TELECOMMUNICATION PRACTICAL WORK

MoST ESECA

2020-2021

1 Study of baseband modulators, spectral efficiency

This part of the work is dedicated to the study of baseband modulators. For that purpose, you will have to implement some on Matlab and compare them in terms of spectral efficiency.

1.1 Modulators to be compared

1. Modulator 1:
   - Mapping: binary symbols a_k with zero average.
   - Shaping filter: rectangular of T_s duration.
2. Modulator 2:
   - Mapping: 4-ary symbols a_k with zero average.
   - Shaping filter: rectangular of T_s duration.
3. Modulator 3:
   - Mapping: binary symbols a_k with zero average.
   - Shaping filter: edge of T_s duration.
4. Modulator 4:
   - Mapping: binary symbols a_k with zero average.
   - Shaping filter: square root raised cosine of roll off factor $\alpha = 0.5$.

1.2 Implementation and study

Previous modulators will have to be implemented with a sampling frequency $F_e = 24000$ Hz and to transmit a bit rate $R_b = 3000$ bits per second.
You can use the *randi.m* function to generate the binary information to transmit.
In digital, the generation of the sum of Delta functions spaced from T_s and weighted by the symbols a_k will consist in inserting $N_s - 1$ zeros between two symbols a_k, if N_s represents the number of generated samples per symbol (see figure 1). N_s is also called the oversampling factor: $T_s = N_s T_e$, if T_e denotes the sampling period. N_s will be defined so as to generate a digital signal fulfilling Shannon condition.
You can use the following Matlab instructions to generate the filers impulse responses (FIR filters: see digital signal processing course) :

- $h = ones(1, N_s)$ to generate a rectangular impulse response of $T_s = N_s T_e$ duration.
- $h = [ones(1, N_s/2)\ -ones(1, N_s/2)]$ to generate an edge of $T_s = N_s T_e$ duration.
- $h = rcosdesign(0.5, 8, N_s)$; to design, for example, a square root raised cosine filter with an impulse response duration of length $N = 8 \times N_s + 1$ samples (or coefficients), a roll off factor $\alpha = 0.5$ and a symbol period $T_s = N_s T_e$.

You can use the following Matlab instruction to filter: *x=filter(h,1,sum_weighted_delta_functions)* (Finite Impulse Response (FIR) Filters: see digital signal processing course).

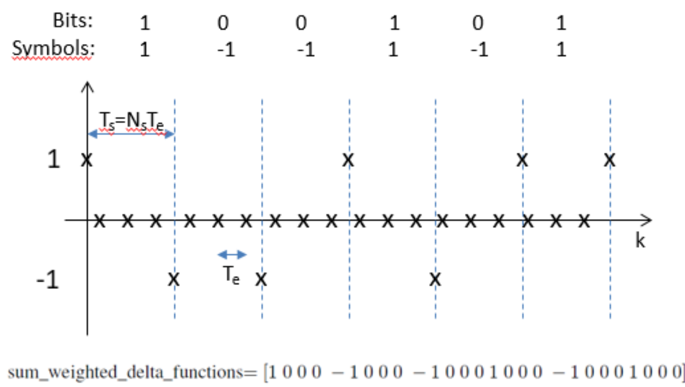

Figure 1: Example for the sum of Delta functions spaced from T_s and weighted by the symbols a_k. Here $N_s = 4$

.

1. For each modulator you will have to:
   - Plot the transmitted signal with the x-axis in seconds.
   - Plot the transmitted signal power spectral density (PSD) with the x-axis in Hz, using the following Matlab instructions:
     PSDx=10 $*$ log10(pwelch(x));
     plot(linspace(0,Fe/2,length(PSDx)),PSDx);
2. Modulators comparison: plot, on the same figure (using *hold.m*) and for the same bit rate, the power spectral densities of the signals generated by the 4 modulators. You can normalize your PSDs before plotting. Class the studied modulators in increasing order of spectral efficiency. Explain your classification.

2 Study of the whole transmission channel : interference problem - Nyquist criterion, impact of the noise - matched filtering

The objective of this part of the work is to study a whole communication transmission channel : the interference problem and the influence of the Nyquist criterion, the impact of the noise and the influence of matched filtering.

2.1 Digital communication channel to implement

The channel will have to be implemented with the following parameters:

- Sampling frequency: $F_e = 24000$ Hz,
- Bit rate: $R_b = \frac{1}{T_b} = 3000$ bits per second.

and with the following elements:

- Mapping: binary symbols a_k with zero average.
- Oversampling: generation of the sum of Dirac functions spaced from N_s samples and weighted by the symbols a_k.
- Shaping filter: rectangular of T_s duration ($h = ones(1, N_s)$).
- Propagation channel: no noise, no filter for the first exercise, no noise but a filter for the second one (see later), no filter but an additive white Gaussian noise for the third one (see later).
- Receiver filter: rectangular of T_s duration ($h = ones(1, N_s)$).
- Sampling at $n_0 + mN_s$, n_0 denoting the number of the sample to collect in T_s duration, composed of N_s samples in digital ($t_0 = n_0T_e$ and $T_s = N_sT_e$), n_0 will be a parameter to test.

- Decisions on the symbols using a 0 threshold thanks to the Matlab *sign.m* function.
- Demapping using the following Matlab expression on the previous retrieved symbols: *(symbols+1)/2* (applicable if bit 0 has been replaced by -1 and 1 has been replaced by 1 in the mapping process).
- Bit error rate estimation.

2.2 Study without propagation channel

You will first have to study the proposed communication channel without the propagation channel, meaning no noise but also no filter due to the propagation channel, just the modulator and the demodulator.

1. Plot the whole transmission channel impulse response g, by plotting the convolution between the shaping filter and the receiver filters impulse responses (h and h_r). You can use Matlab *conv.m* function.
2. Determine, from g tracing, the optimal sampling instant n_0 (allowing to obtain sampling instants $n_0 + mN_s$ without inter symbol interference).
3. Plot the eyediagram at the receiver filter output.
4. From the plotted eyediagram, retrieve the optimal value for n_0.
5. Sample at $n_0 + mN_s$ with the chosen n_0 and check that the bit error rate (BER) is zero.
6. Sample at $n_0 + mN_s$ with $n_0 = 3$. Compute the simulated BER and explain the obtained result.

2.3 Study with a limited bandwidth propagation channel but no noise

We will now add a filter representing a propagation channel with a limited bandwidth BW but no noise. For that, we will consider the previous implemented channel, with a sampling at the optimal time instants, and add a filter representing the propagation channel.

The impulse response of the propagation channel (lowpass filter) can be obtained, on Matlab, using: $hc = (2 * fc/Fe) * sinc(2 * (fc/Fe) * [-(N-1)/2 : (N-1)/2])$, where fc represents the cutoff frequency (BW here) and N the order of the filter (see FIR synthesis in the Digital Signal Processing course).

1. For $BW = 4000$ Hz:
   - Represent, on the same plot, $|H(f)H_r(f)|$ and $|H_c(f)|$, where $H(f)$ is the frequency response of the shaping filter, $H_r(f)$ is the frequency response of the receiver filter and $H_c(f)$ is the frequency response of the channel filter. $|.|$ is given by the Matlab function *abs.m* and the Fourier transform by *fft.m*.
   - Is the Nyquist criterion able to be verified with this propagation channel ? Explain your answer.
   - Plot the eyediagram at the receiver filter output. From the plotted eyediagram, can you say if the Nyquist criterion will be able to be verified in the transmission channel ? Explain your answer.
2. For $BW = 1000$ Hz:
   - Represent, on the same plot, $|H(f)H_r(f)|$ and $|H_c(f)|$, where $H(f)$ is the frequency response of the shaping filter, $H_r(f)$ is the frequency response of the receiver filter and $H_c(f)$ is the frequency response of the channel filter. $|.|$ is given by the Matlab function *abs.m* and the Fourier transform by *fft.m*.
   - Is the Nyquist criterion able to be verified with this propagation channel ? Explain your answer.
   - Plot the eyediagram at the receiver filter output. From the plotted eyediagram, can you say if the Nyquist criterion will be able to be verified in the transmission channel ? Explain your answer.

2.4 Study with an AWGN propagation channel

We will consider here an unlimited bandwidth propagation channel, adding a white gaussian noise $n(t)$ to the transmitted signal $x(t)$. The noise is generated using *n=σ∗randn(1,length(x))*, where σ_n^2 is given by:

$$\sigma_n^2 = \frac{P_x N_s}{2log_2(M)\frac{E_b}{N_0}},$$

with

- P_x representing the transmitted signal power,
- N_s representing the oversampling factor: $T_s = N_s T_e$, $T_e = 1/F_e$ being the sampling period,
- M being the modulation order,
- $\frac{E_b}{N_0}$ representing the signal to noise ratio per bit at the receiver input.

1. Channel implementation without noise: using the optimal time instants (without ISI) and an optimal threshold for the decisions, verify that the transmission BER is 0.
2. Channel implementation with noise: add the noise and plot the Bit Error Rate as a function of the SNR per bit at the receiver input, E_b/N_0, in decibels. Take values for $(E_b/N_0)_{dB}$ from 0 to 8 dB.
3. Compare, plotted on the same figure, the obtained BER with the transmission channel theoretical one. The objective of this plot is to validate the good functioning of your implemented transmission channel.
4. Change the receiver filter in the implemented channel using $hr = ones(1, N_s/2)$ and plot the obtained Bit Error Rate as a function of the SNR per bit at the receiver input, E_b/N_0, in decibels. Take values for $(E_b/N_0)_{dB}$ from 0 to 8 dB.
5. Compare, plotted on the same figure, the BERs obtained using the two different receiver filters. Explain the similarity or the difference between both (comparison of the two channels in terms of power efficiency). Is there a more power efficient channel ? Explain your answer.