Igor Povarich
Comp Eng 5430
09/29/2021

Implementation

Each test ran for approximately 20 seconds in real time to keep the tests relatively similar and reduce any end-effect errors. After running a simulation, the script1.sh was run (appendix) to execute the instructed commands: a hexdump to “in.txt” to record the transmitted symbols, a hexdump to “out.txt” to record the received symbols, a diff command to “diff.txt” to record the differences between the two files, and a grep command to output the bit error rate.

A script was used to evaluate the two files and trim both the beginning and end of the files to help with alignment and reduce end-effect errors of the simulations, hence, simulations that show 0 error had some errors, but those were identified as spurious ones at the beginning and/or end of diff.txt.

Table 1: Simulation Levels

	Control	A	B	C
Noise Voltage Levels	0	0.1	0.3	0.5
Attenuation	1	0.75	0.5	0.1
Encoding	BPSK/QPSK			

Simulation Raw Data

The raw data recorded was the number of symbol errors per simulation run.

Table 2:Symbol Errors per Simulation

Symbol Errors per Simulation								
Encoding →	BPSK				QPSK			
Noise Voltage →	0	0.1	0.3	0.5	0	0.1	0.3	0.5
Attenuation ↓								
1	0	0	0	513	0	0	924	47347
0.75	0	0	3	19615	0	0	16781	248539
0.5	0	0	7471	380812	0	2	200003	772097
0.1	0	439161	3572562	3738610	0	796914	1975914	2027590

To support the simulation error data, the total number of symbols were also recorded.

Igor Povarich
Comp Eng 5430
09/29/2021

Table 3: Symbols per Simulation

Symbols per Simulation								
Encoding →	BPSK				QPSK			
Noise Voltage →	0	0.1	0.3	0.5	0	0.1	0.3	0.5
Attenuation ↓								
1	0	0	0	5242886	0	0	2621448	2632171
0.75	0	0	5767024	4980609	0	0	2493305	2686634
0.5	0	0	5244897	5378030	0	2621319	2669491	2877257
0.1	0	5403274	7788290	7948470	0	2885960	3703235	3740714

Using the following equation, the time per simulation was retrieved for each run.

$$Time\ per\ Simulation = \frac{Num\ of\ symbols}{32k\frac{symbols}{second}}$$

Using this Simulation Time, the symbol error rates were calculated.

$$Symbol\ Error\ Rate = \frac{Symbol\ Errors}{Simulation\ Time}$$

Table 4: Symbol Error Rate

Symbol Error Rate								
Encoding →	BPSK				QPSK			
Noise Voltage →	0	0.1	0.3	0.5	0	0.1	0.3	0.5
Attenuation ↓								
1	0.00	0.00	0.00	3.13	0.00	0.00	11.28	575.61
0.75	0.00	0.00	0.02	126.02	0.00	0.00	215.37	2960.30
0.5	0.00	0.00	45.58	2265.88	0.00	0.02	2397.50	8587.03
0.1	0.00	4337.36	14678.70	15051.39	0.00	8836.31	17074.06	17345.05

Discussion and Plots

During the simulation runs, many of the runs showed zero error, which may have been a bug in the methodology or calculations. For example, all the ones with zero noise voltage showed zero error, as can be seen in figure 2. However, the general trend is still visible. As the noise voltage increases, the symbol error rate increases. This is also true as attenuation increases (i.e. the signal amplitude decreases). The SER peaks at the highest noise voltage and the lowest attenuation as would be expected. So, a lot of noise can be overcome by increasing signal amplitude and a weak signal can still be received in a low-noise environment. This relationship can be seen in Table 4 and Figure 3. Finally, QPSK seems to have

slightly higher SER. This is probably due to the increased encoding/decoding that happens with the 4 symbols per bit in QPSK.

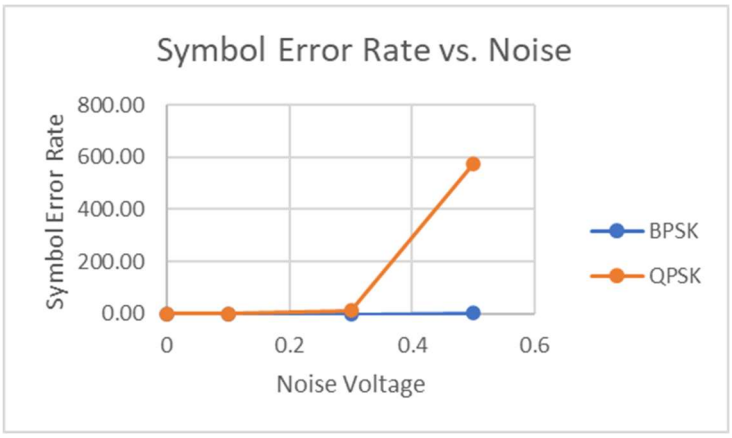

Figure 1: SER vs noise

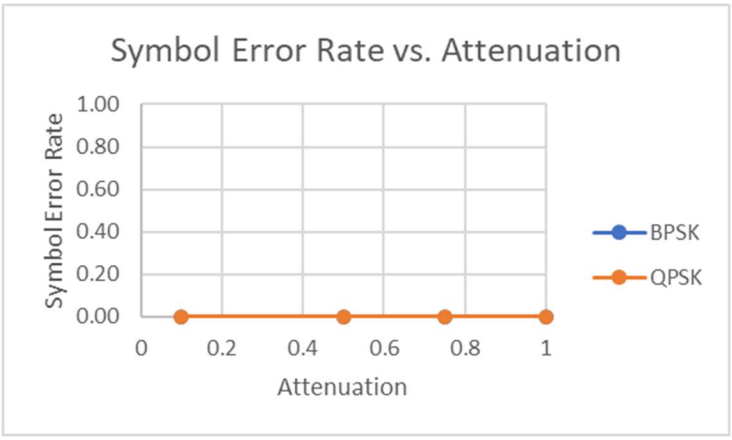

Figure 2: SER vs Attenuation

Igor Povarich
Comp Eng 5430
09/29/2021

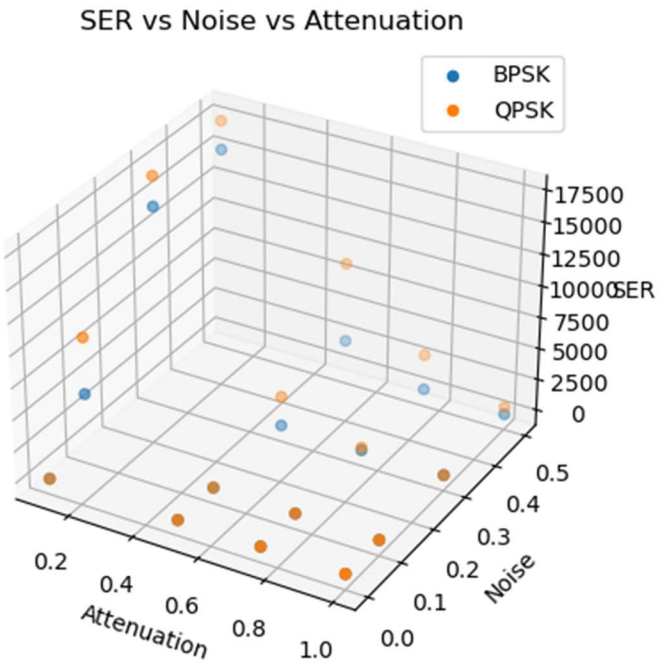

Figure 3: 3d plot of SER, Attenuation and Noise

Appendix

Script1.sh for running the main commands plus some evaluation helpers

```
 GNU nano 4.8                                    script1.sh
#!/bin/sh

# Author: Igor Povarich

# hexdump input file
hexdump sent_symbols.bin -v -e '1/1 "%02X\n"' > in.txt
# clip the last ## lines of in.txt
sed -i "$(( $(wc -l <in.txt)-55)),$ d" in.txt
# Note: skip first 46 lines from out.txt due to corrupted bytes
hexdump recieved_symbols.bin -v -s 46 -e '1/1 "%02X\n"' > out.txt
# difference bw two files
sdiff in.txt out.txt > diff.txt
# print BER
grep -o '<\|>\||' diff.txt | wc -l

# print truncated outputs
echo " Top of diff.txt:"
head -10 diff.txt
echo ""
echo "Bottom of diff.txt:"
tail -20 diff.txt
```

Igor Povarich
Comp Eng 5430
09/29/2021

Example output from script1.sh

```
ipovaric@LAPTOP-9DGKJO60:~/repos/wireless_comm_lab/Lab3$ ./script1.sh
5
2752391 diff.txt
 Top of diff.txt:
01                                                      <
00                                                      <
02                                                        02
01                                                        01
01                                                        01
00                                                        00
03                                                        03
02                                                        02
00                                                        00
02                                                        02

Bottom of diff.txt:
01                                                        01
01                                                        01
02                                                        02
00                                                        00
03                                                        03
01                                                        01
02                                                        02
03                                                        03
00                                                        00
01                                                        01
02                                                        02
00                                                        00
01                                                        01
01                                                        01
03                                                        03
00                                                        00
02                                                      <
01                                                      <
02                                                      <
02                                                        02
```