WIRELESS TRANSMISSION LAB

Antonio Simón Martín - 75576262X
Claudia Tomasicchio
Javier Lobato Martín - 44739277E

(Group XXXXX)

Politecnico di Torino - DSP Module

Professor: Guido Montorsi

Sede Centrale - Cittadella Politecnica: LED2-Lab

INTRODUCTION

In this report we are going to show the work we have done regarding the wireless transmission lab. We will be mostly using 'TOPCOM++', a library of C++ classes for simulating the physical layer of a telecommunication system. This software allows us to implement a communication system with several configurations, so we can proceed from the simplest one, to one that simulates an actual complete communication system. Along this report we will indicate the results that we obtained in each of the sections, discussing the results and understanding them based on our theoretical knowledge of each one of the aspects. Tables and graphs will be added when needed.

1. Simulation of a basic digital transmission system

In this first task we are going to implement the simplest digital communication system, without TX and RX filters. We will take some measurements on it such as comparing BER versus SNR with different types of modulation (QAM and PSK). First we are only going to modify the modulation cardinalities (4 QAM and 16 QAM) and types, then we will focus on the system with filters. We start with the Quadrature Amplitude Modulation.

OJO EN ESTA PARTE NO OLVIDAR HACER TODAS LAS PARTES LAS DE 4 QAM Y 16 QAM Y LAS DE PSK

1.1 System without TX and RX filters.

BER versus SNR for different modulation cardinalities (types).

Figure 1. *Simplified model without filtering.*

At first we observe a simplified digital transmission, in transmission the "shaping filter" and in reception "matched filter" are not present in this case. The system filter satisfies the Nyquist criterion therefore it was possible to build this simple model. The main blocks are:

- **PN_Source:** The source generates a sequence of bits and these save in a buffer called "data".
- **Modulator:** It maps the sequence of bits in constellation points and saved in "mod".

- **AWGN_Channel:** Add the Gaussian noise, the constellation points affected by noise saved in "modRX".
- **Demodulator:** The demodulator tries to find the minimun euclidean distance between the constellation point and the received point.
- **BER_meter:** Compares the received and transmitted points and calculate the error between them.

With the TOPCOM C++ we provide the parameter using an input file in which we modify the values.

- ***AQUI PONE LAS COSAS QUE CAMBIA EN EL INPUT FILE CREO PERO NO SE SI COPIAR ESO TAMBIEN.***

Figure 2. *Input file.*

Once the parameters for the modulation have been selected, we carry out the simulation for a 4 QAM modulation. In the following figure we can observe the results of the simulation in which Eb/N0 increases in each scenario and therefore the BER is varying, being this less and less. The simulation stopped when the BER is of the order of magnitude **10^6** because it is an acceptable value. With increasing Eb/N0 increases the number of transmitted bits and thus also dramatically increases simulation time.

Figure 3. *4QAM modulation. (Pantalla negra con los resultados de la simulacion)*

Figure 4. *Output file 4QAM.*

In Figure 4 we can observe the output file of the simulation 4QAM modulation, in which the first three columns are the input values of each case, name of the output file, the relation Eb/N0 and the number of bits for the modulation. The following columns show the number of errors in each case, the number of transmitted bits and the BER respectively.

Now we are going to observe other cases of modulation in which we will use 4, 6 and 8 in the number of modulated bits, thus carrying out 8QAM, 64QAM and 256QAM modulations. As we can see in the following figures, the number of transmitted bits as well as the Eb / N0 relation vary with respect to the previous case. In particular, we have to note that to obtain an acceptable BER of the order of **10 ^ 6** we will need an Eb / N0 of **14, 18 and 23** dBs respectively for each of these modulations. In this way, to have an acceptable BER value, we obtain a higher Eb / N0 value.

Figure 5. *Output file 16QAM.*

Figure 6. *Output file 64QAM.*

Figure 7. *Output file 256QAM.*

Finally we are going to observe a graph showing the Eb / N0 relationship on the x axis and the BER on the y axis for each of the modulations. To better observe the differences between one modulation and another we use the logarithmic scale. As we can see, the 4QAM modulation curve reaches an acceptable BER value much earlier than the 256QAM modulation curve, this being the type of modulation that takes the longest to reach an acceptable value.

Figure 8. *BER vs Eb/N0 for each modulation type.*

1.2 System with filters

Next we are going to analyze the system with filters and comment on the blocks that we add, as well as obtain the Scattering diagram, IQ plots and spectra for different roll-offs of SRRC filters. **We will check that the behavior of the BER does not change when you insert the filters.**

Figure 9. *Model with filters.*

The new blocks are:

- **Upsampler: is inserted between each sample that comes out of the modulator ns-1 zero.**
- **TX Filter:** shaping filter, SRRC filter.
- **RX Filter:** matched filter, SRRC filter.
- **Downsampler: to have again one sample for symbol.**

In the Input file we add the following new input parameters, **modulation cardinality,** number of samples per symbol and length of filter and roll-off. And we measure the Eye Diagram, IQ plots, scattering diagram and spectra and BER measures.

In terms of performance there is no difference with the case of the system without filters that we have implemented previously, because the new blocks that we have introduced are ideal. However, we truncate and delay the impulse response of the filters in order to obtain FIR filters, this is why we introduce a large value for the filter length and thus ensure that there is no degradation. In addition, we are not interested in implementing a very fast filter, so we put 20 for the filter length, 0.5 for the roll-off and number of samples per symbol 4, we do not need a high number since this entails a high number of samples to simulate.

En terminos de rendimiento no hay diferencia con el caso del distema sin filtros que hemos implementado anteriormente, debido a que los nuevos bloques que hemos introducido son *ideales. Sin embargo, truncamos y retrasamos la respuesta al impulso de los filtros para asi obtener filtros FIR, es por esto que introducimos un valor grande de longitud del filtro y asi nos aseguramos que no hay degradacion. Ademas no nos interesa implementar un filtro muy rapido, de forma que ponemos 20 para la longitud del filtro, 0.5 para el roll-off y numero de muestras por simbolo 4, no necesitamos un numero alto ya que esto conlleva un numero alto de muestras a simular.*

Now we have more columns because wue have more input parameters, the first three columns are the same, the next is the roll-off of the constellation, next the length of the filter, next is the number of samples per symbol and the final three columns are the same than the previous case.

Figure 10. *Output file 4QAM.*

We have represented the same graph as in the case of the system without filters, seeing the Eb / N0 ratio on the x axis and the BER on the y axis. We observe that we find practically the same behavior in both simulations for each type of modulation. This is because we have chosen a high value for the length of the filters to approximate the ideal case of the impulse response and thus not find important differences between the system with filters and without filters.

Hemos representado la misma grafica que en el caso del sistema sin filtros, viendo en el eje x el ratio Eb/N0 y en el eje y el BER. Observamos que encontramos practicamente el mismo comportamiento en ambas simulaciones para cada tipo de modulacion. Esto se debe a que hemos elegido un valor alto e la longitud de los filtros para aproximarnos al caso ideal de la respuesta al impulso y de esta forma no encontrar diferencias importantes entre el sistema con filtros y sin filtros.

Next we are going to carry out a simulation for different values of the filter roll-off, and showing the behavior of the system in the following graph. We set the 4QAM modulation and a fixed filter length at 20, in the graph we see different representations for a roll-off sweep between 0.1 and 0.9. We can see that we did not find any significant difference between any of the represented curves.

A continuacion vamos a realizar una simulacion para distintos valores del roll-off del filtro, y mostrando el comportamiento del sistema en la siguiente grafica. Establecemos la modulacion 4QAM y una longitud del filtro fija en 20, en el grafico observamos diferentes representaciones para un barrido del roll-off entre 0.1 y 0.9. Podemos observar que no encontramos ninguna diferencia significativa entre ninguna de las curvas representadas.

Figure 12. *BER in function of Eb/N0 varying the roll-off.*

Next we are going to re-represent the BER as a function of Eb / N0 but now keeping the filter length fixed at 1 and going through the different roll-off values between 0.1 and 0.9. In this case we can see that differences between the different types of modulation are beginning to be observed, this is due to the low length of the filter order, causing it to move away from the ideal case.

A continuación vamos a volver a representar el BER en función de Eb/N0 pero ahora manteniendo fija la longitud del filtro en 1 y recorriendo los diferentes valores de roll-off entre 0.1 y 0.9. En este caso podemos ver que se empiezan a observar diferencias entre los distintos tipos de modulación, esto es debido a la baja longitud del orden del filtro haciendo que este se aleje del caso ideal

Figure 13. *BER in function of Eb/N0 varying the roll-off, with length of 1*

Now we are going to evaluate the eye diagram for various roll-off values. We are going to check the eye diagram for three roll-off values 0.1, 0.5 and 0.8 and we will observe how the aperture increases or decreases as well as the ISI (Intersymbolic Interference) in each case.
We can see how for the case of higher roll-off (0.8), the traces of the diagram follow a similar path, thus reducing the intersymbolic interference and increasing the opening of the diagram, thus making it easier to detect each symbol of the constellation.

We observe the optimum sampling moment where we can see that we have two values for I (t), these are the real coordinates of the points of the constellation 4 QAM. As we move away from the optimum sampling moment, we obtain a value with some intersymbolic interference.

Ahora vamos a evaluar el diagrama de ojo para varios valores de roll-off. Vamos a comprobar el diagrama de ojo para tres valores de roll-off 0.1, 0.5 y 0.8 y observaremos como la apertura aumenta o disminuye asi como la ISI (Interferencia iIntersimbolica) en cada caso.
Podemos observar como para el caso de mayor roll-off (0.8), las trazas del diagrama siguen una ruta similar, reduciendo asi la interferencia intersimbolica y aumentando la apertura del diagrama, siendo asi mas sencillo detectar cada simbolo de la constelacion.

Observamos el instante optimo de muestreo donde podemos ver que tenemos dos valores para I(t), estos son las coordenadas reales de los puntos de la constelacion 4 QAM. Al alejarnos del instante optimo de muestreo obtenemos un valor con cierta interferencia intersimbolica.

Figure 14. *Eye diagram for 4 QAM with differents roll-off.*

Now we are going to represent the same case with 4QAM modulation and the same three roll-off values with the IQ diagram. Observing the following figure we reach the same conclusion as with the eye diagram, where the case is with the highest roll-off value. 0.8 is where it is best observed and we can differentiate the symbols clearly. With a very low roll-off 0.1, the traces follow very different paths to go from one point to another and a very messy graph will be observed.

Ahora vamos a representar el mismo caso con modulacion 4QAM y los mismos tres valores de roll-off con el grafico de IQ. Observando la siguiente figura llegamos a la misma conclusion que con el diagrama de ojo, donde el caso es con el mayor valor de roll-off 0.8 es donde se observa mejor y podemos diferenciar los simbolos claramente. COn un roll-off muy bajo 0.1, las trazas siguen caminos muy diferentes para ir de un punto a otro y se observar una grafica muy desordenada.

Figure 14. *IQ diagram for 4 QAM with differents roll-off.*

We now represent the scattering diagram for the different roll-off values. This time we do not observe any difference between the different cases, we see a cloud of points around the constellation points, this is due to the truncation of the impulse response.

Representamos ahora el diagrama de scattering para los distintos valores de roll-off. Esta vez no observamos ninguna diferencia entre los distintos casos, vemos una nuve depuntos alrededor de los puntos de la constelacion, esto es debido al truncamiento de la respuesta al impulso

Figure 16. *Scattering diagram for 4 QAM with differents roll-off.*

Finally we are going to represent the frequency spectrum by varying the roll-off again. In this case, if we observe a change in the variation of the roll-off, seeing that for a small roll-off 0.1 we have an almost rectangular filter and that as we increase the roll-off, the occupied bandwidth also increases.

Finalmente vamos a representar el espectro de frecuencias variando nuevamente el roll-off. En este caso si observamos cambio en la variacion del roll-off, viendo que para un roll-off pequeño 0.1 tenemos un filtro casi rectangular y que conforme aumentamos el roll-off aumenta tambien el ancho de banda ocupado.

Figure 17. *Spectrum for 4 QAM with differents roll-off.*

Wireless transmission lab

AUDIO FILTERING LAB

Our main goal for this lab is to implement and test FIR and IIR audio filtering. We will use different structures and data representation types to do the filtering. We will put to test the different DSP techniques that we've learned in the theory classes.

Task 1: Filtering Structures.

In this task we are going to implement and check FIR and IIR filters with floating-point representation. In the project **audio_filter** we have the implementations of an FIR filter using a linear buffer and the code implementation of IIR biquad direct form I. The first of the tasks is to implement a floating-point FIR filter using a circular buffer instead of a linear one. As we know from the theory classes, the difference between them should be zero, as they are two different forms of implementing the same kind of filter (in linear buffer we shift the pointer instead of the actual data, but the rest should be the exact same). To check if our assumption is right, we just substract one output to another, if the result is zero, they are the same. We create the code using the snippets and we find out that, indeed, their output is exactly the same, so we can conclude that they are completely equivalent.

Next, we check the CPU clock cycles (visualizing the parameter called 'Excl Count Average') for both versions and different optimization levels. We can change the optimization level from ‘disabled’ all the way up to 3. For this purpose, we will reflect our results in the following table:

OPT LEVEL	LINEAR BUFFER	CIRCULAR BUFFER
DISABLED	6429912	6025652.22
0	6220514.25	5946822.47
1	6174767.44	5905178.56
2	5739912.47	57003149.66
2	5739912.47	57003149.66

Table 1. Average clock count for FIR filter (linear and circular buffer).

As we can see, we have a general improvement of CPU clock cycles for both versions of the FIR filter each time we increase the optimization level. We can also check that the circular buffer always needs less clock cycles than the linear version, so we can conclude that circular buffer is the most efficient way of implementing this kind of FIR filter.

Next, we have to implement IIR biquad direct form II in floating point, using a single state buffer. For this task, we will also use the code snippets for implementing the filter.

Task 2: fixed-point implementation.

In this task we implement the change between

Task 3: real-time filtering.