EDC 310

Practical Assigment 1

24 July 2018

Compiled by Herman Myburgh and Alistair Yan

Scenario

You are to develop a simulation platform[1] for a BPSK- and QPSK communication system over an additive white Gaussian noise (AWGN) channel. To realise the simulation platform, you will need a number of building bocks.

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 [30]

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

1. Use your uniform random number generator to generate random bits.
2. Map the bits to symbols using the respective BPSK- and QPSK modulation maps.
3. Add noise to the symbols as follows:

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

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

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

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

where $f_{bit} = 1$ for BPSK and $f_{bit} = 2$ for QPSK.

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

4. Detect the received symbols by comparing each to each symbol on the constellation map.
5. Convert symbols back to bits.
6. Compare transmitted bits to received bits and count the bit errors.
7. Determine the BER by dividing the number of errors by the number of transmitted bits.

Deliverables

- Write a report using LaTeX. Reports that are not written using LaTeX will not be marked.
- 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.Y, where X indicates the question number and Y is the platform you used.
- 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_studnr1.zip.
- All reports and simulation software are to be e-mailed to *edc310.2018@gmail.com* no later than 16h00 on 16 August 2018. No late submissions will be accepted.
- Hard copies of your report will be submitted in the tutorial session on the 16th of August

Additional Instructions

- Do not copy! The copier and the copyee (of software and/or documentation) will receive zero for both the software and the documentation. Z-e-r-o.
- For any questions, please make an appointment with me on *u14006007@tuks.co.za*.

- Make sure that you discuss the results that are obtained. This is a large part of writing a technical report.
- You are allowed to use Python's RNG as a comparison, but under no circumstances may you use these in the solution to Questions 1 - 3.

Marking

Your report will be marked as follows:

- 60% will be awarded as indicated for Questions 1 to 3. This only entails the successful completion of the simulations and the sub-sequential graphs.
- 40% will be awarded for the overall report quality. This includes everything from the report structure, grammar and discussion of results.

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.