EDC310 Digital Communications

Practical Assigment 1

12 August 2020

Compiled by Prof HC Myburgh

Introduction

You are to develop a simulation platform[1] for a BPSK, 4QAM, 8PSK and 16QAM communication system transmitting information over an additive white Gaussian noise (AWGN) channel. To realise the simulation platform, you will need a number of building bocks - a uniform random number generator to generate random bits and a Gaussian random number generator to generate Gaussian noise.

Question 1 [15]

Develop a uniform random number generator, able to generate random numbers in the range (0,1) using the Wichmann-Hill algorithm [1]. Verify your random number generator by comparing its statistics (μ, σ, σ^2) to theoretical statistics. Plot the PDF of the uniform random number generator and compare it to the theoretical PDF.

Question 2 [15]

Develop a Gaussian random number generator, able to generate Gaussian random numbers with $\mu = 0$ and $\sigma = 1$, using the Marsaglia-Bray algorithm [2]. Verify your random number generator by comparing its statistics (μ, σ, σ^2) to theoretical statistics. Plot the PDF of the Gaussian random number generator and compare it to the theoretical PDF.

Question 3 [40]

Design and develop a simulation platform to simulate the performance for BPSK, 4QAM, 8PSK and 16QAM modulation trough an AWGN channel. Evaluate the bit-error rate (BER) and symbol-error rate (SER) performance in the range $E_b/N_0 \in [-4{,}12]$ dB and plot the BER and SER using the *semilogy* command in *Python*.

1. Use your uniform random number generator to generate random bits. A value larger than 0.5 is a 1 bit; a value less than or equal to 0.5 is a 0 bit. This will ensure a 50/50 0/1 bit ratio.

2. Map the bits to symbols using the respective modulation constellation maps.

3. Add noise to the symbols using your Gaussian random number generators as follows:

$$r_k = s_k + \sigma n_k, \tag{1}$$

   where n_k is the kth complex zero mean, unity variance, Gaussian random variable. Since

$$SNR = 10\log\left(\frac{|a|}{2\sigma^2}\right) = 10\log\left(\frac{1}{2\sigma^2}\right) = \frac{E_b}{N_0}, \tag{2}$$

[1] All software must be developed in *Python* 3.

$$\sigma = \frac{1}{\sqrt{10^{\frac{E_b}{10N_0}} 2f_{bit}}} \tag{3}$$

where $f_{bit} = log_2(M)$ and M is the number of symbols in the modulation constellation.

4. Detect the received symbols by calculating the Euclidean distance between the received symbol and all of the symbols in the constellation (see (4) below). The symbol with the smallest Euclidean distance is the most probable transmitted symbol.

$$\Delta^{(i)} = |r_t - s^{(i)}|^2 \tag{4}$$

5. Compare the detected symbols to the transmitted symbols and count the symbols errors.
6. Determine the SER by dividing the number of symbol errors by the number of transmitted symbols.
7. Convert the detected symbols back to bits and compare the detected bits to the transmitted bits and count the bit errors.
8. Determine the BER by dividing the number of bit errors by the number of transmitted bits.

Deliverables

- Write a report using LaTeX(preferred) or MS Word. Please use a professional-looking template.
- Answer Question 1 through 3 and report on your findings. Be concise and use proper grammar.
- Include your code as an appendix.

Instructions

- All reports must be in PDF format and be named report.pdf.
- Name the source code files question_X.py, where X indicates the question number.
- Place the software in a folder called SOFTWARE and the report in a folder called REPORT.
- Add the folders to a zip-archive and name it EDC310_prac1_groupnr.zip.
- The submission deadline is Friday 28 August at 12:00. Submission instructions to follow.

- Do not copy! The copier and the copyee (of software and/or documentation) will receive zero and disciplinary action will follow for both parties.
- For any questions, please make an appointment with me Prof Myburgh or Mr Dube.
- Make sure that you thoroughly discuss the results that are obtained. This is a large part of writing a technical report.

References

[1] B. Wichmann and D. Hill, "*Building a random number generator*", Byte, pp 127-128, March 1987.

[2] G. Marsaglia and T.A. Bray, "*A convenient method for generating normal variables*", SIAM Rev., Vol. 6, pp 260-264, 1964.