POLITECNICO DI TORINO

Master of Science in Communications Engineering

COMMUNICATION SYSTEMS

REPORT: ASSIGNMENT 3

December 22, 2022

Luca Nepote (s315234)

1 Introduction

In the third assignment of the Master of Science course "Communication Systems" we have dealt with 'Gold Codes" and its properties, CDMA performance with a single and multiple users and the "Hartogs-Hughes" algorithm for OFDM bit allocation.
The assignments' problems have been solved in the Matlab environment.

2 Exercise 1: Properties of Gold Codes

In the first exercise of the assignment we studied the properties of Gold Codes: the entire set is obtained by the $2^m + 1$ sequences

$$c_1, \quad c_2, \quad c_1 + T^i(c_2) \qquad \text{where } 0 \leq i \leq 2^m - 1$$

and where c_1 and c_2 are two m-sequences generated by two LFSR with paired primitive polynomials $p_1(D)$ and $p_2(D)$. Moreover, $T^i(c_2)$ is the i_{th} cyclic shift of the second sequence.
Basic properties of Gold sequences are:

- the sequence length is equal to

$$N = 2^m - 1;$$

- $2^m + 1$ gold sequences inside the Gold code;
- the cross-correlation could achieve three possible values:
  - -1
  - -t-2
  - t

  where $t = 2^{\frac{m+1}{2}} + 1$ for m odd and $t = 2^{\frac{m+2}{2}} + 1$ for m even.

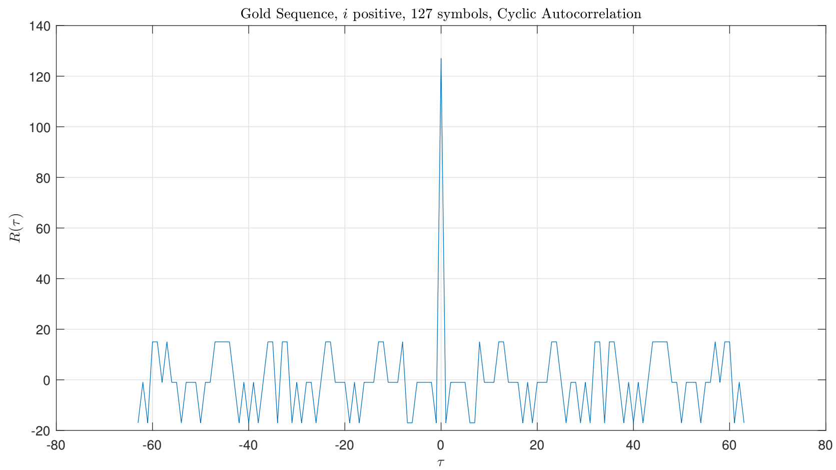
In order to consider paired polynomial, being α the primitive element of GF(2^m) and $p_1(D)$ the minimal polynomial of α, $p_2(D)$ is the minimal polynomial of α^t where t is defined as before.
In the considered case m, the number of cells in the LFSR, is $m = 7$ and the two primitive polynomials are are

$$p_1(D) = x^7 + x^4 + 1$$
$$p_2(D) = x^7 + x + 1.$$

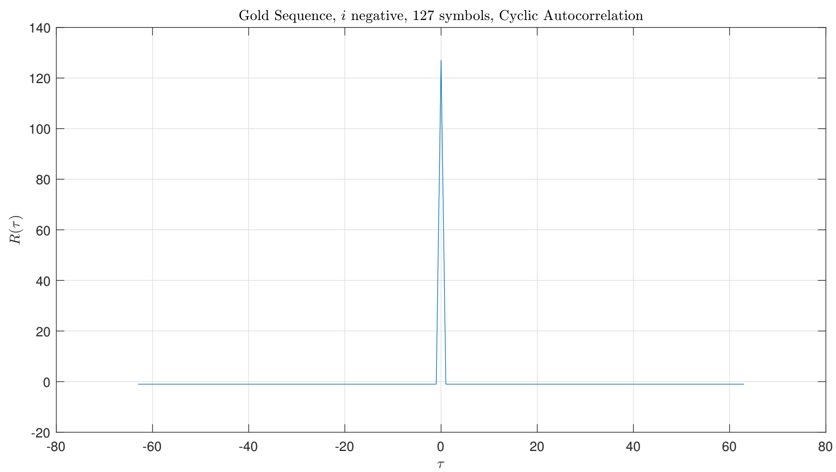
At first we generated a gold sequence considering a positive cyclic shift (index $i \geq 0$), for example equal to $'1'$, and we computed the cyclic autocorrelation, which is defined as

$$r_{c_i'}(\tau) = \sum_{n=1}^{N} c_i'(n) c_i'(n+\tau) \qquad 0 \leq \tau < N$$

where c_i' is the bipolar gold sequence.



Then we repeated for a negative value of i.



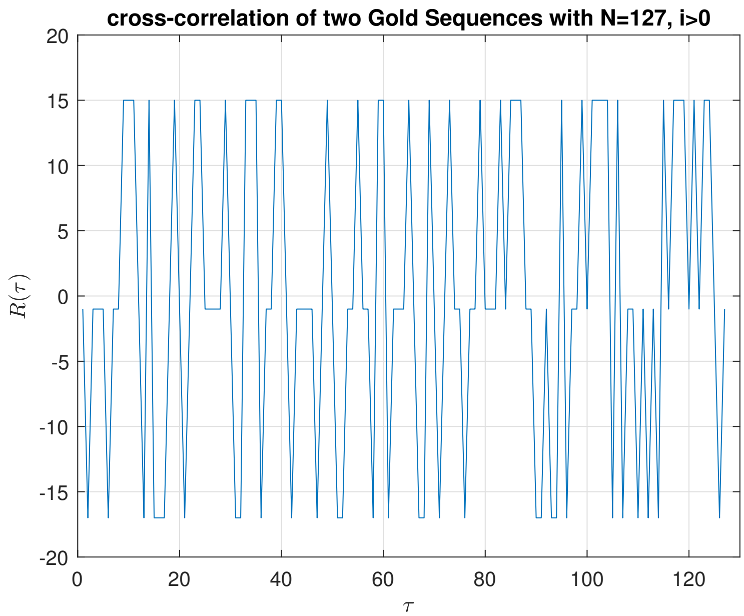
As we can see, the maximum value corresponds to $\tau = 0$. Moreover, in the case with $i > 0$, the values oscillate for $\tau \neq 0$ and they are symmetric with respect to $\tau = 0$, Whereas, for $i < 0$ the value for $\tau \neq 0$ is always $'-1'$, because for negative value of i we obtain simply one of the sequences generated by a single LFSR. Specifically:

- c_1 with $i = -2$
- c_2 with $i = -1$

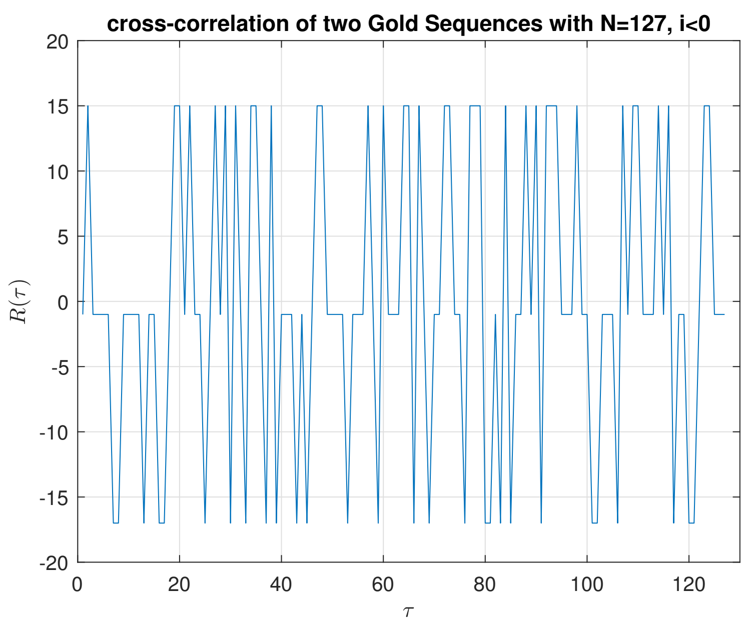
so, the properties are the same of single m-sequence.
In the second part we generated two random Gold sequences, at first both with positive index i, and we computed the cyclic cross-correlation, defined as

$$r_{c'_i c'_j}(\tau) = \sum_{n=1}^{N} c'_i(n) c'_j(n+\tau) \qquad 0 \leq \tau < N$$

The cross-correlation has been reported below.



We then repeated the same procedure considering the second index $i = -2$.



As we can see, because we considered paired m-sequence, using a negative index does not change the behavior of the cross correlation and its properties are

nearly optimal. Moreover, the three reached values are '15', '-17' and '-1', in agreement with the theory.
Successively, we plotted the three versions of the "Welch bound" and the "Sidelnikov bound", considering $N = 127$ and $0 \leq K \leq 200$. They are two lower bounds used to bound the error probability of a code and evaluate its performance. The Welch one refers to linear code, whereas the Sidelnikov one is a generalization for non-linear codes.
For the first bound, considering complex sequences of symbols with unitary magnitude, we have three different expressions:

- $r_M \geq N \sqrt[2s]{\frac{1}{KN-1}\left[\frac{KN}{\binom{N+s-1}{s}} - 1\right]} \qquad \forall s \in \mathbb{Z}^+ = \{1, 2, 3, ...\}$
- $r_M \geq N\sqrt{\frac{K-1}{KN-1}}$
- $\sqrt{N}$ for large K.

Whereas, for binary bipolar sequences, the Sidelnikov bound is defined as

$$r_M \geq \left\lceil \sqrt{\left[(2s+1)(N-s) + \frac{s(s+1)}{2} - \frac{2^s N^{2s+1}}{K(2s)!\binom{N}{s}}\right]} \right\rceil$$

$$\forall s \in \mathbb{N} = \{0, 1, 2, 3, ...\} \qquad 0 \leq s < \frac{2N}{5}$$

that it is tighter than the Welch bound.
We took the maximum value obtained with different values of s and plotted it for each value of K.
In the next step we considered the entire set of $K = 129$ Gold sequences and calculated the value of r_M, which is the maximal cyclic correlation magnitude

$$r_M = max\{r_A, r_C\}$$

where r_A is the maximal out-of-phase cyclic autocorrelation magnitude

$$r_A = max\left\{\left|r_{c_i'}(\tau)\right| \quad 1 \leq \tau \leq N-1 \quad 1 \leq i \leq K\right\}$$

and r_C is the cyclic cross-correlation magnitude

$$r_C = max\left\{\left|r_{c_i' c_j'}(\tau)\right| \quad 1 \leq \tau \leq N-1 \quad 1 \leq i, j \leq K\right\} \quad i \neq j.$$

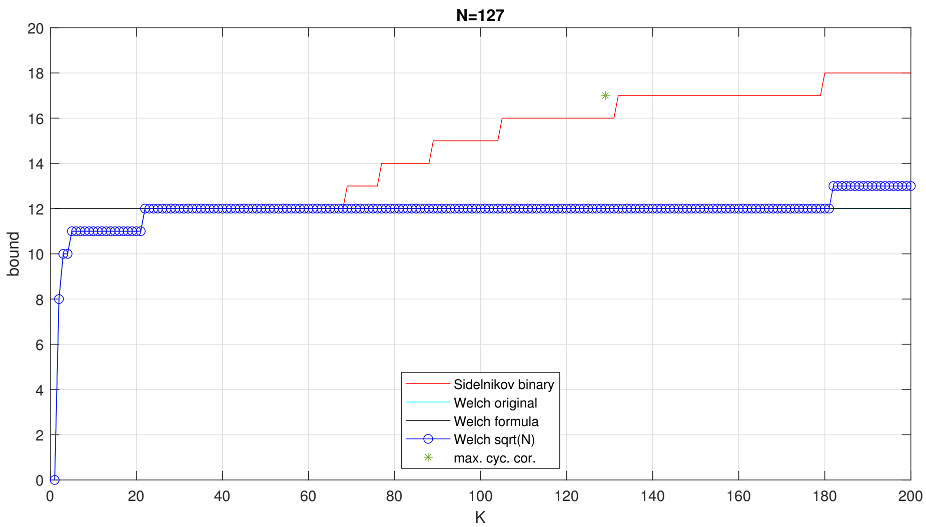
To compute this value, we firstly computed the autocorrelation for each sequence present in the Gold code, dropping the value for $\tau = 0$ and considered the maximum absolute value. For r_C, we iteratively computed the cross correlation between the first sequence and all the following ones, then we passed to the second and so on. Also in this case we took only the maximum absolute value. Finally, we took the maximum between the two.
The value obtained is equal to $r_M = 17$, in agreement with the cross correlation property described above: because we have $m = 7$,

$$t = 2^{\frac{7+1}{2}} - 1 = 15,$$

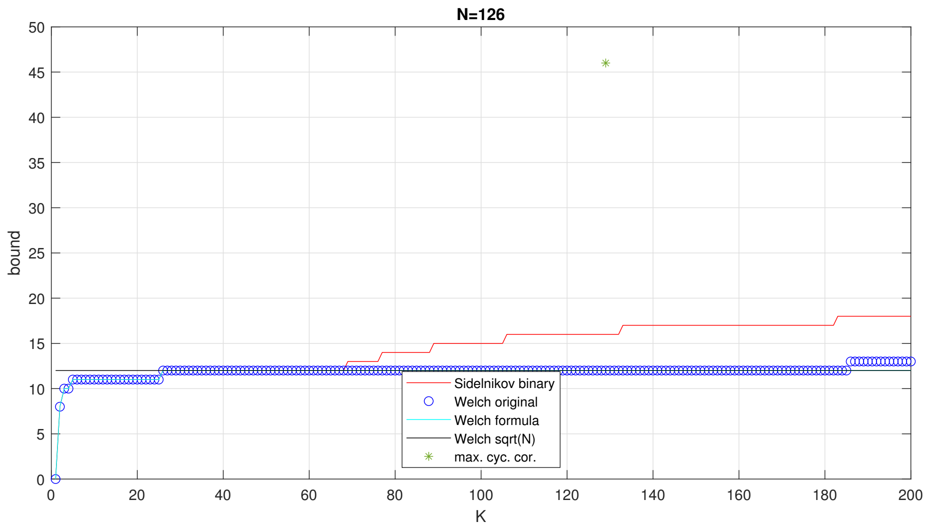
but because we have to take the maximum absolute value, the biggest one is

$$r_M = max\,|-t-2| = |-17| = 17.$$

Moreover, the value stays above the two bounds, so the code has good error-correcting capabilities and it is possibile to correct a large number of errors.



In the last part we reduced the length of the sequences to $N = 126$, repeated the same procedure followed before and computed r_M as before.
The results are represented below: the value of r_M is bigger with respect to the case with $N = 127$, reaching a value equal to $r_M = 46$. This is no surprising because, if the length of a Gold sequence is reduced, r_M might increase because the period of the sequence is shorter, so the shifted versions are more similar to the original one.



3 Exercise 2: Impact of cross-correlation on CDMA performance

In this exercise we analyzed the CDMA performance with a single and multiple users. When multiple users want to communicate, there are different solutions in order to satisfy them at the same time, such as FDMA, TDMA and SDMA. In this case we considered the CDMA: "Code Division Multiple Access". Considering R_b as the information bit-rate and T_b as the bit time, having a constellation with 2^m symbols, the symbol rate and symbol time will be respectively

$$Rs = Rb/m \qquad Ts = 1/Rs = mTb$$

In the case of a 2-PAM, the two symbols correspond to $-\alpha$ and $+\alpha$.
In order to serve two users on the same band at the same time we must use two different spreading sequences, to recover correctly the original information bit transmitted. The sequence could have different N chips, with a Chip Time T_c and Chip Rate $R_c = 1/T_c$. The total symbol time T_s and Symbol Rate will be equal to

$$T_s = N \cdot T_c \qquad R_s = \frac{R_c}{N}$$

We considered two scenarios:

- in the first one, a user U_1 generates a random bit v_1, converts it into the bipolar symbol v_1' and uses an $N = 8$ chips long spreading sequence $\mathbf{c_1'}$ to transmit it. In the channel, the information is affected by noise and we receive a new sequence $\mathbf{r}$ that we project over the spreading sequence $\mathbf{c_1'}$ to recover the information bit transmitted. We then compare it with the original one and if they are different, it means that we have an error.

$$\int r(t)c'_1(t)\,dt \qquad \text{where} \qquad r(t) = v'_1 \cdot c'_1 + \text{noise}$$

- in the second one there is another user U_2 that transmits an information bit v_2 (bipolar version: v'_2) over a spreading sequence $\mathbf{c'_2}$. The total vector $\mathbf{r}$ is then given by the sum of the two signals and the noise. We project over $\mathbf{c'_1}$ and compare the result with v_1.

$$\int r(t)c'_1(t)\,dt \qquad \text{where} \qquad r(t) = v'_1 \cdot c'_1 + v'_2 \cdot c'_2 + \text{noise}$$

FIRST SCENARIO

We generated a random bipolar spreading sequence $\mathbf{c'_1}$ of $N = 8$ chips and simulate the error rate performance of U_1.
Moreover, because we are considering a single user and there is no interference, we wanted to prove that the BER is equal to

$$P(e) = \frac{1}{2} erfc\sqrt{\frac{E_b}{N_0}}$$

The proof is the following one: we start from

$$P(e) = \frac{1}{2} erfc\sqrt{\frac{\alpha^2}{2\sigma^2}}$$

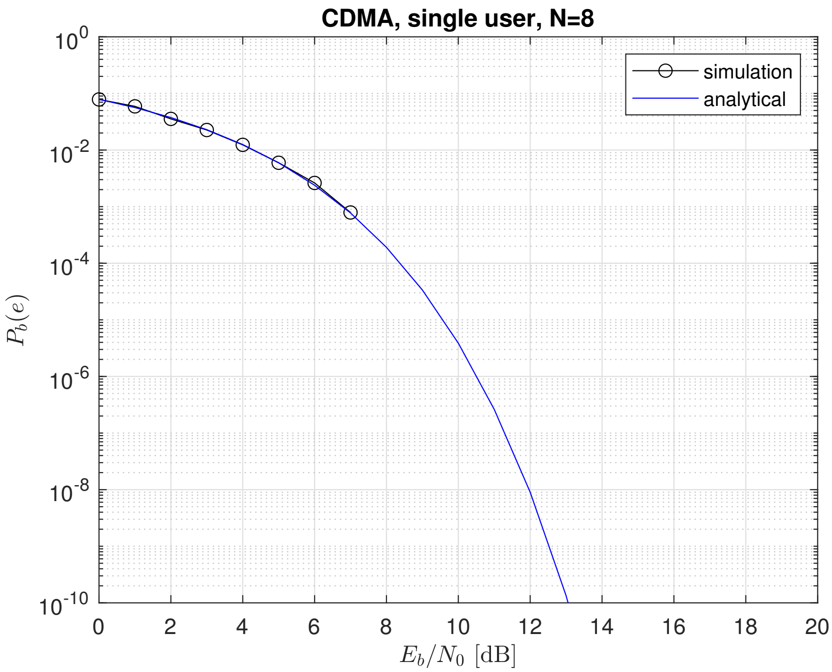
and we know that

$$\sigma^2 = \frac{N \cdot N_0}{2} \qquad E_s = \frac{\alpha^2 + \alpha^2}{2} = \alpha^2 \qquad E_b = \frac{E_s}{N} = \frac{\alpha^2}{N}$$

Substituting we obtain:

$$P(e) = \frac{1}{2} erfc\sqrt{\frac{\alpha^2}{2\sigma^2}} = \frac{1}{2} erfc\sqrt{\frac{N \cdot E_b}{N \cdot N_0}} = \frac{1}{2} erfc\sqrt{\frac{E_b}{N_0}}$$

Notice that at the denominator we have an additional multiplication factor equal to N because we are summing N gaussian variable with the same variance.



As we can see, the simulation results and the analytic expression overlap and the system error rate correspond to a 2-PAM.

SECOND SCENARIO

We generated another random bipolar spreading sequence $\mathbf{c_2'}$ of $N = 8$ chips and simulate the error rate performance of U_1 projecting over $\mathbf{c_1'}$. In this case there is no cyclic shift.
In this case, the dominating parameter p for the two-users performance is the projection of $\mathbf{c_1'}$ over $\mathbf{c_2'}$: if the two sequences were perfectly orthogonal, the projection would be zero, otherwise, there is interference between the two.

$$p = \int c_1' \cdot c_2' \, dt$$

We report the results for different spreading sequences and value of p. Moreover, the analytical curve of the CDMA system with two users and no cyclic shift is obtained as:

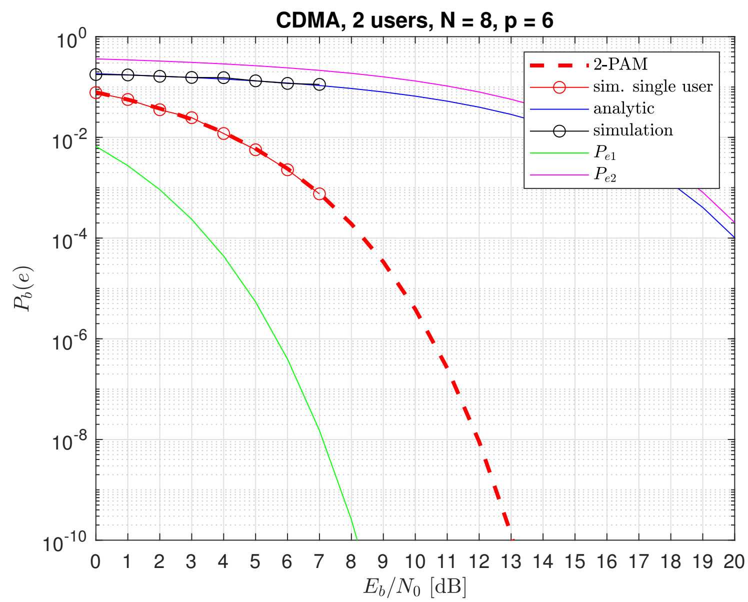
$$\tfrac{P_1(e)+P_2(e)}{2},$$

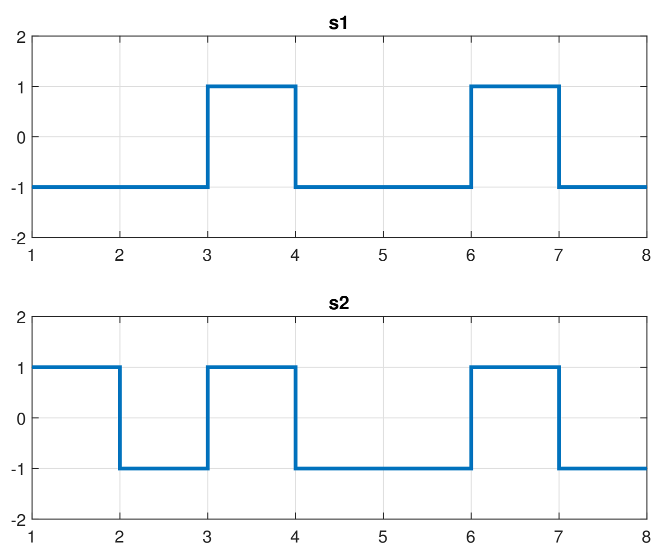
where P_1 and P_2 are

$$P_1(e) = \tfrac{1}{2} erfc\sqrt{\tfrac{\alpha_+^2}{2N\sigma^2}} = \tfrac{1}{2} erfc\sqrt{\tfrac{(N+p)^2}{2N\sigma^2}}$$

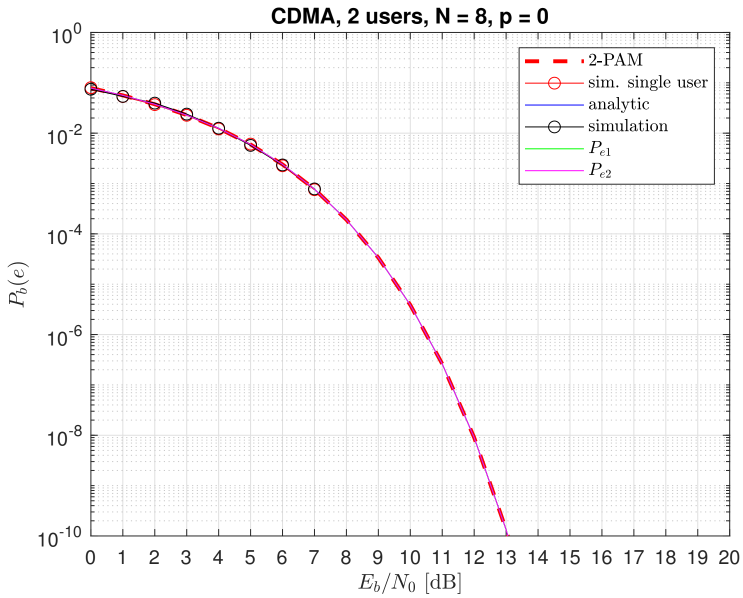
$$P_2(e) = \tfrac{1}{2} erfc\sqrt{\tfrac{\alpha_-^2}{2N\sigma^2}} = \tfrac{1}{2} erfc\sqrt{\tfrac{(N-p)^2}{2N\sigma^2}}$$

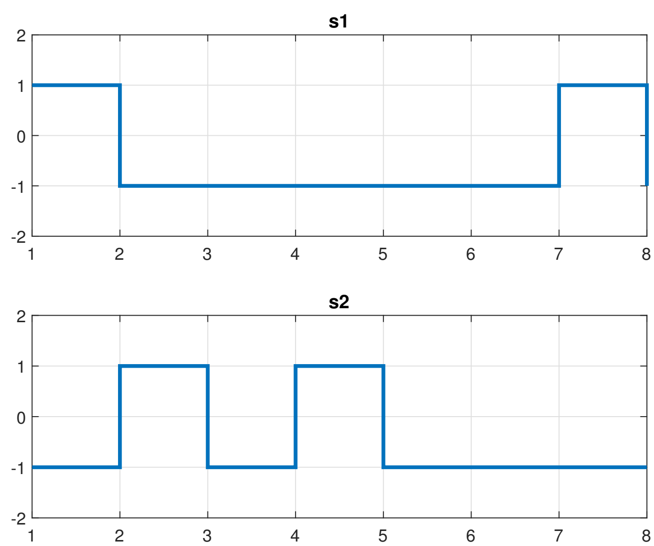
We plotted, for completeness, also the single probabilities $P_1(e)$ and $P_2(e)$: from the results we can see that the case with '-' is the one that characterize most the behavior of the curve. This is not surprising because the error probability is given mostly by the case with minimum distance of the 2-PAM.



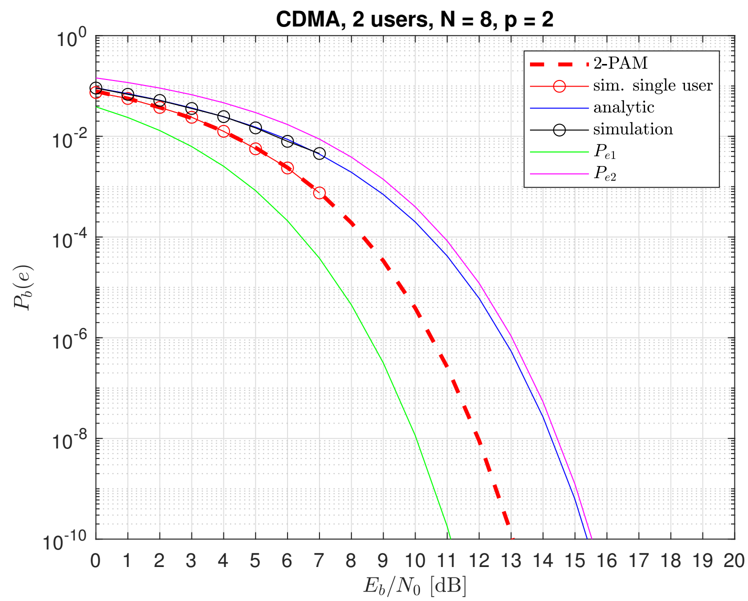


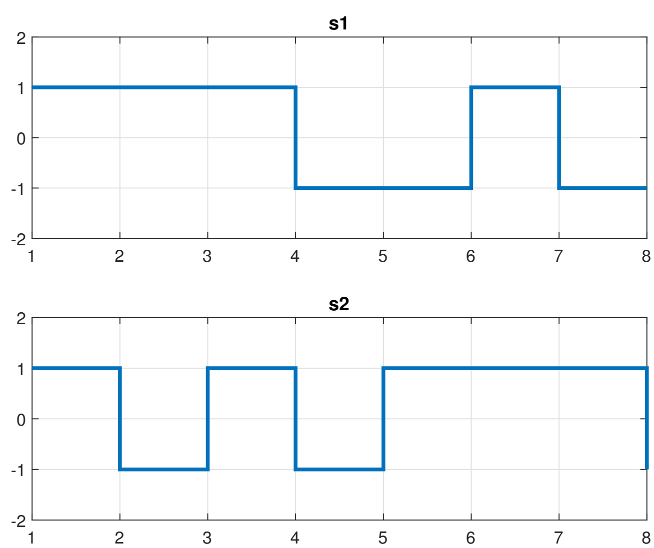
In this first case with $p = 6$ the two spreading sequences are very similar, so the value projection is high and the error probability is bigger due to the interference.



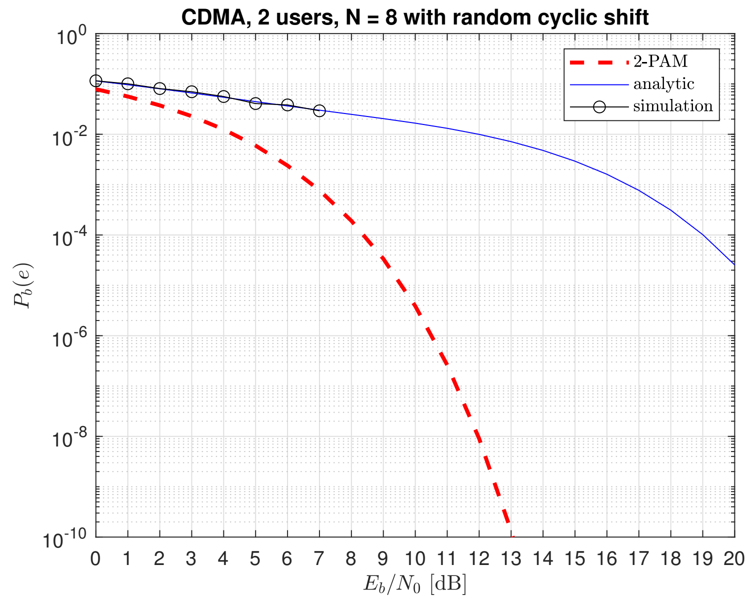


Here $p = 0$, so there is no interference between the two sequences and the curve matches perfectly the one with a single user.





In this case $p = 2$, so there is some interference and the error probability increases, but it is smaller with respect to the first case.
In the last part we considered also a random cyclic shift of the second sequence,



initial case with $p = 6$, plus random shift

as in a real transmission. In this case the dominating parameter p of the performance is the maximum of the cross correlation between the first spreading sequence with respect to any cyclic version of the second. The analytic curve is obtained as the weighted sum:

$$\sum_{i=0}^{7} \frac{1}{8} \frac{P_{1_i}(e) + P_{2_i}(e)}{2},$$

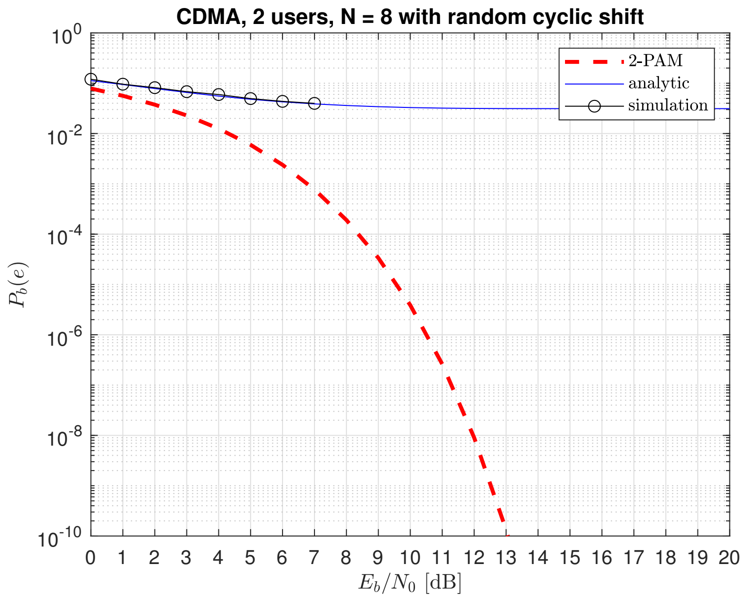
where P_1 and P_2 are

$$P_1(e) = \frac{1}{2} erfc \sqrt{\frac{(N+p)^2}{2N\sigma^2}}$$

$$P_2(e) = \frac{1}{2} erfc \sqrt{\frac{\alpha_-^2}{2N\sigma^2}} = \frac{1}{2} erfc \sqrt{\frac{(N-p)^2}{2N\sigma^2}}$$

computed for each value of p (so with the different cyclic version of $\mathbf{C_2'}$).

We can conclude that, in the case with cyclic shift, the BER increases because it could happen that the two sequences are more similar. Notice also that for the initial case where p was equal to zero, the interference is no longer null.



initial case with $p = 0$, plus random shift

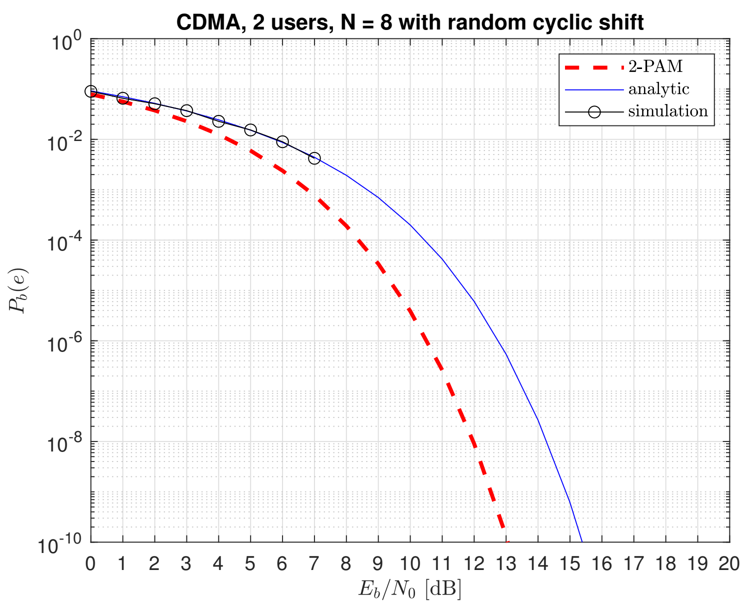


Figure 2: initial case $p = 2$, plus random shift

4 Exercise 3: Hartogs-Hughes algorithm for OFDM bit allocation

In the last exercise we dealt with bit allocation for OFDM: "Orthogonal Frequency Division Modulation". Because we want a flat fading (frequency response flat over the band), the total band is divided into K different sub-bands where in each one the response is flat. We must satisfy the condition:

$$\Delta << B_c = \frac{1}{D}$$

where D is the delay spread given by $D_{max} - D_{min}$.
Moreover, on each sub-band we have a different attenuation A_i, given by

$$A_i = \frac{1}{|H(f=f_i)|^2}$$

where $H(f = f_i)$ is the frequency response corresponding to the f_i frequency inside the band.
Given a total transmitted power P_{TX}, there are different ways to distribute it among the different tones. In particular we could divide equally the power by the N sub-bands, transmit more power to the bad tones (with high attenuation) to help them or to the good tones (low attenuation) to exploit them.
Thanks to the "Water Filling Principle" we can demonstrate that in order to maximize the total bit-rate, and so the total number of transmitted bits

$$m = \sum_{i=1}^{N} m_i,$$

we have to transmit more power to the tones with low attenuation: considering the constraint $\sum_{i=1}^{N} P_i - P_{TX} = 0$ and the link between m_i and P_i

$$m_i \leq \left[log_2(1 + \frac{P_i}{\alpha_i})\right]$$

where α_i is the normalized attenuation for the tone i, we have to maximize

$$m = \sum_{i=1}^{N} m_i = \sum_{i=1}^{N} \left[log_2(1 + \frac{P_i}{\alpha_i})\right].$$

Using the "Lagrange optimization" we arrive to the conclusion that for each tone we choose

$$P_i \text{ as } P_i + \alpha_i = c, \text{ with } c = \frac{P_{TX} + \sum_{i=1}^{N} \alpha_i}{N}$$

so we have to transmit more power to the high tones in order to exploit them.
In the exercise we considered a four rays channel with impulse response

$$h(t) = a_1\delta(t - D_1) + a_2\delta(t - D_2) + a_3\delta(t - D_3) + a_4\delta(t - D_4)$$

where $D_1 = 1, D_2 = 1.01, D_3 = 1.015$ and $D_4 = 1.02$, whereas each coefficient has a random phase uniformly distributed between 0 and 2π with amplitudes $|a_1| = 1, |a_2| = 0.5, |a_3| = 0.9$ and $|a_4| = 0.3$.
The frequency response is then

$$H(f) = a_1 e^{j\phi_1 - 2\pi j f D_1} + a_2 e^{j\phi_2 - 2\pi j f D_2} + a_3 e^{j\phi_3 - 2\pi j f D_3} + a_4 e^{j\phi_4 - 2\pi j f D_4}$$

Each frequency is given by

$$f_i = i\Delta$$

and $1 \leq K \leq 200$.
We initially plotted the attenuation in $[dB]$ for each band. We then applied the "Hartogs-Hughes" algorithm: we look for the tone on the total band with minimum attenuation, we "buy" it and we transmit a bit on that tone. In order to transmit another bit on the same tone we have to spend the double of the initial attenuation value. We repeat the procedure until we can buy a bit in the overall band, then we stop.
Successively, we considered only the active tones obtained with the HH algorithm and we divided equally the total power among them. Then we plotted the number of bits transmitted for each tone according to this power division.
Finally, in the last case we distributed P_{TX} on all the N tones and we repeated the previous procedure. All the results are reported in the same picture to compare better the three cases.
Notice that we repeated the procedure multiple times because the attenuation depends on the frequency response, that changes every time due to the random phase given to the coefficients.

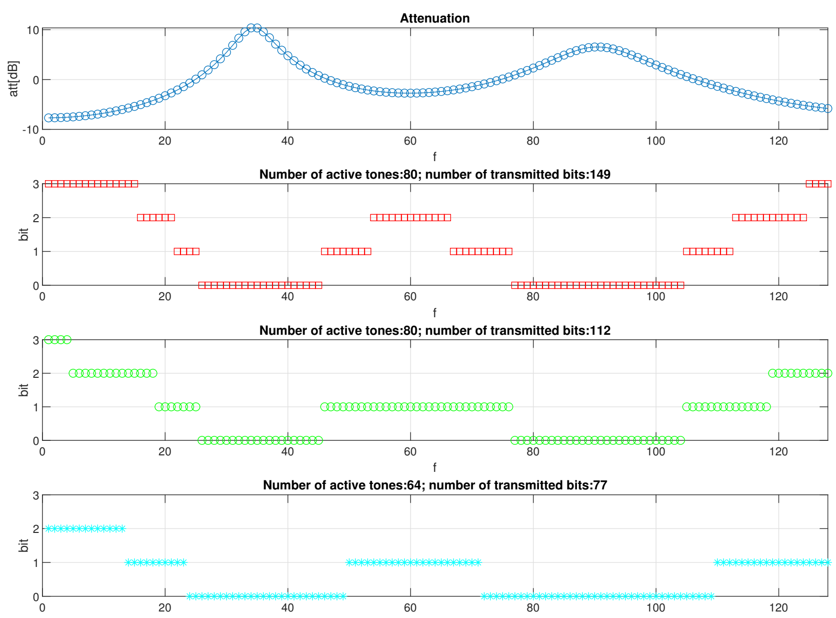


Figure 3: First Case

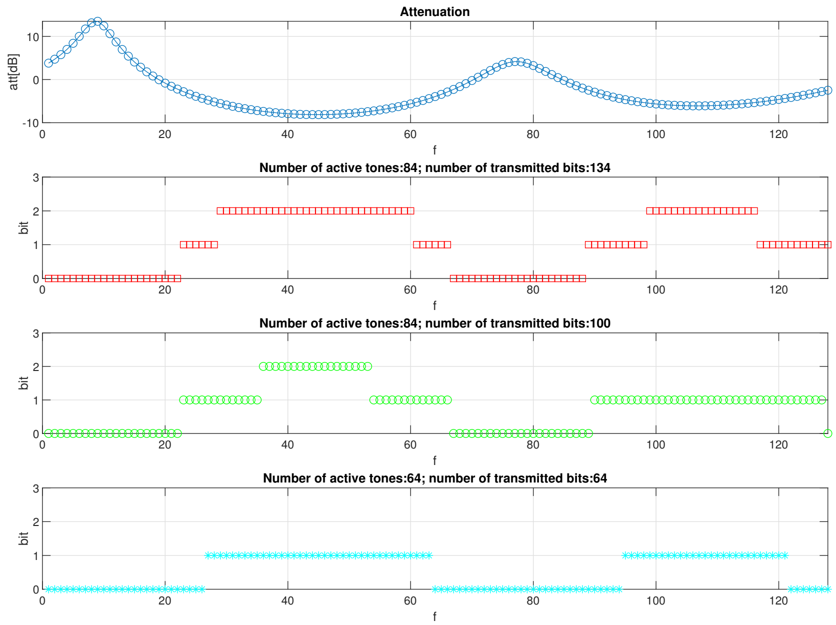


Figure 4: Second Case

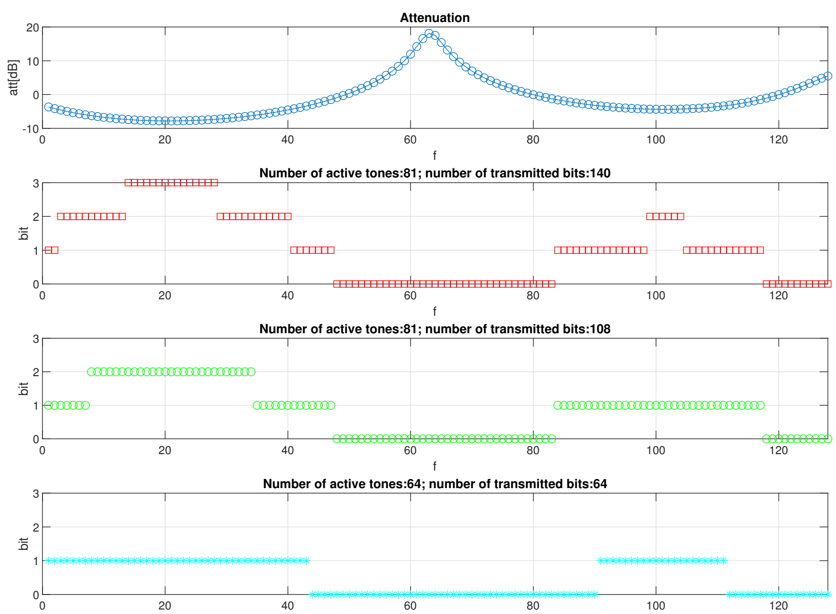


Figure 5: Third Case

As we expected, the biggest number of transmitted bits corresponds to the case of “Water Filling Principle”, because the power is transmitted more usefully. Moreover, we can easily see that the tones with more bits are the ones with smaller attenuation, whereas the high ones are inactive.
In the second scenario, where we divided equally the power on the active tones, the number of bits is smaller with respect to the previous one and the number of sub-carriers active could decrease.
Finally, dividing uniformly the power among the N tones is not very useful: both the numbers of active tones and the of the bits are smaller, so we do not exploit all the available power.
In conclusion, the “Hartogs-Hughes” algorithm is the one which exploits better the total power, because also the remaining power, once the algorithm is finished, could be used or stored for other applications.