

# TEACHING DIGITAL COMMUNICATIONS: A DSP APPROACH

*Jesús Ibáñez, Carlos Pantaleón, Luis Vielva, Ignacio Santamaría*

Dpto. Ing. Comunicaciones, ETSII y Telecom., University of Cantabria  
Avda. Los Castros, 39005 Santander, Spain  
e-mail: [jesus@gtas.dicom.unican.es](mailto:jesus@gtas.dicom.unican.es)

## ABSTRACT

Modern Digital Communications are DSP based. In this paper we present a DSP-based Digital Communications lab. Although based on very simple hardware, we try to show that most practical situations that we can find in the design of digital communication systems can be reproduced in our setting. The main idea in our approach is to give the same importance to discrete-time and continuous-time communication signals. Looking for student motivation the lab is based on a problem solving approach making the students develop from simple to more complex communications systems from the very beginning. It needs very simple equipment already known to the students, so most of the time is devoted to Digital Communications work, and it is low cost. Personal and team work is combined in the lab, pursuing student motivation.

## 1. INTRODUCTION

A digital communications course together with a laboratory is compulsory in most EE undergraduate curricula, following usually a Signals and Systems course that introduces the basics of analog and digital signal processing together with an introductory signal processing lab. Implementing this digital communications lab is not easy. Normally simulation-based labs are the chosen approach, although it has several drawbacks: on the one hand, the theoretical course is normally developed in continuous time with little references to DSP, so that the student has to make a great effort to understand the intricate concepts related to the simulation of digital communication systems; on the other hand, the student motivation is not as high as it could be with a closer relation to more practical settings. In other cases the lab is based on some hardware modules that the student has to interconnect to build real systems and analyze them with the usual equipment. From our experience this type of approaches normally imply several facts: the student devotes much time to learn the particular hardware configuration and to use the equipment, it tends to favour students trying to make things work without really trying to understand what is going on and finally it does not anymore reproduce real communication systems that nowadays are mostly DSP based. Last (but not least) the cost of the lab and its maintenance are a serious drawback when financial and human resources are limited.

Our conclusion of the above discussion is that an ideal digital communications lab should cover both implementation in discrete time and analysis in continuous time. At the University of Cantabria this has been the approach followed for several years.

---

This work has been partially supported by the Spanish Ministry of Science and Technology under project TIC2001-0751-C04-03

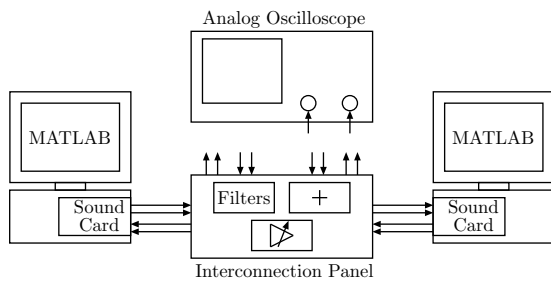
Half of the lab time was devoted to the simulation of digital communications in MATLAB and the other half to signal analysis in continuous time using some hardware modules. The conclusions that follow the guidelines said before are: from one side, the simulation time was too short to cover both the digital communications concepts and the simulation of digital communications; on the other side, the students used much time understanding the hardware and making things work and little understanding the theoretical facts.

After considering several approaches we came with a very simple setting that is shown in Figure 1. The DSP part runs on the PC, the conversion to the analog world is done with a sound card and the channel is a simple board with a filter that can be easily configured to any desired analog response. The paper is structured as follows: in Section 2 the teaching model is described. Section 3 details the hardware and software used in this laboratory together with its advantages and limitations. Section 4 details some experiments that can be implemented with the proposed setting. Conclusions and possible extensions of this lab close this paper.

## 2. A MODEL FOR TEACHING DIGITAL COMMUNICATIONS

The fact that digital communications systems are mainly implemented in discrete time is not covered in usual texts [1, 2]. In our opinion, this limitation affects seriously the development of laboratories based on software like MATLAB, where a precise understanding of the discrete-time signal processing is mandatory.

The model we follow at the University of Cantabria is to parallelly develop the methods for designing both continuous-time and discrete-time digital communications systems. As most digital communication systems are composed of discrete and continuous-time modules, this approach allows us to draw similarities between both approaches and, at the same time, clarify the facts that rules the implementation of different digital communications subsystems in discrete or in continuous time. Basically, it follows a philosophy similar to Oppenheim et. al. [3] when exposing signals and systems. With this approach, it is easy to relate discrete-time and continuous-time communication signals and, for example, make the student understand that we can obtain a continuous-time signal by interpolating a discrete-time signal. Analogously, by applying the sampling theorem, we can obtain a discrete-time communications signal that completely represents a given continuous one. Finally, most of the subsystems in digital communications can be theoretically implemented both in discrete and in continuous time, and we can discuss the reasons beneath this decision. This approach can be adapted directly to a digital communications lab, where the students can generate any discrete-time commu-



**Fig. 1.** Laboratory bench with two computers equipped with sound cards, one interconnecting panel, and one analog oscilloscope.

communications signal by using MATLAB, and convert them into their continuous-time equivalent by using a sound card.

Another pedagogically interesting characteristic of this laboratory is that it combines personal with team work in a quite effective way. The design of the communication systems is mostly individual, but at the end, the students have to transmit and receive an analog communication signal between them. As both students have to work together to achieve a functional communication system, team work is compulsory. It is even possible to establish some type of competition among students to produce the most effective communication system for a given channel, bandwidth and signal-to-noise ratio.

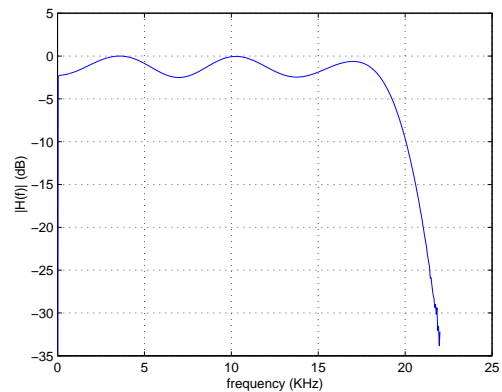
### 3. HARDWARE AND SOFTWARE DESCRIPTION

The proposed laboratory bench is presented in Figure 1. It is composed of two PC computers running MATLAB and hosting compatible sound cards, one interconnection panel and one analog oscilloscope. Every two students share one of this workbench, encouraging team work.

The MATLAB software, with its Signal Processing and Communications Toolboxes, is the core of the laboratory, and allows to create and to process digital communications signals, and to implement and simulate digital communications systems. The recent developed Data Acquisition Toolbox give us control over several data acquisition cards. Every computer of the proposed Digital Communications Laboratory uses a conventional and low-cost sound card (Creative Labs SoundBlaster PCI 128). These cards have at least two analog outputs, that are used as generators for the transmitted signals, and two analog inputs, that capture the incoming signals. Its 16 bit Analog to Digital and Digital to Analog converters have configurable sample rate conversion. They include antialiasing and reconstruction (smoothing) filters with a highly linear phase response, and a passband amplitude response.

The interconnection panel possesses several input and output RCA terminals, attenuators, signal combiners (operational amplifier based adders) and pluggable analog filter modules. This panel allows to distort the communication signals by means of analog filters simulating the channel, to add noise or jamming signals, to attenuate the signal level, and to connect or combine any of the sound card output signals of any of the laboratory workbenches to the sound card inputs.

Finally, there is one analog oscilloscope in which the students can observe the interconnection panel signals (any of the transmitted, received, attenuated, distorted or combined signals), the eye



**Fig. 2.** Sound card frequency response amplitude ( $f_s = 44100$  samples/s.)

diagram, and the constellation diagrams. There is no special requirements regarding the oscilloscope specifications, and the simplest low profile analog oscilloscope (i.e. 20 MHz oscilloscope) is able to properly show the different signals and diagrams.

#### 3.1. Sound card characteristics

It is possible to use many sound cards in this setting, obviously compatible with MATLAB's Data Acquisition Toolbox. In our case we selected the SoundBlaster PCI128 [4]. The frequency response of the analog parts of both cards (the transmitter and the receiver) due to the D/A conversion, the reconstruction filter, the amplifiers, the antialiasing filter of the receiver, and the A/D conversion is presented in Figure 2. It shows a bandpass behaviour (20 Hz – 19 KHz) and a passband ripple of 2.5 dB. The phase response is almost linear. The band ripple is not a problem when analog audio signals are transmitted, but it may affect considerably communication signals. This situation may be considered disadvantageous, but, at the same time, offers a lot of educational possibilities. An inverse filter has been designed and splitted in two matched components at the transmitter and at the receiver. The ripple of the equalized frequency response is 0.05 dB (40 Hz and 18 KHz). In the first sessions of the lab, this filtering is done in a transparent way. But at the end of the lab this equalization problem is highlighted, and the students have to solve it.

Technical characteristics of the SPCI128 board are: 16 bit resolution (effectively 14 bits, that means a signal-to-noise ratio of 70 dB), maximum sampling frequency 48 KHz, crosstalk  $-80$  dB, input and output voltage range  $\pm 1$  V. These characteristics allow us to produce high-quality communication signals.

#### 3.2. MATLAB sound card functions

A set of functions to facilitate to the student the management of the sound card has been developed on the basis of the MATLAB Data Acquisition Toolbox [5, 6]. Namely, the function `init_card()` initialises the card and sets the parameters for the proper behaviour of the other functions. The function `sampling_freq()` sets the sampling frequency of the analog-to-digital (ADC) and digital-to-analog (DAC) converters to a value from the set of admissible sampling frequencies that dictate the combination of operating sys-

tem, Data Acquisition version and sound card. In our case (Windows NT4.0 sp6, Data Acquisition Toolbox r2.0, and SoundBlaster PCI128), sampling frequencies of 8000, 11025, 22050, and 44100 samples per second are allowed. The function `send()` is used to send to the DAC one or two arrays (left and right channels), to specify how many times they have to be sent, and to pre-distort the signals in order to compensate for the reconstruction-filter distortion. The function `close_card()` liberates the resources allocated by the transmission computer after execution of function `send()`. The function `receive()` directs the ADC to receive one or two signals of length `Nsamples`. The function is able to filtrate the signals to compensate the distortion of the anti-aliasing filter. Finally it is interesting to note the gain control that the students have over the signals that are transmitted or received, using the sound control of the PC.

#### 4. LABORATORY EXAMPLES

##### 4.1. From analog to digital communications

The communications system defined previously may be used, in first place, to illustrate an analog communication system. Using speech or audio, the signals may be transmitted in baseband or AM modulated. If the receiver adds noise, the effect of this noise may be perceived when the signal is received. The student may try to eliminate this noise, but the conscience of impossibility soon arises. We then may justify the advantage of using digital communications, due to the possibility of recovering exactly the same signal when a well designed receiver is used. At the same time, another important fact can be highlighted: It is well known that voice-band data modems are incapable of achieving the same voice quality as an ordinary telephone in the speech bandwidth. It is interesting to discuss this problem in the lab to make clear that there are advantages and disadvantages in using digital communications. The developing of more complex modems, so the bandwidth gap between the analog and digital communications is decreased, is an interesting way of introducing more complex algorithms with an objective in mind. The effect of quantization on the fidelity of the reconstructed signal (that the students can hear) is also highly interesting from a pedagogical point of view. In this first experiment, the understanding of the different frequencies involved in a digital communication system is a very important issue. In this lab setup, the student must continuously work with the bit rate, the symbol rate, the sampling frequency (that must fulfill the sampling theorem) and the simulation sampling frequency (several times higher than the Nyquist frequency, so the discrete-time signals, when drawn in the PC screen, looks like its continuous-time counterparts in the oscilloscope).

##### 4.2. Baseband communications and baseline wander

Usually, some experiments with line codes are conducted in this type of labs. The soundblaster exhibits a bandpass behaviour with the lower cut-off frequency around 40 Hz. This produces a considerable amount of baseline wander, that advanced students can even model mathematically. The effect of the baseline wander is very easily observed in the oscilloscope by using the eye diagram. Figure 3 shows a MATLAB graphical application called `codlin` that allows the students to generate baseband signals. It presents the communication signal, the power spectral density, and the eye diagram. It is possible to simulate the channel (filtering the signal and adding noise) and to send the result to the sound card in

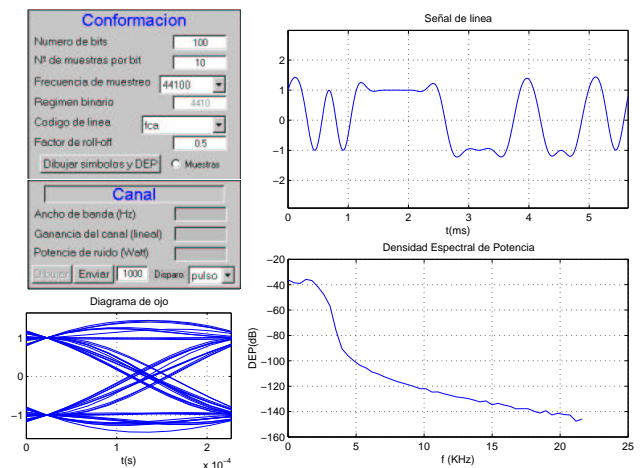


Fig. 3. Baseband data transmission window

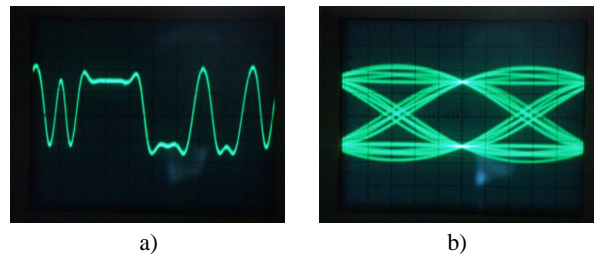


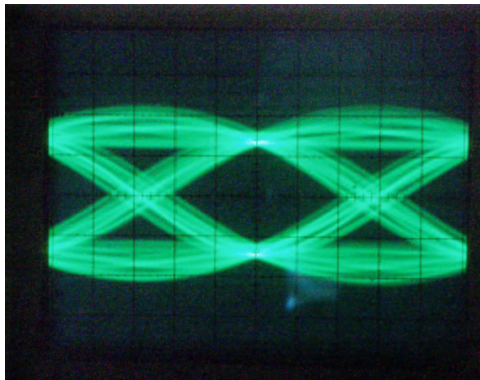
Fig. 4. Oscilloscope pictures of a 4410 bps raised cosine signal ( $r = 0.5$ ) signal (a) and its eye diagram (b)

order to see the signal and its eye diagram at the oscilloscope display. Figure 4 shows an oscilloscope picture of the same 4410 bps raised cosine signal ( $r = 0.5$ ) and its eye diagram (the reader can observe the great similarity between these oscilloscope pictures and the discrete-time signals shown in Figure 3). In Figure 5 it is possible to appreciate the baseline wander of a 441 bps raised cosine signal due to the sound card highpass behaviour.

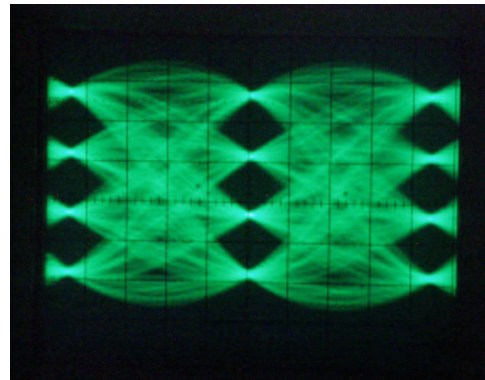
Note that, besides the transmitted signal, it is possible to send to the sound card right channel a symbol clock signal or a frame pulse signal (see the `disparo` box in Figure 3) to synchronize the analog oscilloscope and show the obvious equivalence of the discrete-time signal in the PC and the analog one in the oscilloscope.

##### 4.3. Passband communications

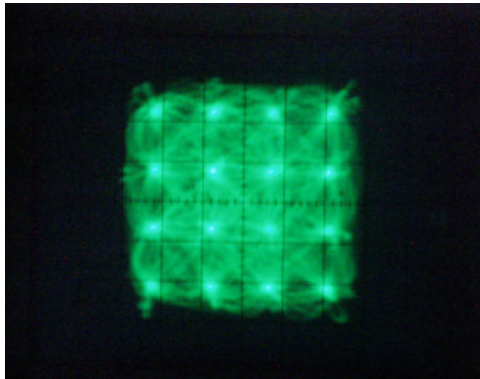
Any modulation may be easily implemented in this setting using a carrier frequency around the center of sound card passband (i.e.  $10 \text{ KHz} @ f_s = 44100 \text{ Hz}$ ). The student must produce the MATLAB code and send the communication signal to the sound card. For example figure 6 and figure 7 show different graphical analysis that the student may conduct with a 16QAM signal. The modulated signal, or a distorted or noisy version, can be acquired by the other student of the team, who will have to demodulate the signal facing the carrier and symbol synchronization problem, considered next.



**Fig. 5.** Baseline wander effect on a 441 bps raised cosine signal ( $r = 0.5$ )



**Fig. 7.** 16QAM eye diagram



**Fig. 6.** 16QAM constellation

#### 4.4. Synchronization

Students find quite difficult to understand the problem of synchronization. The idea that it is impossible to produce two clocks of exactly the same frequency, and that extremely small clock difference will accumulate to a point that finally makes communication impossible it is not easily understood. Even the idea that it is impossible to know the optimal sampling instant without some kind of signal processing is obscure to some students. The fact that we have a two-people team, and that we continuously play the game of one sending a message for the other to recover (for example to implement a chat between both students) makes it clear to the student from the beginning that he has to do something to recover the optimal sampling instant to be able to recover automatically the information in a baseband signal, or that she/he has to synchronize the local phase of the oscillator to that of the transmitter in order to recover passband communication signals.

#### 4.5. Real time operation

The fact that the MATLAB toolbox has an unlimited buffer, allows on line algorithms to be analysed in a simulated real-time. Adaptive equalization algorithms, as the LMS, may be easily implemented by taking sequentially samples from the buffer and doing

the necessary processing. Nonetheless, it is important to make the students aware of the implications of “real” real-time operations.

### 5. CONCLUSIONS

In this paper we presented a DSP-based digital communications laboratory. Based on very simple hardware, it can reproduce most of the problems that an engineer has to consider when designing digital communication systems. The main theoretical approach is to give the same importance to discrete-time and continuous-time communication systems. The lab combines personal and team work from the very beginning, because the students have to implement “real” communication systems and communicate through them. The teaching experience with this lab has been highly successful. In the future we plan to include a digital signal processor in this setting (basically to complement the DSP labs) that could be used at the final sessions of the digital communications laboratory to separate the data generator (PC) from the discrete-time part of the modem (the digital signal processor), and to introduce the student to real implementation issues, as the effect of finite precision.

### 6. REFERENCES

- [1] B. Sklar, *Digital Communications, Fundamentals and Applications*. Prentice Hall, 1988.
- [2] S. Haykin, *Communications Systems*. John Wiley & Sons Inc., 3rd ed., 1994.
- [3] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*. Prentice Hall, 2nd ed., 1997.
- [4] Creative Labs Technical Support web page, <http://www.americas.creative.com/support/>.
- [5] *Data Acquisition Toolbox User's Guide v2.2*, The MathWorks, Inc., 2002.
- [6] MathWorks Data Acquisition Toolbox web page, <http://www.mathworks.com/access/helpdesk/help/toolbox/daq/daq.shtml>.