Advanced Wireless Communications: Algorithms and Architectures **Assignment 3.a**
Spring Semester 2013 Out: 2013-03-13
Telecommunications Circuits Laboratory Due: 2012-04-04
EPF Lausanne $Revision : 1.3$

DSSS & CDMA (Part 1)

The current assignment deals with `direct sequence spread spectrum` (DSSS). The basic idea behind DSSS is to spread the energy of the signal over more bandwidth, than would be strictly required. One should note that increasing the bandwidth of a constant-rate transmission is essentially the same thing as reducing the rate of a constant-bandwidth transmission.

Orthogonal Pulses

In narrow-band transmission orthogonal pulses are used to carry the information. Usually the pulse shape is chosen in a way that minimizes the used bandwidth, which means that the "length" of the pulse equals the symbol period T_s which has a bandwidth $B \approx 1/T_s$.

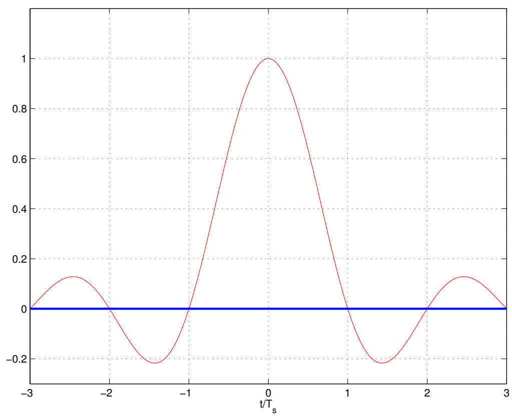
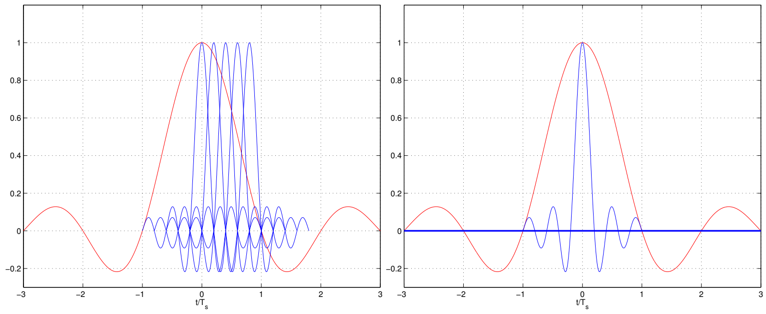


Figure 1: The central part of a sinc pulse of bandwidth 1. The X axis is normalized with T_s.

However one can also use shorter pulses for transmission. We call these pulses chips. A chip of duration $T_c = T_s/G$ is still orthogonal to its T_s-spaced and shifted copies. Its bandwidth B_c is due to the shorter length approximately G times higher $B_c \approx 1/T_c = G/T_s = GB$. And we can fit up to G such short pulses in the symbol time T_s. In Fig. 2 we can see such a sequence of chip rate Nyquist pulses. Looking at the first chip only we can clearly see that there is no inter symbol interference between the chips of successive symbols.

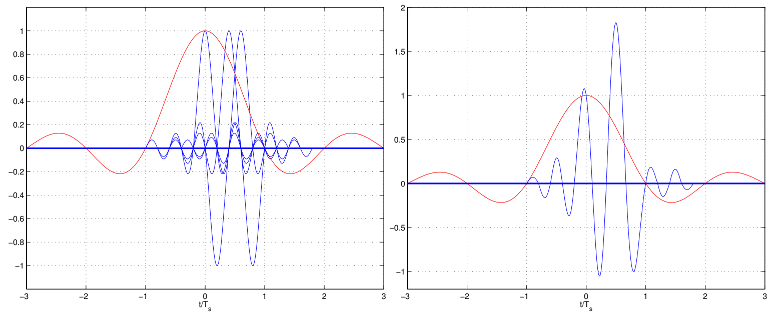


(a) A set of chips. (b) The first chip.

Figure 2: Comparison of a $\frac{T}{1}$ pulse and a set of orthogonal $\frac{T}{5}$ pulses. All zero crossings of the longer pulse are also zero crossings of the shorter pulses.

Spreading Sequence

The pulses in Fig. 2a correspond to a chip sequence $\mathbf{c_1} = [+1, +1, +1, +1, +1]$. This is one possible choice for a spreading sequence. In fact one has G degrees of freedom to choose the spreading sequence. All vectors $\mathbf{c}$ of the G-dimensional vector space that fall within the total energy limitation E_s can be used.



(a) Pulses of Chip (b) Resulting Waveform

Figure 3: The pulse train and the actual waveform for the sequence $[+1, -1, +1, +1, -1]$

Auto Correlation

Another advantage of DSSS is that it allows simple exploitation of the frequency diversity of a frequency selective channel. However, it imposes additional constraints on the spreading sequence. To reduce the amount of interference, the correlation between the sequence and its shifted copy has to be as small as possible.

A set of sequences that features an especially low off-peak auto-correlation are the Barker codes. They are widely used in wireless communications for example in the WLAN standard IEEE 802.11b. Examples for such Barker codes are listed in the following table.

Length	Code						
2	+1	-1					
4	+1	-1	+1	+1			
7	+1	+1	+1	-1	-1	+1	-1

This list is not complete. You can find the remaining Barker codes on the Internet (for example, on Wikipedia).

Framework

For our simulation we use a modified version of the framework of the last exercise. The Fading and the AWGN channel we assume a flat fading **single carrier** environment.

$$y_{i,j} = h_{i,j}c_jx_i + n_{i,j}$$

With i being the index of the symbol and $j = 1, \ldots, G$ the index within the spreading sequence. There is no inter-symbol or self interference. c_j are the coefficients of the spreading sequence. The noise $n_{i,j}$ is a zero mean Gaussian. The fading coefficients $h_{i,j}$ are correlated Gaussian random variables.
The multipath channel model convolutes the spread signal with the impulse response of the channel and add white Gaussian noise afterwards. The taps of the impulse response are Gaussian random variables with a variance of 1. The length of the impulse response can be restricted with the `P.ChannelLength` parameter.
To use DSSS, the framework was extended with a `P.Sequence` parameter. This should be a vector that specifies the coefficients c_j of the pulses. The framework will increase the sampling frequency according to the sequence length. The power of the spreading sequence will be automatically normalized. A spreading sequence $\mathbf{c} = [1]$ will simulate the behavior without spreading.
The transmission is split into frames which are assumed to be transmitted apart in time. For each frame an independent channel realization is generated. Hence the channel coefficients $h_{i,j}$ are correlated within a frame but not between different frames. Be sure to simulate not only a sufficient number of bits but also a sufficient number of frames to get reliable results.
The framework was programmed in a very modular fashion. Try to integrate additional coding and receiver schemes as configurable as possible. Usually, you don't need to make multiple copies of the file.

Tasks

1. The framework now offers the possibility to specify the spreading sequence. The file `paramspread.m` invokes a simulation of a non-fading, non-multipath AWGN channel with a spreading sequence of length 7. Unfortunately the teaching assistants forgot to implement a suitable receiver.

    - Implement a flexible `Correlator` receiver that can be used to receive any given spreading sequence. The inner product of the correlator can be very efficiently programmed with a vector matrix multiplication.
    - Find a second spreading sequence of the same length which achieves the same performance in the AWGN scenario.
    - Question: What is the criterion for such a sequence?

`Hint:` A very handy way to create a multi-simulation comparison plot is a parameter file that invokes the `simulator.m` function multiple times with a different `P.Sequence` parameter.

2. Now you can compare the effect of different spreading lengths.

   - Simulate the AWGN channel with spreading sequences of 4,7,11 and 13. Plot the BER curves against the SNR of all simulations in the same figure.
   - Question: What can you observe?

3. The comparison in Task2 could be considered misleading as our SNR definition is based on the power of the noise
$$\text{SNR} = \frac{P_s}{P_n}$$
Unfortunately the noise power changes with the width of the bandwidth in the AWGN channel. That's why for this case often a different measurement is used
$$\text{EbN0} = \frac{E_b}{N0}$$
where Eb is the used energy per bit and $N0$ is the noise variance. In practice switching the definition leads to a shift of the BER curve. We have provided a function `EbN0shift` which generates from a parameter structure `P` a correctly shifted EbN0 range corresponding to the simulated BER values.

   - Use the `EbN0shift.m` function to extend the simulation of Task 2 and plot the BER curves bases on the EbN0.
   - Question: Can you explain your observation?

4. We consider now a multipath channel model. The impulse response is time invariant during the frame but independent between different frames.

   - Implement a rake receiver for the given multipath case. The receiver should have a configurable number of fingers. (Note: The number of fingers can also be smaller than the number of non-zero channel coefficients)
   - Simulate the parameter file `parammultipath.m` with 1,2 and 3 fingers.
   - Question: Which effects do you observe? Can your explain them?

Hand-In Instructions

You are required to hand in one report and code for assignments 3.a and 3.b (which you will receive next week).