1 SNR estimation: spectrum-method

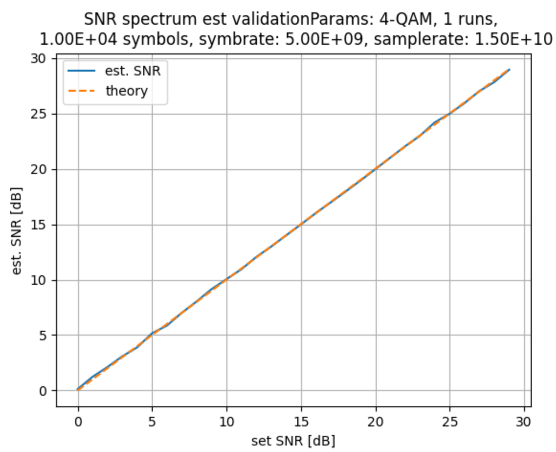
Since many system investigations or even individual functions such as MRC combining depend on the correctly investigated SNR, it is necessary to establish an SNR estimator that is as accurate as possible. This was attempted using the spectrum method within the toolbox. Analogous to the interpolation method used to determine the OSNR in the optical domain, an SNR estimator based on the power spectrum was established. This calculates the powers and the resulting signal-to-noise ratio based on signal and noise ranges of the signal. Thus, this estimator is subject to the following conditions for reasonable use:

- because these algorithm working in the frequency domain, we need a oversampling of 2 sample per symbol as the bare minimum of the signal
- the quality of the estimators results are highly conditioned by setting the right limits of signal and noise ranges
- also the quality should be corresponds to the window length of samples used for frequency domain transformation
- therefore the estimation quality should be good over a wide SNR range in terms of the fixed signal and noise ranges

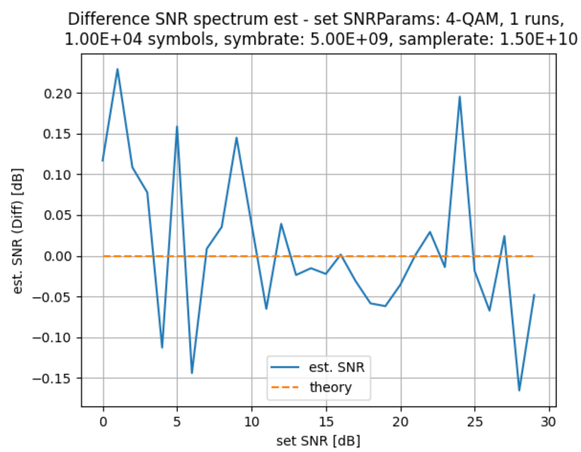
Determining the signal and noise limits is sometimes complicated. These were created using the available plots, for the present simulation, which was performed with an oversampling of 3sps at a symbol rate of 5GHz. The exact dynamically programmed limits can be taken from the simulation script.

1.1 Behavior in different SNR regimes

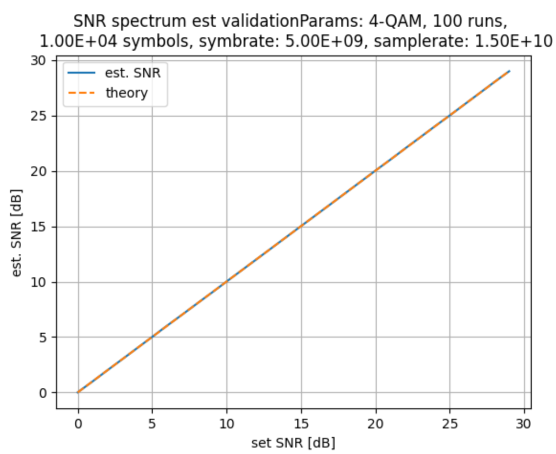
To verify that the SNR estimator works over a wide range of SNRs, as theoretically assumed, a simulation was performed. As can be seen in the figures 1, this is the case. Under the boundary parameters, which can be seen in the figure headings, good results of the estimator are evident. It is important here to compare between a different number of runs of the estimator. If one compares 1 estimation runs with the mean of 100 estimation runs, the estimator draws a good overall, especially in the poor SNR range. The Y-axis of the difference plot was chosen here differently between the trials with different runs, since thus also for 100 runs an approximate numerical value is readable, even if this variant appears less impressive.



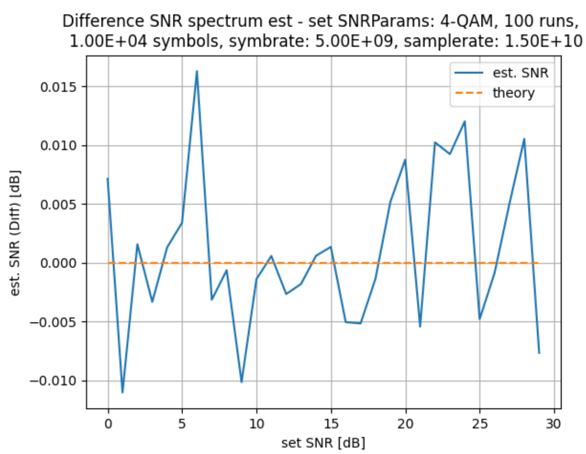
(a)



(b)



(c)

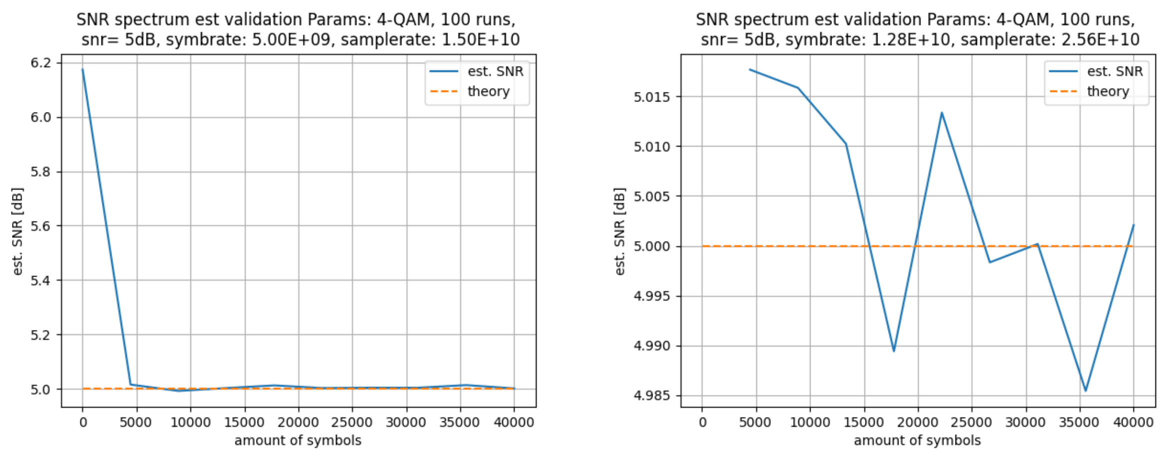


(d)

Figure 1: (a/b) 1 run per SNR value (c/d) mean of 100 runs per SNR value

1.2 Behavior in terms of different block size

To estimate how many symbols or data points are needed for the SNR estimator to function, a simulation was also made. The results can be seen in Figure 2. Here it becomes clear that with the randomly chosen settings of 5e9 as symbolrate and 15e9 as samplerate a minimum number of 5000 samples must be given in order to allow a reasonable estimation. Also visible within the other representation (b) are more realistic values for the symbol and sample rate, as these could also be used in a possible experiment. On the one hand, it becomes clear that the estimator does not return an estimated value in every case (here, for example, NaN at position ”4000 amount of samples”). This is due to the fact that the estimator estimates a negative linear SNR in this case, and thus no output of the log10 operation occurs. This must be considered in possible applications. It also becomes clear that with increasing sample rate/symbol rate, the estimation error obviously decreases in the range of ¡5000 symbols in relation to smaller sample and symbol rates. A test with lower sample and symbol rates indicated that there is no improvement or deterioration in the estimation result relative to the randomly chosen settings in (a). It can therefore be concluded, as a first approximation, that the estimator with a symbol count of ¿5000 symbols can produce good results.



(a) (b)

Figure 2: (a) randomly choosen settings, (b) settings from experiment approach