EQ2410 - Advanced Digital Communications

Project 1 : Channel Equalization

Baptiste Cavarec
940321-T197

Hugo Lime
920707-4635

Adrien Anxionnat
######-####

February 14, 2016

1 Problem 1

1.a Identification of signals

- `Vect_1`: Random QPSK data symbol sequence
  $\texttt{Vect\_1} \in \{1+i, 1-i, -1+i, -1-i\}^{\texttt{N\_symbols}}$
- `Signal_1`: `Vect_1` upsampled by factor `T_sym`/ `delta_t` and normalized
  $\texttt{Signal\_1}(1+k*\texttt{T\_sym/delta\_t}) = \texttt{Vect\_1}(1+k)/\texttt{delta\_t}$ with $k \in [0, \texttt{N\_symbols}-1]$
- `Signal_2`: Complex baseband data transmitted waveform
  $\texttt{Signal\_2} = \texttt{Signal\_1} \star \texttt{Filter\_1}$
- `Signal_3`: Complex baseband data received waveform
  $\texttt{Signal\_3} = \texttt{Signal\_2} \star \texttt{Filter\_2}$
- `Signal_4`: Complex baseband noise waveform with two-sided variance N_0
- `Signal_5`: Complex baseband total received waveform
  $\texttt{Signal\_5} = \texttt{Signal\_3} + \texttt{Signal\_4}$
- `Signal_6`: Complex baseband total received waveform matched filtered
  $\texttt{Signal\_6} = \texttt{Signal\_5} \star \texttt{Filter\_4}$
- `Vect_2`: Sampler output received symbol sequence after synchronization
  $\texttt{Vect\_2}(k) = \texttt{Signal\_6}(k*\texttt{T\_sym/delta\_t} + \delta_{offset})$ (`Signal_6` downsampled)

1.b Identification of filters

- `Filter_1`: Transmitter filter
- `Filter_2`: Channel filter
- `Filter_3`: Transmitter and Channel chain filter
- `Filter_4`: Matched filter of `Filter_3`, can be used as receiver filter
- `Filter_5`: Transmitter, channel and receiver chain filter, used to simulate the overall system response

1.c Normalization coefficient

In the program, we use a small time resolution (delta_t) to represent continuous-time signals. We consider the signals to be constant over these small intervals. With this approximation, the integrals, used in the convolutions for example, are transformed into sums in the following way:

$$(f \star g)(m\delta_t) = \int_{-\infty}^{+\infty} f(t)g(m\delta_t - t)dt = \sum_{k=-\infty}^{\infty} \left(\int_{k\delta_t}^{(k+1)\delta_t} f(t)g(m\delta_t - t)dt \right) \approx \delta_t \sum_{k=-\infty}^{\infty} f(k\delta_t)g((m-k)\delta_t)$$

And we can notice that, compared to the discrete-time convolution formula used by Matlab, the last sum is multiplied by delta_t.

The factor 1/delta_t when we generate Signal_1 is another normalization used to keep the signal power indepedant of the time resolution. In the same manner we have:

$$\begin{aligned} P &= \frac{1}{N_{symbols}T_{sym}} \int_0^{N_{symbols}T_{sym}} |S_1(t)|^2 dt \\ &\approx \frac{1}{N_{symbols}T_{sym}} \sum_{k=0}^{(N_{symbols}-1)T_{sym}\delta_t} |S_1(k\delta_t)|^2 \delta_t \\ &\approx \frac{{\delta_t}^2}{N_{symbols}T_{sym}\delta_t} \sum_{k=0}^{(N_{symbols}-1)T_{sym}\delta_t} |S_1(k\delta_t)|^2 \\ &\approx \frac{1}{N_{symbols}T_{sym}\delta_t} \sum_{k=0}^{(N_{symbols}-1)T_{sym}\delta_t} |\delta_t^2 S_1(k\delta_t)|^2 \end{aligned}$$

And by dividing by 1/delta_t, the discrete-time signal power of Signal_1 is independant of delta_t and equal to the one of Vect_1.