_X(0), \cdots, R_X(50)]$. Plot $S_X(f)$ versus f.

   Note:

   - Use provided m-file Rx_est.m to compute the sample autocorrelation function;
   - In MATLAB, in the folder containing Rx_est.m, use "`help Rx_est`" to check the usage of this function.
   - (Optional: extra 3 pts) Since the sample autocorrelation function computed numerically for different realization of a random process can vary significantly, you may consider to repeat Task 1 several times to generate different realizations of sequence $\{X_1, \cdots, X_{1000}\}$, and compute autocorrelation $R_X(m)$ for each such sequence; then average the sample autocorrelation functions to get a more accurate result.
   - For $S_X(f)$: In MATLAB (and any commercial system), DFT is efficiently computed using fast Fourier transform (`fft`). Use MATLAB function `fft(`χ`)` to compute DFT of χ, then for the output S_X, use MATLAB function `fftshift(`S_X`)` to shift the zero-frequency component ($f = 0$) to the center of spectrum. Then plot the output.

4. (10pts) (Linear filtering of random process) Consider the i.i.d. sequence of uniform random numbers $\{X_1, \cdots, X_{1000}\}$ generated in Task 1. Let this sequence be the input, passing through a linear filter with impulse response $h[n]$ given as

$$h[n] = \begin{cases} (0.95)^n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

   The recursive equation that describes the output of this filter, as a function of the input, is

$$Y_n = 0.95Y_{n-1} + X_n, \quad n \geq 0, \text{and} \quad Y_0 = 0.$$

   Obtain output sequence $\{Y_1, \cdots, Y_{1000}\}$ based on this linear filter. For this sequence, compute the autocorrelation function $R_Y(m)$, for $m = 0, \cdots, 50$ (Similar to Task 3; use Rx_est.m). Plot $R_Y(m)$ over $m = 0, \cdots, 50$, and plot the corresponding PSD $S_Y(f)$ (use the same instruction as in Task 3)

5. **Simulating Binary Signal Transmission in Gaussian Noise** (15 pts)

   Let N be a zero-mean Gaussian random variable as $N \sim (0, \sigma^2)$. Let $X \in \{a, b\}$ be a binary random variable with two possible values with $P(X = a) = P(X = b) = 0.5$. X is the input symbol and N is the receiver noise, and X and N are independent. The received signal is

$$Y = X + N. \tag{1}$$

   Define the signal-to-noise ratio (SNR) (in dB) as

$$\text{SNR[dB]} = 10 \log_{10} \frac{\mathbb{E}[X^2]}{\sigma^2} \text{ [dB]}$$

(a) Let $a = -1$, $b = 1$, and $\sigma^2 = 1$. Compute $\mathbb{E}[X^2]$. Obtaining 10^8 samples of Y, by generating a sequence of X and a sequence of N, and obtaining Y based on eqn (1).

(b) Based on the obtained sequence of Y, estimate X using a threshold detector as follows: Let $\hat{X}$ be your estimate of X. Determine $\hat{X}$ as

$$\hat{X} = \begin{cases} 1 & \text{if } Y > 0 \\ -1 & \text{if } Y \leq 0. \end{cases} \tag{2}$$

Count the percentage of wrong estimate, $\hat{X} \neq X$. Compare this percentage value (we call it the "simulated error") with the probability of error (in theory) you obtained in Problem 9(c) of Assignment 1 (Q-function). Does your simulation result match with the theory?

(c) Repeat the above simulation for SNR = [0 : 1 : 10]dB. To do so, for each SNR value, change the noise variance σ^2 accordingly, and regenerate sequence of N (keep the same input X sequence). Plot the percentage of wrong estimates (error) as a function of SNR (x-axis is SNR[dB], and y-axis is the error percentage). Use log scale for your y-axis (use MATLAB function `semilogyy`).

3 Project Report

- Work individually.
- Project report should include the required figures, MATLAB code (with appropriate commentation of the code), and observations and/or conclusions that you may draw from the simulation results. Make sure to check grammar and spelling.
- When plotting figures, clearly scale and label your plots (both x- and y-axis; specify the unit whenever exists). When more than one curve is shown in a figure, clearly put legend to describe each curve is for what.
- (5pts) Technical writing (with clear report organization of each task results, clarity in description and discussion, and appropriate display of figures).