Introduction to Communication Systems

Luciano Sbaiz, Patrick Thiran, Rüdiger Urbanke

September 18, 2007

Contents

Chapter 1

Signal processing

1.1 Signals and systems

1.1.1 Introduction

In this module, we are going to talk about signal processing. What is a signal? A signal is a mathematical representation of a physical quantity, for example the air pressure corresponding to a certain sound. We obtain signals either measuring them with sensors (e.g. a microphone) or generating them (e.g. a synthesizer of a musical instrument). What can we do with signals? We can transform them using a “system”. A system takes a signal at the input and produces a signal at the output. The output signal has some properties we are interested in. Take the microphone of the previous example. We can consider it as a system that transforms the sound pressure into an electrical signal. Sound pressure is difficult to amplify or record (even if one can think of a purely pneumatic system to record sounds) so we convert sound pressure into an equivalent electrical signal. However, the conversion has some limitations. For example, you cannot use the same microphone to record a singer and the noise of a jet turbine: there is a limit to the range of the signal that you can record. Also, normally you cannot record ultrasounds with a normal microphone (e.g. suppose you want to record a bat) i.e. there is a frequency limit. In summary, a system like a microphone is not an ideal converter of signals. This is a general fact about systems: they have some qualities we are interested in and some others that we don’t like. The work of an engineer is often to design a chain of systems so that the bad qualities are minimized while keeping the good qualities, at least for a reasonable range of parameters of the input signal.

Now, how is this related to the transmission/record of audio sounds and mp3? If you take for example an internet connection, you can consider it as a system that receives bits on one side and outputs bits on the other side. The bit is the information unit and corresponds to the information carried by a signal that can take only two values. For an internet connection, input and output can be very far apart. Unfortunately, when you connect your computer to the internet you realize how long it takes to transfer a web page in some cases. The system has some limitations, exactly as in the case of the microphone. For example, there is a maximum number of bits that you can send per second (what we call bit rate). Also, on this type of connection you can send only bits. However, a sequence of bits is not that interesting but we want to transmit audio, images, videos, texts etc. Another problem is the transmission delay. The system performs some processing on each bit, and that takes time. Also there is a limitation to the propagation of signals on cables and optic fiber, so at the end there is a transmission delay which is never zero. Moreover, there are errors! From time to time some of the bits that are transmitted are not detected correctly. It could be one over one billion but the consequences can be catastrophic e.g. for a computer program. There are other types of limitations and the same for other media such as optic disks like Compact Disks (CD) and Digital Video Disks (DVD) or tapes and mobile phones etc. In conclusion, we need some additional systems that we add to complete the chain. Every system of the chain will do a transformation on the signal so that the next system will receive a signal compatible with its input. Now we can answer the question: what is mp3. Mp3 is a standard that specifies a family of bit sequences. These sequences are used to describe audio signals. Implicitly mp3 defines how you can build a system that takes an audio signal e.g. from a microphone, and transform it to something suitable to be transmitted or recorded using few bits. Since the standard specifies only how the output of the system must be, there is a lot of freedom on the design of the system itself. Hence, two mp3 files of the same song may sound differently. The standard is the result of compromises, first of all between quality and number of bits that have to be sent.

In this module we will see some principles of signal processing and we will describe some of the modules that are used to design an audio coder. Unfortunately, an accurate description of these modules would need some advanced math, so you will have to wait a couple more years. Hopefully, this will motivate you to learn math in the courses of the first two years.

1.1.2 Signals

Intuitive idea of signal

Let us be a bit more formal with the definition of a signal. We already said that the signal is a function associated with a certain physical quantity. We know that a function has a domain and a codomain. Which are the domain and the codomain of the signals? The typical domain is time, so if you believe that time is continuous (probably a pedantic physicist wouldn't agree with that) the domain is $\mathbb{R}$. What about the codomain? It is something that we can measure so it is an integer or a real number. Often, we prefer real numbers because of the nice properties they have (you will see that in other courses).

Example: temperature vs. time. Continuous and discrete-time signals

Suppose we want to measure the temperature in Lausanne over a certain time. What type of signal is this? The domain and the range can be represented with real numbers. We call these type of signal **continuous-time signals**, because the time axis is continuous (note that the signal is not necessarily a continuous function!). So one can model temperature as a function defined on $\mathbb{R}$ and values in $\mathbb{R}$. However, it would be difficult to "measure" such function! Our instruments measure a certain quantity only on discrete-time instants. Even if we use a mercury thermometer, we would need an operator permanently in front of the thermometer (and he could not register the measurement). We could plot the temperature directly on a sheet of paper, that would allow keeping the real function, but later it would be very difficult (even if not impossible) to do something with the drawing of the function (for example to compute the average temperature over a year). However, one can take into account that temperature changes very slowly, so we can decide to measure it for example every hour. It is very unlikely for the temperature to be very irregular between two measurements, so we can be satisfied with this approach. We call this type of signals **discrete-time signals** (because the signal is defined on a discrete set of points). Modern signal processing deals almost always with discrete-time signals even if reality is continuous. We will see later how this is possible, however you already have the intuition that we can at least "approximate" a continuous-time signal with a discrete-time signal. we have simply to take enough measurements on discrete points of time axis. We call this procedure **sampling** (see Figure 1.1). We will see in the third lecture how this works.

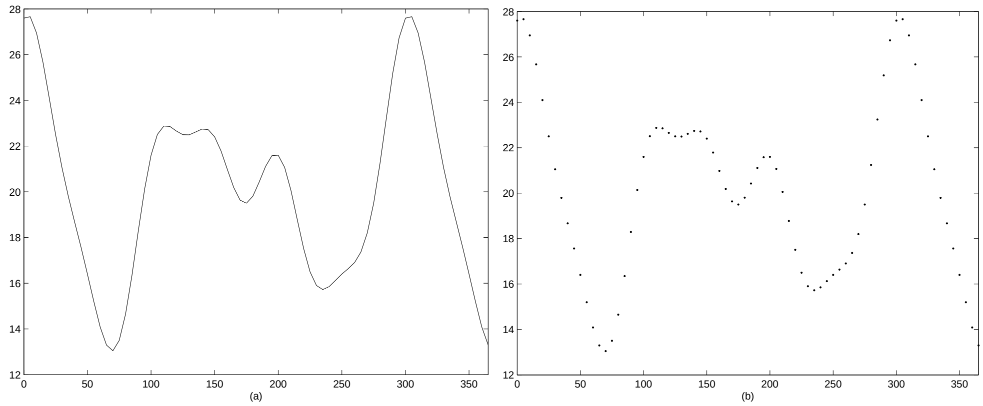


Figure 1.1: Example of representation of continuous and discrete-time signals. (a) Continuous-time signal. (b) Discrete-time signal.

Signals in the computer

Let's develop further what we have seen in the previous example. First of all, let's answer to the following question: what type of information can we store in a computer (or a disk or send through the net)? Can we store a real number? One can remember that a real number is something of type "3.1416...". In general, what comes after the dot can be an infinite non-periodic series of digits. Can we store that in a computer? Surprisingly we can store "some" of these numbers. In fact suppose that a certain quantity can take only the values $1, 1/3, e, \pi$, then we can represent this quantity with simply two bits. In fact, with two bits we have four combinations that correspond to the four values that we want to represent. For example, if the two bits are b_0 and b_1 we choose

Value	b_0	b_1
1	0	0
1/3	0	1
e	1	0
π	1	1

This defines perfectly the values that we want to record even if the values don't have a finite decimal representation. As you see, this is a general trick but we can use it only for a finite set of elements. In fact, all the resources of a computer (or a communication system) are limited. Even if today we can store many bits in a computer, the number of combinations of all the bits remains finite and so is the number of values that we can represent (but we have the freedom of choosing what we associate to each combination). Computer scientists use several representations for integers, real numbers and other numeric values. To use these representations, one has to *approximate* the actual value with one element of the representation.

What about signals? Which are the signals that we can record/process/transmit using digital systems? Can we treat continuous-time signals? As in the previous case we can show that we can represent "some" continuous-time signals. In fact, take for example the signal:

$$y(t) = at^2 + bt + c.$$

You probably remember that this function is a parabola. Can we record this type of signals on a disk? The answer is yes, since every parabola is represented by three real numbers a, b, c and we know how to represent a finite set of real numbers. As a result we can record a finite set of continuous-time signals. If we know that a certain physical quantity (like the temperature of the previous example) has a parabolic variation we can use a device that allows measuring the parameters of the parabola and store them. Later, we will be able to retrieve the parameters and reproduce the parabola with another device and do some processing on it.

What about transmission of continuous-time signals? Which are the signals that we can transmit? We can think in the same way as in the previous example but now we don't need to store the parameters since we transmit them. The number of parameters that we can send per second is a finite quantity, so we can transmit signals that "locally" can be represented with a finite number of parameters. For example, think of a signal which is built of segments of one second and each segment is a parabola. We can measure the parameters of parabolas segments and send them to the receiver. There, we are able to reconstruct exactly the input signal simply generating the pieces of parabolas corresponding to the parameters. We can do this only because the input signal is taken in a set of functions that can be described locally by a finite number of parameters. We call the sets of this type "set of signals with finite rate of innovation."

What about audio signals on a computer? Can we describe them with a function that has a finite number of parameters? If we consider a generic sound no. We have to approximate it with a function that has a finite rate of innovation. A common way is to take samples of the continuous-time signal as we have seen in the temperature example. We will see this in more detail in the third lecture.

Definition of signal

Let's see a more formal definition of signal. The first concept that we need is that of the cartesian product. We assume that you know the concept of set as a "collection of elements".

Definition 1. *The cartesian product of two sets A, B is the set*

$$A \times B = \{(a, b) | a \in A, b \in B\}$$

that is, the cartesian product is the set of all the possible ordered pairs of elements of A and B.

Definition 2. *A relation R is a subset of the cartesian product of two sets A, B, i.e.*

$$R \subseteq A \times B.$$

In other words, a relation puts in correspondence points of two sets A and B. However, a point in A can be in relation with more than a point in B (see Figure 1.2). For a function we don't allow that and every point in A is in relation with exactly one point in B:

Definition 3. *A function is a relation from A to B such that:*

1. *for each element $a \in A$ there is an element $b \in B$, such (a, b) is in the relation*

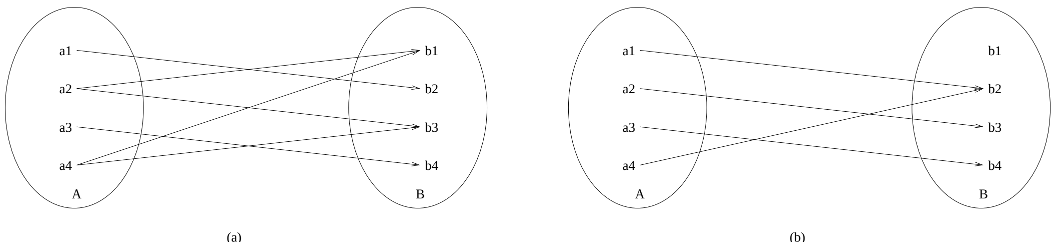


Figure 1.2: Relations and functions are both subsets of the cartesian product of two sets A and B. (a) A relation defines a set of correspondences between elements of two sets. (b) A function $f : A \to B$ is a relation that assigns to each element of A a *single* element of B.

2. if (a, b) and (a, c) are in the relation, then $b = c$.

Note that, points in B are not constraint to be in relation with exactly one point in A. They can be in relation with multiple points or with no point at all.

It is common to write the function f as

$$f: \quad A \to B$$

where A is called the domain and B the codomain.

Now we can define a signal as the function

$$\begin{array}{rccc} f: & A & \to & B \\ & a & \mapsto & f(a) \end{array}$$

where A is $\mathbb{R}$ for **continuous-time signals** and $\mathbb{Z}$ for the **discrete-time signals**. You may be surprised that we consider sets with an infinite number of elements. In reality, we always start our measurement (or transmission or acquisition) at a certain time and probably we will stop it at a certain time in the future. However, we prefer to define mathematically the signals on the entire real axis to simplify the notation for many operations. We can simply imagine that we extend the signal outside of the actual range. Of course, we will have to know how to do that extension if we want to build a device for signal processing.

The set B is $\mathbb{R}$ or $\mathbb{C}$. The complex values are used in some cases because they give a simpler notation, but most physical quantities are actually real. We have seen that we cannot represent values taken in an infinite set such as reals. However, the approximation error is normally small and in the equations we often neglect the rounding errors.

The sinusoid

Let us see a type of signals that engineers like to use (in the second lesson we will see one of the reasons). It is the sinusoid

$$y(t) = P \sin(2\pi f t + \phi) \qquad t \in \mathbb{R} \tag{1.1}$$

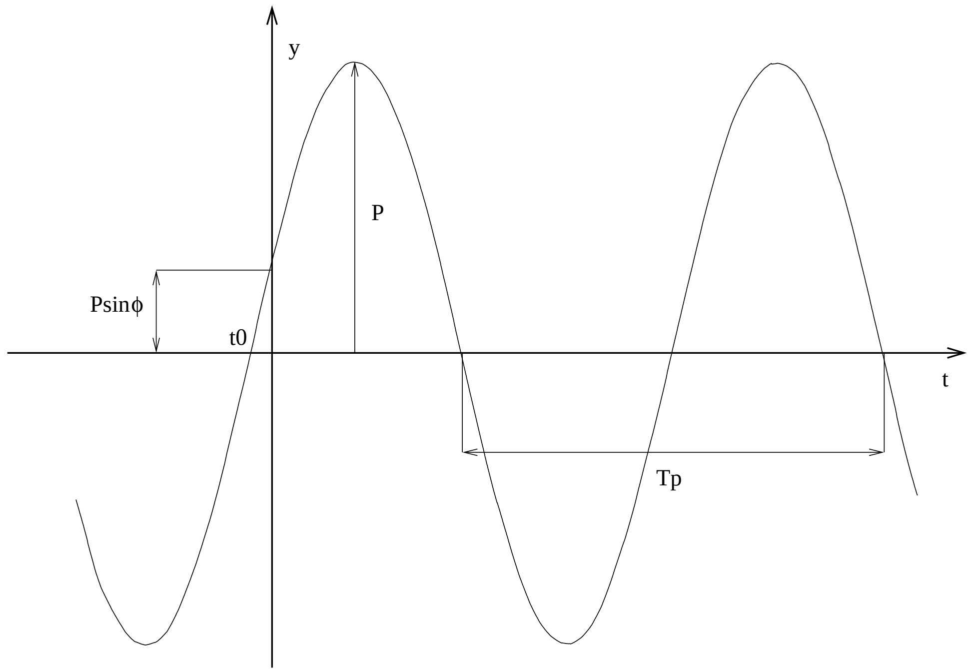


Figure 1.3: The parameters of a sinusoid.

This is a continuous-time signal. The time t is on the real axis and the instantaneous value y is also real (actually it belongs to the range $[-P, P]$. The parameter P is called **amplitude** and f is the **frequency** and is measured in **Hertz** (**Hz**). The frequency corresponds to the number of periods completed in one second. For example, a frequency of 440 Hz means that the sinusoid completes 440 cycles per second[1]. Alternatively, one can specify the time needed to complete one cycle of the sinusoid. This is called the period $T_P = 1/f$. The sinusoid becomes

$$y(t) = P \sin\left(2\pi \frac{t}{T_P} + \phi\right) \tag{1.2}$$

Often we want to get rid of the factor 2π so we prefer to measure the frequency in **radians per second**. The symbol ω is commonly used to denote the frequency in radians per second. Of course, the relation between the frequency in Hertz and in radians per second is:

$$\omega = 2\pi f = \frac{2\pi}{T_P} \tag{1.3}$$

The **phase** ϕ can be considered as a shift of the sinusoid along the time axis. In fact we can write

$$y(t) = P \sin(2\pi f(t - t_0)) \tag{1.4}$$

with $t_0 = -\phi/(2\pi f)$ (see Figure 1.3).

[1]This is actually the note "middle A" in western music (a "La")

This was a continuous-time sinusoid. We can define the discrete-time sinusoid:

$$y(n) = P \sin(2\pi f_D n + \phi) = P \sin(\omega_D n + \phi) \qquad n \in \mathbb{Z} \tag{1.5}$$

You noted that now we use the variable n instead of t to stress that the signal is defined on the integers. If you plot the discrete-time sinusoid and you compare with the continuous-time sinusoid you will see that they look quite similar. However, there is a main difference that concerns the periodicity of the discrete-time sinusoid. Remember that a function $h : A \to B$ is periodic with period p if

$$h(x) = h(x + lp) \quad \forall x \in A, l \in \mathbb{Z},$$

i.e. the signal repeats itself every shift p along the time coordinate. It is trivial to verify that the sinusoid is a periodic signal with period $p = T_P$. However, the discrete-time sinusoid in general is not periodic. In fact, if the frequency f_D is not rational, there is no value of $n \in \mathbb{Z}$ such that $f_D n$ is an integer. That is, the angles on which we compute the sin are always different and the signal never repeats.

Multidimensional signals. Images, television and video

The signals that we have seen so far are one-dimensional since they are function only of time. However, many physical phenomena cannot be described by a function of a single coordinate. A typical example is a measurement on surface (temperature, pressure, deformation, etc.). The position on a surface is described by two coordinates so it is natural to model the measurement with a two-dimensional signal. A multidimensional signal is a function

$$f : \quad A \to B$$

as for a one-dimensional signal, but in this case the domain A is obtained composing $\mathbb{R}$ and/or $\mathbb{Z}$ with the cartesian product. For example, consider the intensity of light reaching the film of a camera. We can define a system of coordinates on the surface of the film, and the intensity is described by a two-dimensional signal:

$$i_C : \quad \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

In the following, we use the shorthand $\mathbb{A} \times \mathbb{A} = \mathbb{A}^2$ to simplify the notation.

What we have discussed on recording of one-dimensional signals can be repeated for two-dimensional signals. Again, we can only store images that we can describe with a finite number of parameters. Normally the parameters are the values of the image on a uniformly spaced grid. We obtain a discrete image:

$$i_D \ \ \mathbb{Z}^2 \to \mathbb{R}.$$

Today, it is very easy to obtain discrete images, since with have digital cameras. A digital camera instead of a film contains a Charge-Coupled Device (CCD). The CCD has many elements sensitive to light. Such elements are organized as a matrix of points called pixels (the name "pixel" is derived from the abbreviation of "picture element"). A recent camera can have several

Figure 1.4: A discrete image is a function of two indexes defining the pixel intensity.

millions pixels (or megapixels). Each sensor of the CCD measures light intensity corresponding to one pixel, hence we obtain directly a discrete image without conversion (see Figure 1.4).

Note that we can mix continuous and discrete coordinates and define signals like:

$$i_{CD}: \ \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{R}$$

We can obtain a signal like that by taking lines of a continuous image at discrete positions (this procedure is called "sampling" and it will be better explained in the third lecture). An example, of a signal of this type is a TV signal. We consider black and white signal television for the time being. At the beginning of television everything was analog (actually TV was invented before digital electronics). The TV signal was obtained using an electronic beam to scan a surface sensitive to light. The result can be described with the signal that we have seen. Today cameras contain also CCDs but the signal that is broadcasted is still of the same type.

On TV you don't have just a static image but you have an image that changes over time, i.e. a video signal. We can describe a video signal in continuous-time as the function:

$$v_C: \ \mathbb{R}^3 \rightarrow \mathbb{R}$$

i.e. a video is a function $v(x, y, t)$ where x and y are the spacial coordinates and t the temporal coordinate. Where can we find such a signal? We can find it on the surface of the sensor of any video camera. However, the sensor operates a transformation to a discrete-time signal. An analog camera (still commonly used in broadcasting) "samples" along the temporal and the y coordinate giving a signal of the form:

$$v_{CD}: \ \mathbb{R} \times \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$$

i.e. now we have a signal $v_D(x, m, n)$ where $x \in \mathbb{R}$ is the continuous horizontal position along a certain line, m is the index of the line and n is the temporal index (see Figure 1.5).

To store a video on a digital support we need to sample also along the lines (as consumer digital cameras). We obtain a signal of the form

$$v_D: \ \mathbb{Z}^3 \rightarrow \mathbb{R}$$

i.e. now a video is a function of three indexes corresponding to position and time. This is the type of videos that are stored on computers, DVD and CD.

What about color? Vectorial signals

You probably notice that in the previous discussions we neglected color, i.e. what we said is correct for black and white images and videos. How do we deal with color? We see colors because light is composed by different spectral components, i.e. components with different frequencies. Our eyes have cells which have different sensitivities to the spectral components. There are three types of sensor, so a certain color can be described by three quantities. Hence, a color image is also described by three values for each position. We can do that using three distinct signals. However, the three images are strictly related, so we prefer to use a vectorial signal (see

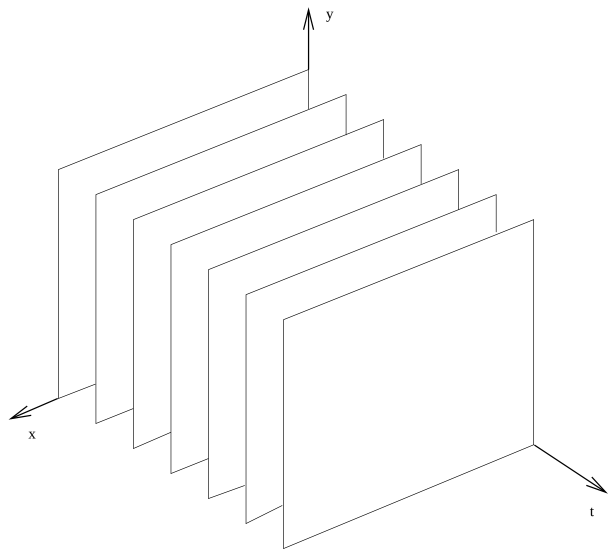


Figure 1.5: A video sequence can be considered as a signal defined on $\mathbb{Z}^3$. The temporal index defines an image of the sequence (called also a **frame**), the remaining two indexes determine the position of a pixel.

Figure 1.6: Four images (called also *frames*) of a video sequence.

Figure 1.7). We can consider a vector as an ordered set of numbers, so it is also an element of a cartesian product of $\mathbb{R}$ with itself. A color image is represented on a computer with a signal:

$$i_D^{(3)}: \quad \mathbb{Z}^2 \to \mathbb{R}^3$$

and a color video with the signal

$$v_D^{(3)}: \quad \mathbb{Z}^3 \to \mathbb{R}^3$$

You probably know other types of signal that can be described using a vectorial representation. For example, stereo audio: you have two channels that are related and synchronized on time, so it is convenient to represent them using a vectorial signal with two components. The principle can be extended to multichannel audio which are used in cinemas and home cinema systems. They normally use five or six channels to record audio. With the progress of technology sensors and processing devices are becoming cheaper and smaller. It is reasonable to expect that we will have more and more applications that use arrays of sensors.

Symbols and sequences

We have seen examples in which temporal or spatial information is represented by functions of a variable representing time or space. In many situations, information is represented as sequences of symbols that represent data or an event stream. The main difference with signals is that the values of a sequence are taken in a set that is not directly related to a physical quantity. For example, in a text file each letter can be considered as a symbol but it doesn't correspond to something that we can measure. As you see, symbols are abstract entities which do not correspond directly to a specific physical representation (i.e. a certain signal.) However, they need a physical representation to exist. For example, you can read this text on a computer monitor or printed on paper. In both cases, there are certain signals associated to each letter of the text which will be different for the two media. However, the information carried by the signals, i.e. the symbols, is the same. If you print the text with a different quality or you change the settings of the monitor you will change the signals used to represent the symbols but not the symbols themselves. Hence, the concept of symbol is related to the semantics, i.e. the meaning that we associate to a class of signals.

We need devices to change the representation of symbols. For example, a text is represented in the computer memory as a certain combination of charges on certain components. To show the text on the computer screen, we need to measure the charges corresponding to each letter and convert to a set of points that represent each pixel. The status of each pixel is used to generate the signals that drive the CRT of the monitor. There are many devices that do this type of conversion: printers, scanners, modems, CD reader/writer, keyboards and many others.

What is a symbol exactly? Since it is an abstract object, it is arbitrary to define what a symbol is. In a text file, are the symbols the letters or the words? It seams that we can define a hierarchy of symbols: letters are grouped together to form words, words are grouped to form sentences and so on (see Figure 1.8). At the basis of the hierarchy we have signals, i.e. the physical support, at the higher levels we represent symbols with higher information content. This is a general concept applied by engineers. Information is organized in layers, each layer is associated

Figure 1.7: A color image and its decomposition into three color components. (a) Original image. (b) Red component. (c) Green component. (d) Blue component.

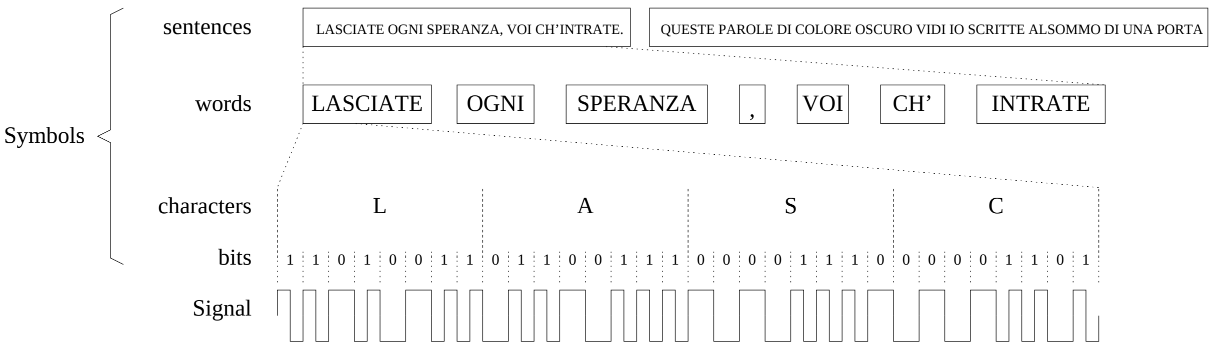


Figure 1.8: Information can be organized in layers. The bottom layer represents signals. The higher layers represent information using symbols.

to a certain representation. We can also consider the operations that we do on the information as a transformation at a specific layer. For example, the text file can be transmitted using a modem: that changes the signal used to represent the text. We can operate at a higher level of the hierarchy and change a character with a text editor, or we can change a word. At a higher level we can change the meaning of a sentence and so on.

1.1.3 Systems

The device (or the software) that realizes a transformation of the information is called a "system". This is a generic term that denotes something that takes a sequence or a signal at its input and produces a sequence or a signal at its output. Mathematically we can describe a system as a function that takes a function at its input (the input sequence or signal) and produces a function at the output (the output sequence or signal). In signal processing we are interested in systems whose input, output or both input and output are signals.

There are many cases where we need a system to change the media on which the information is represented. For example, to send a text file over a telephone line, you have to convert it to a signal which is similar to voice. We can describe mathematically the device that performs the transformation as a function that takes symbols at its input, i.e. the characters, and produces a voice-like signal at its output. We call this function a **modulator**. At the receiver side the voice-like signal is transformed to a sequence of symbols by the **demodulator**. A device that is composed by a modulator and a demodulator is called a **modem**. Note that the use of symbols to represent the input of the modulator and the output of the demodulator is a mathematical formalism to get rid of the representation of such symbols using signals. A real device always deals only with signals.

There are many other cases where we need to convert signals to use a different medium to store/transmit information. Since every medium has some specific characteristics we need some specific devices. Among the different media we have cables, optic fibers, optic disks (CD, DVD, etc.), magnetic supports (disks and tapes), paper, air (acoustic signals) and many others.

Other types of systems deal with the transformation of signals in order to improve some qualities.

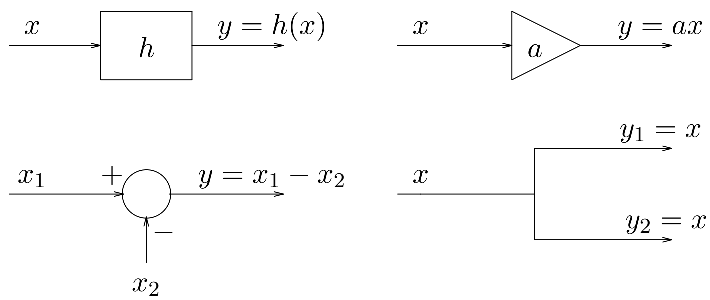


Figure 1.9: Symbols used in block diagrams.

For example, we can think of the tone control of an HiFi chain: the signal is filtered to amplify or attenuate certain frequencies. We can think to more sophisticated examples of system for enhancement. For example to improve an image you acquired with a digital camera (e.g. to eliminate the effect "red eyes").

Another type of systems are used to control a physical process. Think for example to a heating system. There are a number of sensors that measure temperature and a number of heating devices that we can control. A system is needed to process the measurements and compute the control signals so that some conditions on the temperature are satisfied. For example, we can impose a certain constant temperature that we want to keep with the minimum error, or we can impose a certain temperature profile over time.

Block representation. Subsystems

We often represent systems with blocks. A sequence of systems is represented by a chain of blocks interconnected by arrows. We can also write names for the signals at the input and output of each block. Some common systems are represented by special symbols. For example, the addition of two signals is represented by a circle. If we want to send a certain signal to two systems, we simply draw a bifurcation (this can also be considered a system). In Figure 1.9 some block symbols are shown.

The block representation is a way of representing a complex system in term of subsystems. Each subsystem is represented as a "black box," that is, we know the functionality of the block but we don't put our attention to the way it is implemented. It is the layer representation that we have seen previously. If you are programming a computer, you have to deal only with the language syntax and not with the current flowing in each of the millions transistors of the processor. If the layers below are working properly, the method works and you can concentrate only on that that you are designing.

An example of modulator: dual-tone multifrequency (DTMF)

Let us see an example of a modulator. It is called dual-tone multifrequency and used in telephony to transmit a telephone number through the telephone line. There is one of these in every

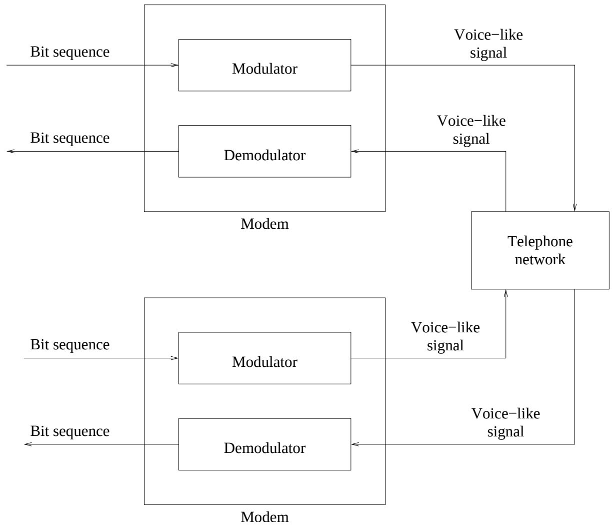


Figure 1.10: Voice band data modems.

telephone.

The system is based on a keyboard with twelve keys. On the keys the ten figures plus the special symbols "*" and "#" are represented. The symbols are organized as a rectangle of four lines and three columns. To each line and to each column distinct frequencies are associated. When the user presses one of the keys, the modulator generates a signal by adding two sinusoids of frequencies corresponding to the line and the column of the key. For example, a "0" is represented as a sum of two sinusoids with frequencies 941 Hz and 1,336 Hz.

We will see in the second lecture how we can demodulate the output signal and recognize which key was pressed, but you can already think of something similar to the tone control of an HiFi chain to separate the different sinusoids.

Quantization of signals

In the previous sections we have discussed about several types of signals. We said that we assume that they take values in $\mathbb{R}$. However, we have seen that computers and communication systems have finite resources and they can only deal with finite sets of values. Therefore, real numbers are approximated with appropriate values. There are different ways to choose the set of values. Every choice corresponds to a different amount of memory needed to represent the values and a different precision of the representation.

In signal processing, we call **quantization** the procedure of conversion of a real number to a finite size representation. We call **quantizer** the device that performs the conversion. Sometimes, we want to change from one representation to another. For example, this is done to reduce the amount of memory needed to store information. Even in this case we talk about quantization.

Since quantization is a transformation on signals we can call it a system. Let us call I the finite

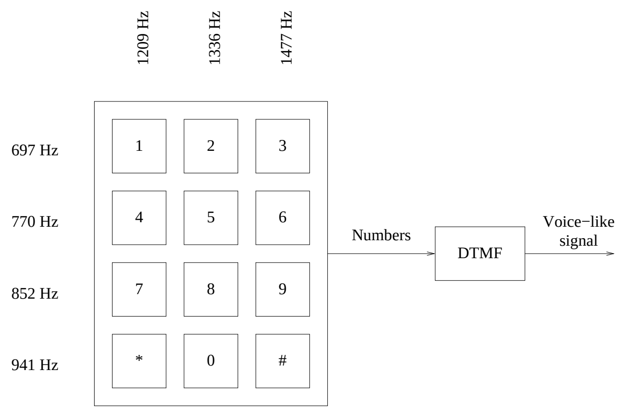


Figure 1.11: A dual-tone multifrequency system converts numbers from a keypad into a voice-like signal.

set that we decide to use to approximate real numbers. The quantizer, for one-dimensional discrete-time signals, is represented by the function:

$$q: \quad [\mathbb{Z} \to \mathbb{R}] \to [\mathbb{Z} \to I].$$

In the same way we can define the quantization of other types of signal.

A quantizer that converts values in the representation I_A to representation I_B is represented by the function:

$$q: \quad [\mathbb{Z} \to I_A] \to [\mathbb{Z} \to I_B].$$

There is a type of quantization that you probably know. It is the rounding and the truncation of real numbers. These are ways to map a real number to an integer. Integers are still an infinite set so we have also to fix a minimum and a maximum value to the values that we are going to represent. Let us see how it works considering an example. Suppose we want to acquire an image with a computer. In order to do that, we need to convert the output signal of a camera to a digital representation that can be stored in the computer. The output of the camera is an an analog signal v. There is a maximum value of the light intensity that can be measured. Suppose that such an intensity corresponds to the output signal V_0. We also know that $v \geq 0$, since light intensity is not negative. To convert such a signal to a digital representation, we need a device called Analog to Digital Converter (ADC or AD). The AD combines in the same device the quantizer and the sampler. The sampler transform the signal from continuous-time to discrete-time. We will see it in the third lecture. The quantizer represents the amplitude using a finite number of values, L. We normally choose $L = 2^b$, i.e. a power of 2. In this way, we use all the combinations of b bits to represent the values. We show an example using $b = 3$ in Figure 1.12. The input range is decomposed in intervals of size

$$\Delta = \frac{V_0}{2^b}.$$

The output y is computed by

$$y = q(v) = \left\lfloor \frac{v}{\Delta} \right\rfloor \Delta.$$

In other words, the values of a certain interval are mapped to the *minimum* value of the interval. The quantization error $e = v - y$ is always positive and its maximum value is Δ (provided that the input signal remains in the expected range).

In practice, the number of bits b is normally higher than 3. For example, it is common to represent a black and white image with values in the range $[0, 255]$. This range can be represented with b=8 bits, i.e. one byte of memory. A black and white image becomes a function

$$i_D : \ \mathbb{Z}^2 \to I_8,$$

where I_8 is the set of integers in $[0, 255]$. Color images need three values for every pixel so they require 24 bits, i.e. three bytes. Therefore a color image is a function

$$i_D^{(3)} : \ \mathbb{Z}^2 \to I_8^3.$$

For audio signals, we use even more bits. Compact-Disks are recorded using 16 bits and DVD-Audio uses 24 bits.

Image warping

Let us see another system that transforms an image to another one. The main idea is to take the pixels of the input image and reorganize them to obtain the output image. To move the pixels we use continuous functions, so the result is a deformation of the image. This is also a system. For color images it has the form

$$w : \ [\mathbb{Z}^2 \to I_8^3] \to [\mathbb{Z}^2 \to I_8^3]$$

where I_8 represents an integer represented on 8 bits. To define precisely the warping, we need to specify how the pixel values are displaced. For example, if the input image is $i_I(x, y)$ and the output image is $i_O(x, y)$, we can define:

$$i_O(x, y) = i_I(x', y')$$

with

$$x' = x(1 + \rho(x^2 + y^2))$$
$$y' = y(1 + \rho(x^2 + y^2))$$

where ρ is a parameter. A warping of this type for $\rho > 0$ produces an image similar to one acquired using a "fisheye" lens, i.e. a lens with very small focal length (see Figure 1.13).

Simulcam

Systems can be defined on every type of signal. We can see an example on videos. It is called simulcam and it is commercialized by the company Dartfish in Fribourg (http://www.dartfish.com). The idea is to take two videos of a certain scene. On the scene there are different persons or objects that are moving. We choose one of the two sequences as a reference and we want to add

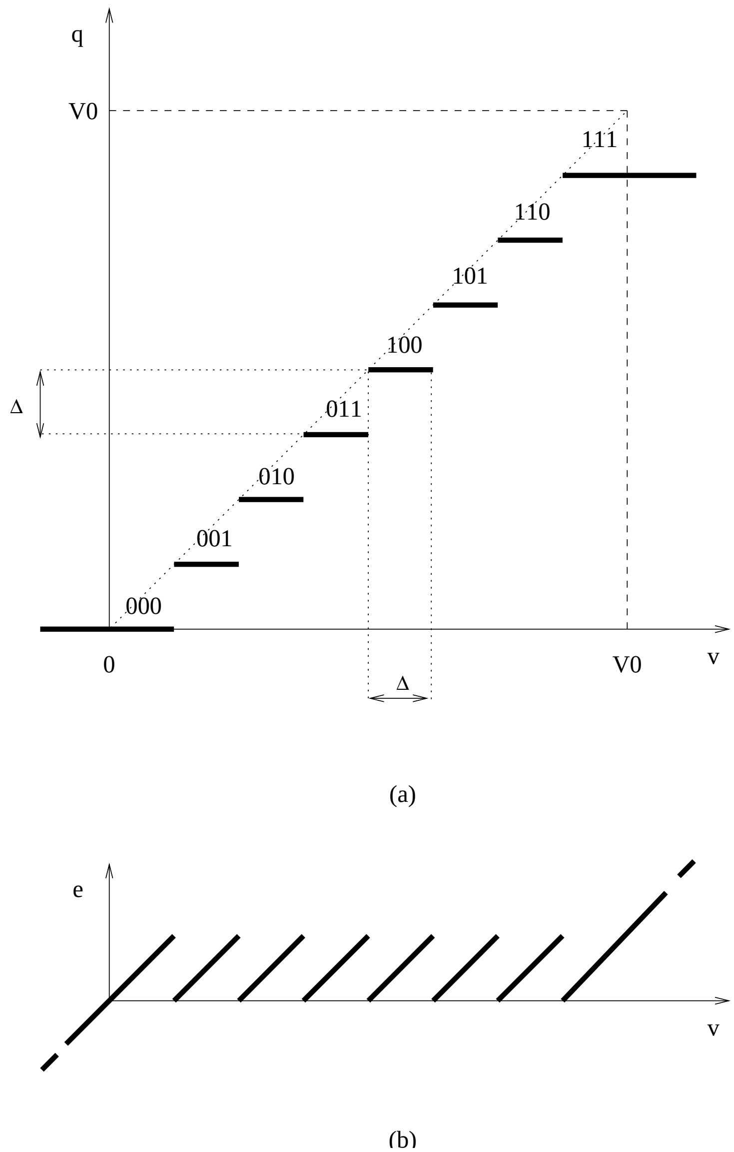


Figure 1.12: Quantization of a real value between 0 and V_0 using 3 bits (8 levels). (a) The input-output relation of the quantizer. Note how the result is obtained by truncation of the input value. In correspondence of each output value a possible representation on 3 bits is shown. (b) The quantization error is positive and smaller than the quantization step Δ. Note that input values outside of the design range $0 - V_0$ would lead to quantization errors bigger than Δ.

Figure 1.13: Image warping can be considered a system that transforms two-dimensional signals. On the left the original image (288×720 pixels). On the right the output image obtained setting $\rho = 2e - 6$.

Figure 1.14: Superposition of two video sequences can be considered a three-dimensional signal transformation

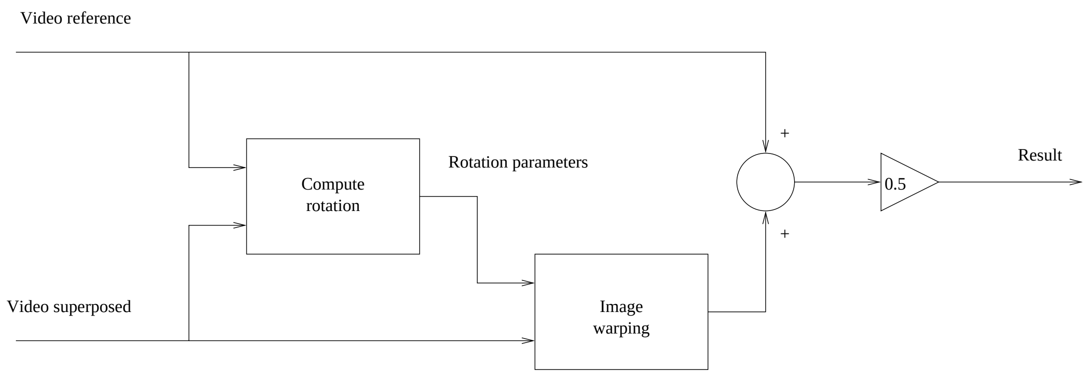


Figure 1.15: Simplified block diagram of a system for video superposition.

the objects of the other sequence on the reference sequence. Note that the objects have to be placed in the correct position with respect to the scene and the camera is moving differently for the two sequences.

This system is a bit complex, but it can be also described mathematically. To simplify the notation, we define the set of color video sequences:

$$V = \{v | v : \mathbb{Z}^3 \to I_8^3\}.$$

Each video sequence is represented as a function of three indexes, for position and time, to three color components represented on 8 bits. Now, we can define Simulcam as a function

$$s : V \times V \to V$$

that is, a system that takes two videos and produces one video sequence.

To define completely the system we need to specify what the function does. We can separate the systems in subsystems. The first subsystem takes each image of the two videos and finds the rotation of the camera on one sequence with respect to the other. It would be a bit complex to explain the details. We can simply suppose that this system tries different rotations in order to minimize the differences on the majority of the pixels of the images. The second subsystem takes the parameters computed by the first block and the video that we are going to add to the reference video. The output of the block is a video sequence where camera rotation has been compensated. This is the warping system that we have seen, applied to every image of the sequence. In this case, the equations used for the warping reproduce the rotation of the camera. The third block takes the reference sequence and the compensated sequence and combine them to keep the moving objects of both and the common background. One can use very sophisticate techniques to perform such an operation. However, an average of the two sequences gives already an interesting result, even if the moving objects can be a bit transparent.

Linear functions and systems

Let us see the definition of linear function. Suppose that A and B are sets on which addition of two elements and multiplication by a real number are defined. For example, $\mathbb{R}$ or $\mathbb{C}$ are good sets.

Definition 4. *A function f*

$$f: \ A \rightarrow B$$

is a linear function if $\forall u \in \mathbb{R}$ *and* $a \in A$

$$f(ua) = uf(a)$$

and $\forall a_1, a_2 \in A$,

$$f(a_1 + a_2) = f(a_1) + f(a_2)$$

The first property is called **homogeneity** and the second **additivity**. When the domain and the codomain are $\mathbb{R}$, a linear function can be represented as

$$\forall x \in \mathbb{R}, \ \ f(x) = kx$$

for some constant k. In fact, it is easy to verify the properties of homogeneity and additivity. The term "linear" comes from the fact that the graph of this function is a straight line through the origin, with slope k. The two properties of homogeneity and additivity can be combined into the **superposition** property:

Definition 5. *f is linear if* $\forall a_1, a_2 \in A$ *and* $u_1, u_2 \in \mathbb{R}$,

$$f(u_1 a_1 + u_2 a_2) = u_1 f(a_1) + u_2 f(a_2).$$

A system is also a function, so we can ask if a certain system is linear. First of all, we have to understand which are the domain and the codomain of the system. We said that this sets could contain signals or sequences of symbols. However, we note that we cannot define addition and multiplication on symbols. For example, you cannot add two text files. So we consider only systems that process signals and produce signals.

For example, take a system that transform a continuous-time signal to another continuous-time signal. In this case,

$$A = B = \{s | s: \ \mathbb{R} \rightarrow \mathbb{R}\}.$$

We can define the addition of two elements of A as

$$(a_1 + a_2)(t) = a_1(t) + a_2(t) \ \ \forall t \in \mathbb{R}$$

and the multiplication by a real quantity $u \in \mathbb{R}$ as

$$(ua)(t) = ua(t) \ \ \forall t \in \mathbb{R}.$$

With these definitions, all the operations in the definition of linearity are defined and we are allowed discussing the linearity of a system. Let us consider for example the system

$$\begin{aligned} d: \quad & A && \rightarrow A \\ & s(t) && \mapsto s(t-1) \end{aligned}$$

that is, the system d delays the input signal by one second. Is this system linear? We check this by verifying the superposition property for two generic signals and constants:

$$\begin{aligned} d(u_1 s_1(t) + u_2 s_2(t)) &= d((u_1 s_1)(t) + (u_2 s_2)(t)) = d((u_1 s_1 + u_2 s_2)(t)) \\ &= (u_1 s_1 + u_2 s_2)(t-1) = u_1 s_1(t-1) + u_2 s_2(t-1) \\ &= u_1 d(s_1(t)) + u_2 d(s_2(t)). \end{aligned}$$

Note that the third equality is a consequence of the particular system that we consider, and the others equalities of the operations defined on signals.

We show now that quantization is not a linear system. Let us take for example a quantizer that converts real numbers to integers represented on 8 bits:

$$q: \quad [\mathbb{Z} \to \mathbb{R}] \to [\mathbb{Z} \to I_8]$$

Suppose that q converts the value of the input signal to the closest integer on 8 bits. For example, 12.3 is converted to 12, but 12.7 is converted to 13. At this point, it is easy to see that the quantizer is not linear. In fact, for example

$$q(4.3 + 5.4) = q(9.7) = 10$$

but

$$q(4.3) + q(5.4) = 4 + 5 = 9$$

Since there are at least two input signals for which the property of additivity is not verified, the system is not linear.

We have seen that we need quantization if we want to process signals with a computer. This implies that most of the systems that we can build are not linear. However, in practice, quantization is designed to introduce only small errors to the input signal. Hence, engineers continue to talk about linearity of a certain system neglecting the non-linearities introduced by quantizers.

We can verify that even complex systems as image warping and simulcam are linear, when pixel values are assumed to be in $\mathbb{R}$. We need just to define the addition of two images (and two videos) and the multiplication by a constant. The result follows very easily from the definition of linearity.

1.1.4 Exercises

1. Give examples of physical phenomenons that we come across in everyday life and for which you can find a representation in the form of signals. Specify the domains and the codomains of the signals? What are their dimensions?

2. Give examples of signals in the following spaces

   (a) $\mathbb{Z} \to \mathbb{R}$

   (b) $\mathbb{R} \to \mathbb{R}^2$

   (c) $\{0, 1, \ldots, 600\} \times \{0, 1, \ldots, 600\} \to \{0, 1, \ldots, 255\}$

   (d) Give a practical application for the last space. What does a signal on this space represent?

3. Sketch the following signals :

$$\text{Triangle}(t) = \begin{cases} 0 & \text{if} \quad |t| > 1 \\ 1 - |t| & \text{if} \quad |t| \leq 1 \end{cases}$$

$$\delta_{-1}(t) = \begin{cases} 0 & \text{if} \quad t < 0 \\ 1 & \text{if} \quad t \geq 0 \end{cases}$$

$$\delta_{-2}(t) = \int_{-\infty}^{t} \delta_{-1}(\tau)\, d\tau$$

$$\text{Sum}(t) = \text{Triangle}(t) + \delta_{-1}(t)$$

$$\text{Diff}(t) = \text{Triangle}(t) - \delta_{-1}(t)$$

$$\text{Sinc}(t) = \begin{cases} 1 & \text{if} \quad t = 0 \\ \frac{\sin(\pi t)}{\pi t} & \text{if} \quad t \neq 0 \end{cases}$$

4. Specify the amplitude, frequency and phase of the signal:

$$x(t) = 5 \cos\left(10t + \frac{\pi}{2}\right) \quad ?$$

   What is the period of $x(t)$?

5. We know that a continuous-time sinusoid is a periodic signal. Is the sum of two sinusoids also periodic? Under which conditions? What is the period?

6. Sketch $x(t) = 5\cos(10t + \frac{\pi}{2}) + 2.5\sin(5t)$. Show that $x(t)$ is periodic. Which is the period?

7. We want to backup some images on a hard disk using as small a space as possible. The images are originally on memory. The size of all images is 768×1024 pixels. For each pixel, the color is represented in the memory using 24 bits. We know that on each image only 16 colors are used but we don't know which ones in advance. The 16 colors might be different for each image. How should we represent the information in order to minimize the space used on the disk? How many bits are needed to backup each image?

8. Give examples of systems in which the information is organized hierarchically. What are the signals used to represent information in the physical layer? What are the symbols used in the other layers? Are you acquainted with systems that elaborate the information on each layer?

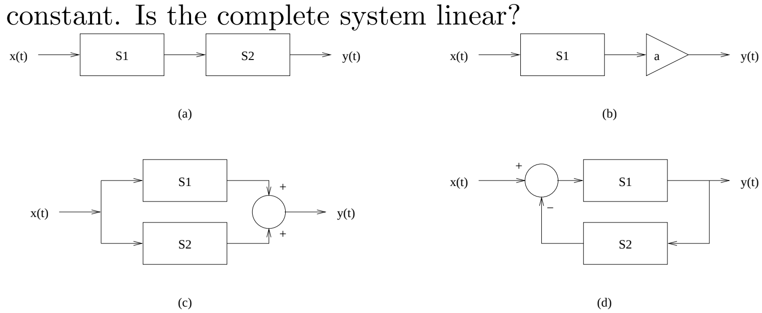
9. There is a big difference between the sets A, B and $S = \{s | s : A \to B\}$ (The set of signals from A to B). The following exercise explains this topic.

   (a) Suppose $A = \{x, y, z\}$ and $B = \{0, 1\}$. Make a list of all the functions from A to B (in other words the elements of S). Propose a procedure to list all such functions.

   (b) If A has m elements and B n, how many elements does S have?

   (c) Suppose that $A = \{0, \ldots, 287\} \times \{0, \ldots, 719\}$ and B is the set of color representable by 24 bits. Give an estimate of the number of elements of S in the form of 10^n, $n \in \mathbb{Z}$?

10. Suppose that the systems S_1 and S_2 are linear and that they are constructed for treating time continuous signals. Connecting the two systems as in the following picture, allows for constructing more complex systems. In the three pictures, the signal at the input of the complete system is called $x(t)$, the signal at th output is called $y(t)$ and α is a real constant. Is the complete system linear?



1.2 Filtering

1.2.1 Introduction

In this lecture we will see some special type of systems that we call **filters**. I am sure that you have an intuition of what a filter is. It is a device that allows eliminating something that we "don't like" while keeping something that we "like." Of course, what we like and what we don't like is relative to the application. You probably use a browser to access the internet. When you want to find the pages on a certain topic you use a search engine, choosing a list of keywords. We can think about the keywords as a description of the filter and the search engine as a procedure to apply it to the input data, i.e. the pages available on the internet.

Often, you don't want (or more likely you can't) eliminate the "disturbing" signal but you desire simply to reduce its level. For example, if you have a HiFi chain, you probably have some tone controls. When you set the controls to "off" all the frequencies are set at the same level by the system. That is, if you imagine that the input of the system is a pure sinusoid, you obtain the same level independently of the frequency of the sinusoid. If you change the tone controls, the sound will be different for certain frequencies. This is also a type of filter. There are many other examples of filters in nature. Actually, all the system that we can find in nature, i.e. all the phenomena for which you can define an input and an output, show a "filter" behavior. Most of them are "low-pass," that is, when you send a sinusoid at the input, you see that the amplitude decreases for high values of the frequency. You don't notice that with the HiFi system, but if you measure it in a laboratory, you would find the low-pass behavior. However, our ears are also low-pass so this is not a problem.

A numeric example: the moving average

Let us have a look at another example which concerns discrete-time signals. It is called the **moving average**. We consider a discrete-time signal that is affected by errors. Take for example the grades that you get at each exam or the number of goals you made at the last hockey match. In both cases you can think that the result is related to your actual effort and to a random perturbation that you cannot control (for example, the day of the exam you were sick or the teacher was not good, etc.). We can write

$$g(n) = s(n) + e(n),$$

i.e. the grade $g(n)$ you get at exam n is the result of your skills $s(n)$ plus an error term $e(n)$. Suppose that you would like to know your actual skills, for example to check if you are improving and at which rate. How would you do that? An idea is to compute the average at the end of each year. This is a solution, but you need to wait a whole year to have a new value and maybe take countermeasures (e.g. work harder). Also, your skills are changing over time, so an average on one year would hide such a change giving just a single Figure. A better solution is to recompute the average every exam, taking into account the L last exams. Here L is a certain value, for example $L = 8$. Why do we not take into account *all* the exams since the beginning of the studies? Because we want to be able to see the trend of our skills, i.e. the signal $s(n)$.

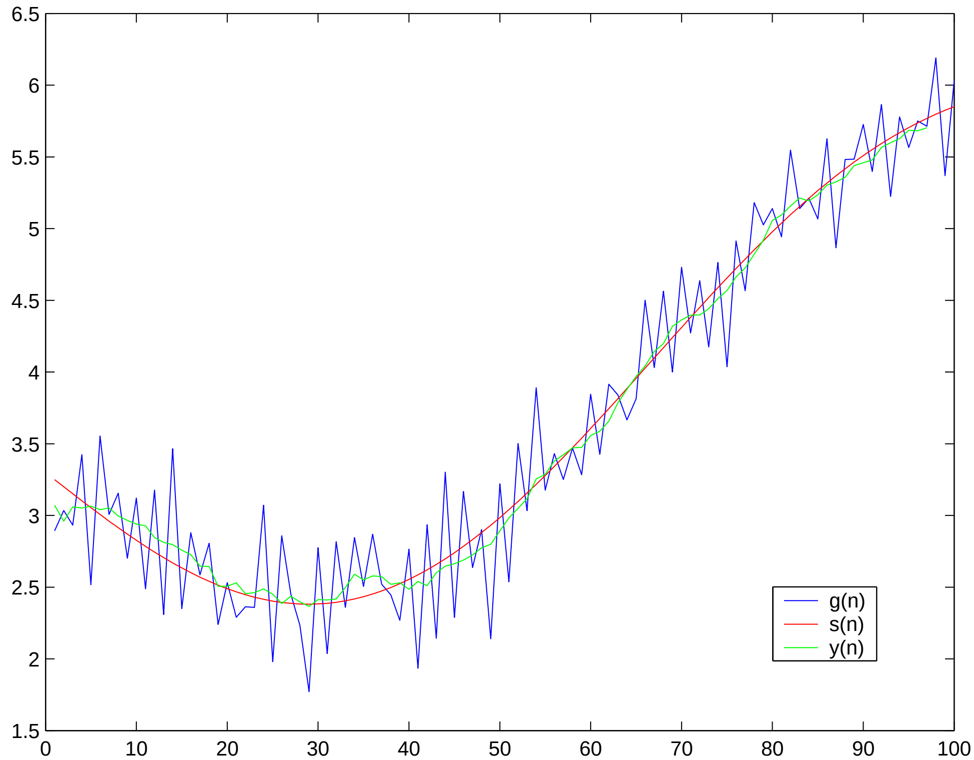


Figure 1.16: Example of moving average of the noisy signal $g(n)$. $s(n)$ is the original signal not affected by error. $y(n)$ is the filtered signal obtained with a moving average of length $L = 8$.

If we average too many values the result is less and less influenced by the last result. On the other hand, if L is too small the average is too much perturbed by the error terms that are only mildly attenuated. In conclusion, the length of the average L is a tradeoff between the attenuation of the errors and the speed of reaction of the system to the variations of the signal $s(n)$. In Figure 1.16 an example of filtering using the moving average is shown. You note that the measures $g(n)$ are very irregular because of the errors. The filtered signal $y(n)$ is obtained using a moving average of length $L = 8$. We see that $y(n)$ is quite close to the error-free signal $s(n)$ showing that the method is effective.

Some general properties of filters

We have seen three examples of filters. The first operated on symbols (the web pages) the second on continuous-time signals (the audio signals) and the third on discrete-time signals (a sequence of numeric values). Can we find some common properties to these filters? The first thing that we note is that the "scheme" that we apply to compute the result remains the same over time. For example, the search engine will propose the same web pages if the available pages remain the same. In other words, the filters do not age or learn from the past. We call these property **time-invariance**. Note that we can imagine more complex systems that are not time-invariant.

For example, the search engine can remember which pages we accessed in the past to propose better matches for the next searches.

The second property that these filters satisfy is so apparent that you probably do not notice it. It is called **causality** and it means basically that you cannot obtain an output of the filter before you apply an input. For example, you cannot know the moving average of your grades in the fourth year now that you are in the first year! It seems trivial, we simply say that we cannot predict the future.

In the next sections, we will consider only system working on signals and in particular discrete-time linear systems. We will also see more formally the properties of time-invariance and causality for these systems.

1.2.2 Impulse function. Impulse response

Let's define a signal that will be useful in the following. It is called **impulse** or **Kronecker delta function**. We define it in discrete-time but the concept can be defined in continuous-time as well.

$$\delta(n) = \begin{cases} 1 & \text{if} \quad n = 0 \\ 0 & \text{if} \quad n \neq 0 \end{cases} \quad \forall n \in \mathbb{Z}.$$

As you can see, the impulse is a very simple signal. Now, we want to use it to analyze the behavior of a discrete-time linear system. Suppose we send the impulse to the input of the system and that we measure the output of the system. The output, $h(n)$ is a discrete-time signal that we call **impulse response**.

At this point, I have to precise that what we have done is correct mathematically but it is unfeasible in practice. In fact, suppose that someone gives us a "black box" with an input and an output and we want to measure the impulse response. We would like to send an impulse to the input. However, the impulse is defined on the whole $\mathbb{Z}$ axis and it is zero for negative values. That means that, wherever we set the origin of the time coordinate, we have to guarantee that the black box received only zeros at its input before we apply the impulse! This is a common problem that engineers encounter in their work. We make some assumptions about the reality and that allows us to describe the problem mathematically. At the end, there may be some differences between what is predicted on the model and what we measure on a real system.

1.2.3 Time invariance

What happens if we shift the impulse along the time axis? A delayed impulse is represented by $\delta(n-m)$, where m is the delay and corresponds to the position of the "1" of the impulse. Suppose that we send this delayed impulse to a linear system, what do we measure at the output? We can call the signal at the output $\bar{h}(n,m)$, i.e. a generic function of two integer variables. Of course, when $m = 0$ the impulse is positioned in 0 and we obtain the impulse response defined in the previous section, i.e. $\bar{h}(n,0) = h(n)$. What happens for other values of the delay? One could think that the output is delayed by the same amount as the input. In other words, if you shift the input signal, the output signal is shifted in the same way. This is the property of time-invariance that we mentioned earlier. We can now give the following definition:

Definition 6. *A discrete-time linear system is* **time-invariant** *if the impulse response* $\bar{h}(n,m)$ *satisfies:*

$$\bar{h}(n,m) = h(n-m) \quad \forall n,m \in \mathbb{Z}.$$

Can we verify time-invariance for a certain physical system? As discussed in the previous paragraph, we cannot generate and measure signals on the complete real axis. We can only verify it for signals of finite duration and under appropriate hypotheses. Moreover, a system that is time-invariant in the short term could show time-variance in the longer term. For example, electronic components can age after some time. The same happens basically with all physical systems. However, many systems are time-invariant on a "reasonable" time scale. In particular, digital systems are extremely stable over time, at least until they break (a failure can also be considered as a form of time-variance). This is one of the main qualities that motivate the use of digital systems.

1.2.4 Definition of filter

We give the following definition of a filter.

Definition 7. *A filter is system which has the following properties:*

1. *It is linear.*
2. *It is time-invariant.*
3. *The domain of the input signal coincides with the domain of the output signal.*

Since the domains of the input and output signals are the same, we have only two types of one-dimensional filters: discrete-time and continuous-time. We can consider more complex signals and define filters on multidimensional signals, like images and video or vector signals like color images and color videos.

In this lecture we will discuss only discrete-time filters. In the following, we show that they are completely described by their impulse response $h(n)$.

1.2.5 Causality

Let's go back to the definition of the impulse response. We applied an impulse to the input of a linear system and we measured the output. We call the output $h(n)$ the impulse response. If we think of the negative part of the time axis $n < 0$, we see that the impulse is constantly zero. That means that we imagine to apply a series of zeros to the system starting infinitely far in the past. If the system that we are analyzing corresponds to a physical system, we can suppose that during this infinite amount of time it reaches an "equilibrium" state, i.e. the output is also zero[2]. Suppose we fix the output of the system to zero in correspondence of the equilibrium state (we just set the scale of the measurement device appropriately). At this point, is it possible to have something different from zero (the equilibrium value) in the region $n < 0$ of the impulse response? For example, if I measured $h(-10) = 1$ that would mean that something happens at the output, for $n = -10$, **before** I do something at the input! I know that the system was at an equilibrium condition, so I cannot explain the output with something that happened internally to the system and that is not related to the input. Therefore, I would conclude that the system is able to "predict" the future: it knows when I am going to send an impulse at the input and produces an output 10 samples in advance. It seems that, if we neglect time travels and clairvoyants, we have to exclude this possibility, at least for physical systems.

Definition 8. *A linear system is* **causal** *if the impulse response $h(n)$ satisfies:*

$$h(n) = 0 \ \ \forall n < 0.$$

Is causality a universal principle? When the domain of the signals is time, the answer is yes. However, at least formally you may have non-causality for systems that process non-temporal signals. For example, an image is a signal defined on two spatial coordinates. A system that treats images can access the whole domain of the input image, hence the impulse response can be non-causal. For example, suppose you have a camera and you take a picture of a small black spot on a white surface. You take the picture setting the focus to the wrong value, so the image appears to be unfocused. You can see the image as a system that takes the input image of the black spot and produces as a result the unfocused image. If the black spot is very small, we can consider it as an impulse function, so the output image is the corresponding impulse response. We observe that on the output image the effect of the impulse is propagated along all the directions and the result is a wide spot. Therefore, in the mathematical description of the system, we could use an impulse response which is non-causal.

1.2.6 Stability

In this section, we want to talk about "stability" and the relations between stability and the impulse response. What is stability? For sure, you have an intuition of what a stable system is. Basically, you say that a system is stable if the output does not grow too much when the input is limited. Certainly, when you compute the moving average of your grades, it would very strange to see that the result grows indefinitely if your grades are mediocre!

This type of stability is called **Bounded Input Bounded Output** (BIBO). Formally, the definition is:

Definition 9. *A filter with input $x(n)$ and output $y(n)$ is* **stable** *if*

$$\forall x \in \{s | s : \mathbb{Z} \rightarrow \mathbb{R}\}, \ \ |x(n)| \leq N, \forall n \in \mathbb{Z} \ \ \Rightarrow \ \ |y(n)| \leq M, \forall n \in \mathbb{Z}$$

for appropriate positive real constants N and M.

[2]it may not be the case for some particular system, as a pendulum with no friction

In other words, if the input signal remains in the range $(-N, N)$, i.e. is limited, the output signal is in the range $(-M, M)$. Note that we are free to choose the two constants N and M. For example, the system $y(n) = 10^6 x(n)$ is stable. You simply choose for example, $N = 1$ and $M = 10^6$. The fact that the system amplifies the input signal so much does not matter, since the output remains limited. Conversely, if you take the linear system with impulse response,

$$h(n) = e^n$$

you have a system that is unstable. In fact, if you send to the input an impulse, which is a bounded input, you obtain $h(n)$. You remember that the exponential grows indefinitely, when n increases, so you cannot find a bound for the output signal.

The definition of stability holds for any type of system, even non-linear systems. Here, we consider linear systems and we give a condition for stability based on the impulse response. From the last example, it is clear that the impulse response of a stable system cannot diverge. It can be proven that the stability implies a more restrictive condition on the impulse response. In fact, we have the following theorem.

Theorem 1. *A linear time-invariant system is stable if and only if the impulse response $h(n)$ is absolutely summable, i.e.*

$$\sum_{m=-\infty}^{\infty} |h(m)| < \infty.$$

For example, the impulse response

$$h(n) = \begin{cases} \rho^n & \text{if} \quad n \geq 0 \\ 0 & \text{if} \quad n < 0 \end{cases},$$

is stable if $|\rho| < 1$.

1.2.7 Convolution of signals

In this section, we see that a linear time-invariant system is completely specified by its impulse response, i.e. we can fully describe the relation between input and output signals.

The relation can be determined easily, decomposing the input signal $x(n)$ in a sum of shifted impulses. In fact, we have

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m). \tag{1.6}$$

We can verify this relation taking a particular value of $n = n_0$. All the impulses of the sum have the "1" at different positions. For $n = n_0$, only the impulse at position $m = n_0$ has value 1 and the term that multiplies the impulse is $x(n_0)$. This holds for any value of n_0, so the identity is verified.

Suppose that we send the signal $x(n)$ to the filter $\mathcal{H}$ with impulse response $h(n)$. How can we compute the output $y = \mathcal{H}(x)$? We know that the filter is a linear system, i.e. the output to a *finite* sum of signals is the sum of the outputs to each signal. If we add the condition that the filter is stable, the filter $\mathcal{H}$ is a continuous function on the space of the input signals[3]. In other words, we can apply the superposition principle even for infinite convergent sums of the type of (1.6).

We know that at each shifted impulse $\delta(n-m)$ the output is $h(n-m)$ for the time-invariance. Therefore, the output is simply the sum of the outputs to every impulse (see Figure 1.17)

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m).$$

We call this sum the **convolution** between the input signal and the impulse response of the filter. We write it using the notation $y(n) = (x * h)(n)$.

Let us verify some properties of the convolution. First of all **commutativity**, i.e.

$$(x * h)(n) = (h * x)(n).$$

In fact, if one defines $m_0 = n - m$ and we eliminate m in the sum we have

$$(x * h)(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = \sum_{m_0=-\infty}^{\infty} x(n-m_0)h(m_0) = (h * x)(n).$$

This means that the result is the same if we swap the input signal with the impulse response, i.e. a filter with impulse response x and input h would give the same result.

Convolution is **linear**, since it is the input-output relation of a linear system. That means that

$$(u_1 x_1 + u_2 x_2) * h = u_1(x_1 * h) + u_2(x_2 * h).$$

The property of **associativity** allows grouping arbitrarily a chain of convolutions:

$$(x * h_1) * h_2 = x * (h_1 * h_2)$$

That means that we can replace a cascade of two filters h_1, h_2 with a single filter $h_1 * h_2$. Taking into account commutativity, we notice also that in a chain of filters the result does not depend on the order of the filters.

Example of convolution

Normally, we use computers to compute the convolution of signals. However, it is helpful to learn how to compute manually a convolution to fully understand how it works. We consider the simple example, depicted in Figure 1.18. The result is obtained taking $h(n)$ and mirroring it with respect to the origin, i.e. we obtain $h(-n)$. At this point, we have to shift $h(-n)$ along the time axis. For each position m, the shift gives us $h(m-n)$ which are the weights of the values $x(m)$. We compute all the products $h(m-n)x(m)$ and we sum to obtain the result $y(n)$.

[3]The proof is simple but it would need some concepts about metric spaces that you will study in second year

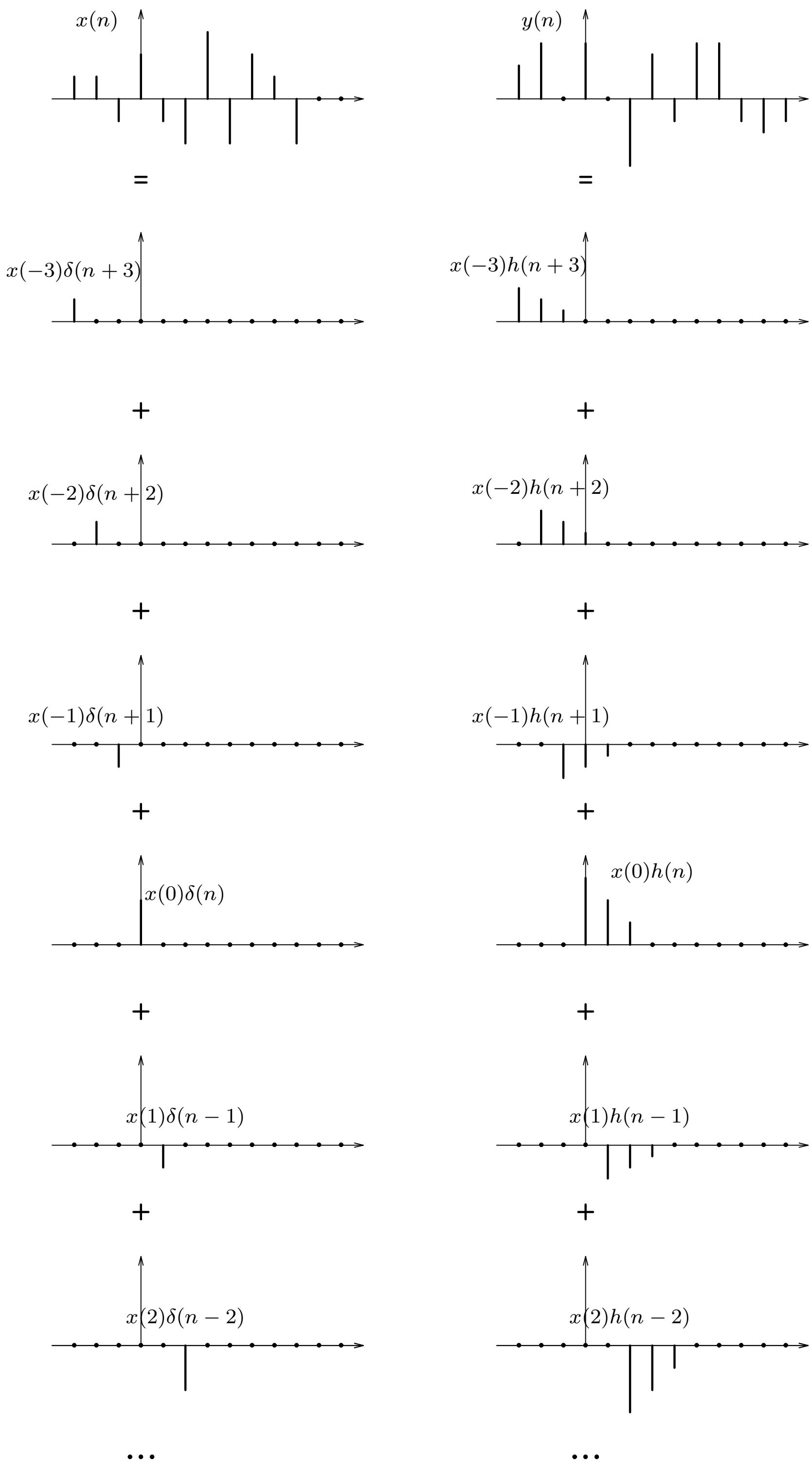


Figure 1.17: Input-output relation of a filter. On the left column, the input signal $x(n)$ is decomposed in a sum of weighted impulses. The output $y(n)$ is obtained by summing the impulse response after shift and weighting corresponding to each impulse at the input.

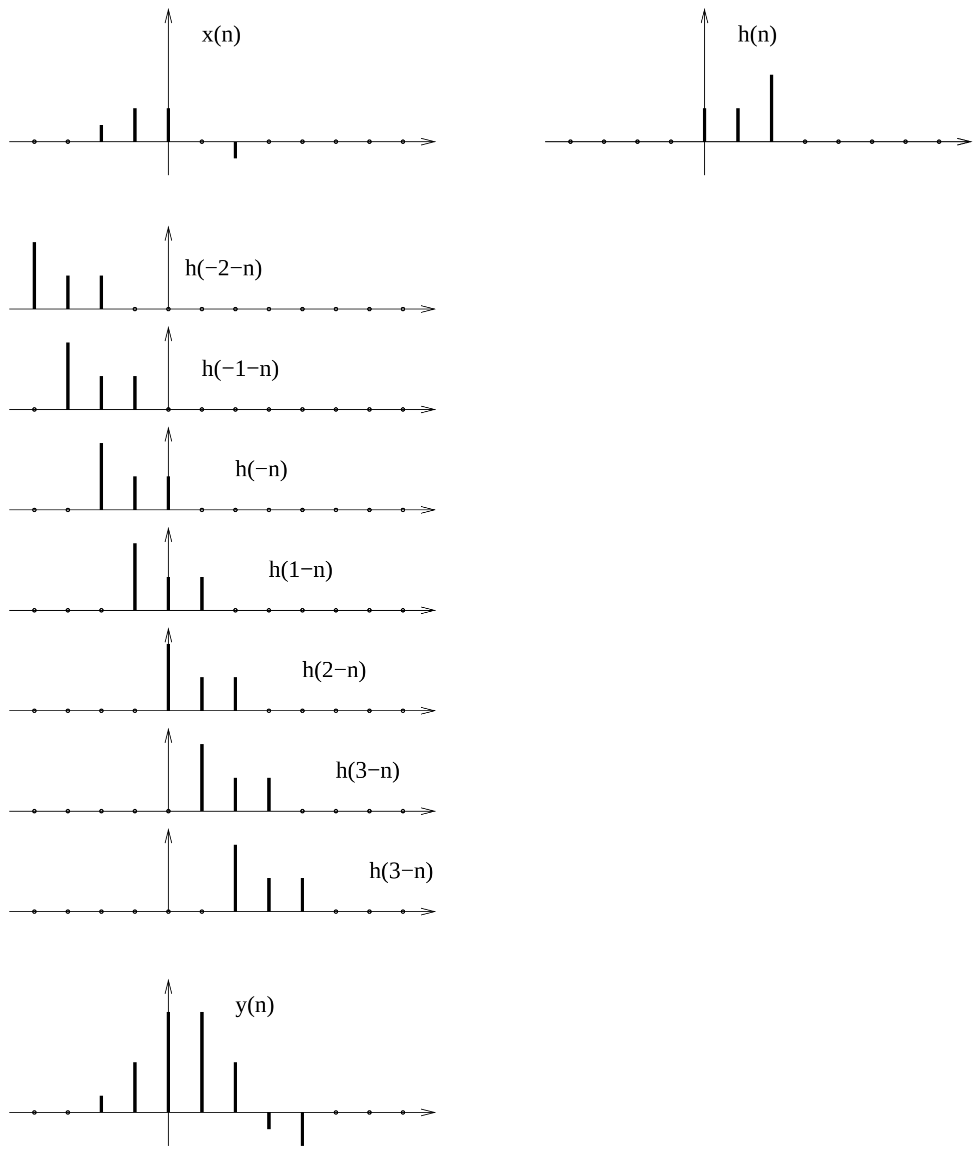


Figure 1.18: Convolution of the signal x with the filter impulse response h. The result is computed considering all the shifts of the signal $h(-n)$. For every position n the corresponding output is computed by summing the products $h(n-m)x(m)$.

Convolution of a sinusoid with a signal

Now that we know convolution, we can compute the output of a certain filter for different types of input signals. According to the filter impulse response and the input signal, we may note that in some cases the output is relatively similar to the input. It is the case for the example of Figure 1.18. The amplitude of the signal has changed and there is a translation along the time axis, but the shape of the output signal is similar to that of the input signal. Are there signals that keep exactly the same shape when they pass through a filter? The answer is yes, and the signals are the sinusoids! Let us take $x(n) = \sin(\omega_d n)$ and compute the convolution with the impulse response $h(n)$. A good method to do that is to use a complex exponential. Remember that

$$e^{j\alpha} = \cos(\alpha) + j\sin(\alpha).$$

Therefore the input signal can be written as

$$x(n) = \mathrm{Im}(e^{j\omega_d n}),$$

where "Im" means imaginary component. The advantage of using a complex exponential is that we avoid to use difficult trigonometric formulas. We just have to remember to take the imaginary part to compute the result. In fact,

$$y(n) = \sum_{m=-\infty}^{\infty} sin(\omega_d(n-m))h(m) = \mathrm{Im}\left(\sum_{m=-\infty}^{\infty} e^{j\omega_d(n-m)}h(m)\right).$$

Now we can decompose the exponential in two factors:

$$y(n) = \mathrm{Im}\left(e^{j\omega_d n}\sum_{m=-\infty}^{\infty} e^{-j\omega_d m}h(m)\right).$$

The term $e^{j\omega_d n}$ does not depend on m, therefore it has been moved outside of the sum. We remark that the sum is not a function of the time n, i.e. it is a complex value which depends only on the sinusoid frequency ω_d:

$$P(\omega_d)e^{j\phi(\omega_d)} = \sum_{m=-\infty}^{\infty} e^{-j\omega_d m}h(m).$$

You see that we represented the complex value in polar representation: $P(\omega_d)$ is the magnitude and $\phi(\omega_d)$ the argument. We can write the output of the filter as,

$$y(n) = \mathrm{Im}(P(\omega_d)e^{j\omega_d n+\phi(\omega_d)}) = P(\omega_d)\sin(\omega_d n + \phi(\omega_d)).$$

Therefore, the output is a sinusoid with amplitude $P(\omega_d)$ and phase $\phi(\omega_d)$. Note that the amplitude and the phase are function of the frequency ω_d, i.e. if you change the frequency of the sinusoid, the amplitude may also change.

Why are complex exponentials (or sinusoids) so special? See how we compute a convolution:

$$(x * h)(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m)$$

i.e. the output signal is obtained combining shifted versions of the input signal. For the complex exponentials, when you take different shifts and you sum, you still obtain a complex exponential. In fact, if

$$x(n) = e^{j\omega_d n},$$

the shifted signal $x(n-m)$ can be written as

$$x(n-m) = e^{j\omega_d (n-m)} = e^{j\omega_d n} e^{-j\omega_d m} = x(n) e^{-j\omega_d m},$$

i.e. the result is the input signal multiplied by a number that is function of the shift and of the frequency. This is not a general property of functions.

1.2.8 Finite impulse response (FIR) filters

In this section we consider some particular linear time invariant filters, for which the impulse response has a finite duration. That is,

$$h(n) = 0 \qquad \text{if} \quad n < 0 \quad \text{or} \quad n \geq L,$$

where L is some positive integer. Such a system is called a **finite impulse response (FIR)** filter because the "interesting part" of the impulse response has finite duration. Because of that property, the convolution sum becomes a finite sum:

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m) = \sum_{m=0}^{L-1} x(n-m)h(m).$$

This equation suggests a way to easily implement the filter on a computer. We note that to produce the output at time n, we need the input signal at time n, $n-1$, ..., $n-L+1$. These values will be stored in the computer memory. The impulse response is a series of coefficients that we can also store in the memory. A program to compute the output simply takes the values of the input memory and multiplies them by the impulse response coefficient. The result is obtained by summing all the products. When a new value is available at the input, we discard the oldest value that we saved in memory and we shift the others to insert the new one. The output is computed applying the same scheme.

An important remark concerning FIR filters is that they are *always stable*, independently of the coefficients of the impulse response. This is a direct consequence of theorem 1.

Example: the moving average

Let us consider again the moving average that we saw at the beginning of the lecture. We said that the moving average of the grades is obtained by computing the average of the most recent L grades. In formulas, we can write

$$y(n) = \frac{1}{L} \sum_{m=0}^{L-1} x(n-m) \quad \forall n \in \mathbb{Z}.$$

This is exactly a FIR filter with impulse response:

$$h(n) = \begin{cases} \frac{1}{L} & \text{if} \qquad 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

We mention that the choice of L is the result of a compromise between the need of filtering the errors $e(n)$ while keeping the variations of the skills $s(n)$. Let us see how this happen. Suppose that the two signals $s(n)$ and $e(n)$ are available. Of course, we can do that only with simulated data. We know that the filter is linear, so filtering $g(n) = s(n)+e(n)$ is the same as filtering $s(n)$ and $e(n)$ separately and then summing the results. Therefore, we can understand the behavior of the filter, considering the two signals separately. What happens when we filter these signals with filters of different length? The results are depicted in Figure 1.19. We notice how both $e(n)$ and $s(n)$ become more and more flat as L increases. If you imagine that L goes to infinity (we suppose here that you have enough grades to compute such long averages) the result of the moving average of the error signal will go to zero. In fact, we supposed that the errors are "fair", i.e. they increase or decrease your grade with the same probability. For the signal $s(n)$, when L goes to infinity, we smooth the variations of the skills and the result converges to the average of the whole set of measurements. What changes between the filtering of the two signals, is the rate at which the results are smoothed with respect to L. For example, take $L = 8$. You see how the error signal is already much attenuated, while the signal $s(n)$ is still very similar to the original. We can explain this by noting that the error signal is very irregular, while $s(n)$ is smooth. In other words, the parameter L controls the speed of variation of the signals that pass through the filter. In practice, one would take some hypothesis on the signals $s(n)$ and $e(n)$ and would choose the optimal compromise for the parameter L. You can imagine that the best results are obtained when the useful signal and the error have a very different behavior.

We can reconsider the analysis of the moving average by considering a very simple signal, i.e. the sinusoid. The frequency of the sinusoid represents the speed of variation that we mentioned before (the higher the frequency, the steeper the signal). We know that the output of the filter to a sinusoid is also a sinusoid. Therefore we can study the attenuation of the filter by analyzing the amplitude of the output sinusoid as a function of the frequency. As we saw in the previous section, if the input signal $x(n) = \sin(\omega_d n)$, the output is

$$y(n) = \text{Im}(P(\omega_d)e^{j\omega_d n+\phi(\omega_d)}) = P(\omega_d)\sin(\omega_d n + \phi(\omega_d)),$$

where the amplitude and the phase are computed by

$$P(\omega_d)e^{j\phi(\omega_d)} = \sum_{m=-\infty}^{\infty} e^{-j\omega_d m} h(m).$$

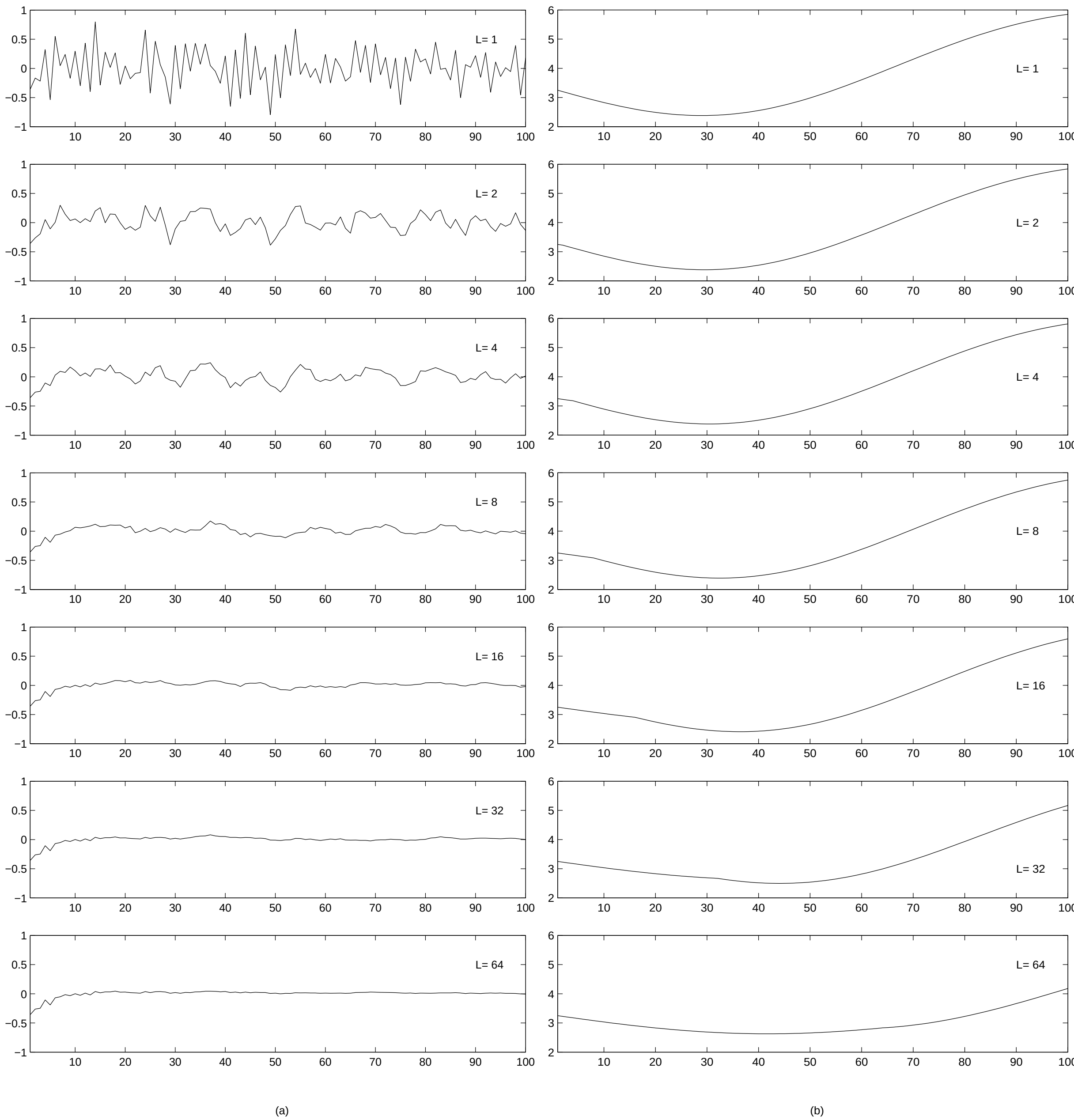


Figure 1.19: Moving average of the error signal and the error-free signal. (a) Output of the moving average to the error signal using different values of the filter size L. (b) Output of the error-free signal for the same values of L. Note how both the error and the error-free signal are smoothed when L increases. The optimal L is the value that gives the best trade off between error attenuation and non distortion of the useful signal.

We substitute $h(m)$ with the impulse response of the moving average and we obtain,

$$P(\omega_d)e^{j\phi(\omega_d)} = \frac{1}{L}\sum_{m=0}^{L-1} e^{-j\omega_d m}.$$

Remember that the sum of a geometric sequence is given by

$$\sum_{m=0}^{L-1} q^m = \frac{1-q^L}{1-q}.$$

Therefore,

$$P(\omega_d)e^{j\phi(\omega_d)} = \frac{1}{L}\frac{1-e^{-j\omega_d L}}{1-e^{-j\omega_d}}.$$

If we take into account that

$$\sin\alpha = \frac{e^{j\alpha}-e^{-j\alpha}}{2j}$$

we can continue the derivation, obtaining

$$P(\omega_d)e^{j\phi(\omega_d)} = \frac{1}{L}\frac{e^{-j\omega_d\frac{L}{2}}}{e^{-j\omega_d\frac{1}{2}}}\frac{\sin(\frac{\omega_d L}{2})}{\sin(\frac{\omega_d}{2})} = e^{-j\omega_d\frac{L-1}{2}}\frac{\sin(\frac{\omega_d L}{2})}{L\sin(\frac{\omega_d}{2})}.$$

In conclusion,

$$P(\omega_d) = \left|\frac{\sin(\frac{\omega_d L}{2})}{L\sin(\frac{\omega_d}{2})}\right|.$$

We add the absolute value, because we can take into account the sign in computing the phase ϕ (we add π to the phase when $P(\omega_d) < 0$).

In Figure 1.20, $P(\omega_d)$ is shown for different values of L. As expected, we notice how amplitude decreases for higher frequencies, so the filter is actually a "low-pass". We also see that the parameter L controls the attenuation of high frequencies.

Design of FIR filters

We have seen that a filter is completely described by its impulse response. When you change the length and the coefficients of a FIR filter you obtain different performances. We saw that in the previous paragraph considering the parameter L of the moving average. The parameter was chosen according to the type of evolution of the useful signal and the error signal. In the same way, we can consider to change every parameter of the impulse response. An analogy in continuous-time is the equalizer of a HiFi chain. When you turn the knobs, you change the behavior of the filter. In the same way, there are tools to design FIR filters. The user imposes some constraints to the filter. Normally these consist of the level of attenuation of sinusoids at different frequencies (what is called the **frequency response**). The software finds the impulse response that best matches the constraints.

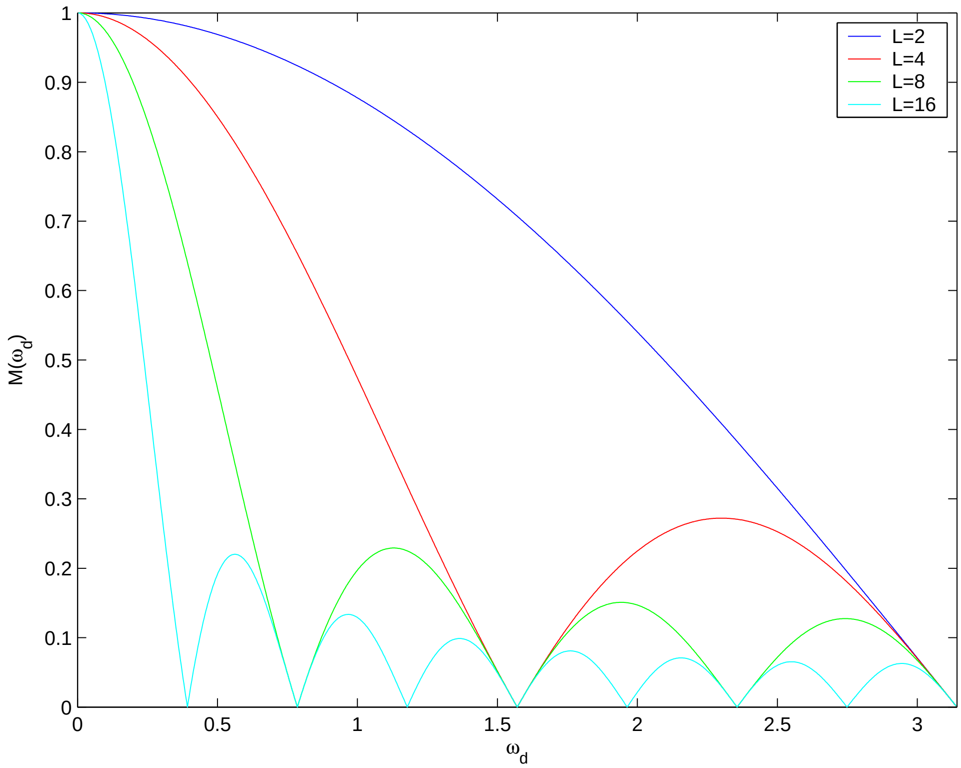


Figure 1.20: Amplitude of a sinusoid at the output of a moving average filter as a function of the frequency. The filter shows a "low-pass" behavior, i.e. sinusoids with high frequency are strongly attenuated. The filter size L controls the attenuation and the range of frequencies that can reach the output.

1.2.9 Infinite impulse response (IIR) filters

Let us consider again the convolution sum,

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m).$$

We have seen that when the impulse response is FIR, we can compute easily the output of the filter. When the impulse response is not finite, i.e. there is no finite number of coefficients different from zero, we say that the filter is an infinite impulse response filter (IIR). One could think that for IIR filters, it is not possible to compute the sum since it involves an addition of an infinite number of terms. Actually, it is true that in general for an *arbitrary* IIR, it is not possible to compute the sum. However, we see that there are some very special responses for which we can. We show that with an example.

Let us consider the equation

$$y(n) = \rho y(n-1) + (1-\rho)x(n).$$

We note that the output $y(n)$ is computed combining two terms: the first is related to the output itself at the *previous* step, i.e $y(n-1)$, the second to the current input, $x(n)$. The factors ρ and $1-\rho$ allow us to set the proportion of the two contributions. We choose $0 < \rho < 1$. For example, we can take $\rho = 0.5$. Let us see how it works when you apply it to average your grades. After the first exam, you have the first grade $x(0)$ (we number the exams starting with zero). Since we just started, $y(-1)$ is not defined. We impose $y(-1) = x(0)$. Applying the equation, we obtain

$$y(0) = 0.5x(0) + (1-0.5)x(0) = x(0),$$

which is correct: the first average is the first grade. After the second exam, you have x(1). When we reapply the rule, we get

$$y(1) = 0.5y(0) + 0.5x(1) = 0.5x(0) + 0.5x(1),$$

which is the average of the first two grades. At the third grade, we have

$$y(2) = 0.5y(1) + 0.5x(2) = 0.25x(0) + 0.25x(1) + 0.5x(2).$$

That is the first unusual average: the first two grades are multiplied by the factor 0.25 while the last one by the factor 0.5. If we continue the iterations we have

$$\begin{array}{l} y(3) = 0.125x(0) + 0.125x(1) + 0.25x(2) + 0.5x(3) \\ y(4) = 0.0625x(0) + 0.0625x(1) + 0.125x(2) + 0.25x(3) + 0.5x(4) \\ y(5) = 0.03125x(0) + 0.03125x(1) + 0.0625x(2) + 0.125x(3) + 0.25x(4) + 0.5x(5) \\ ... \end{array} .$$

See how the oldest grades are multiplied by factors which are smaller and smaller but never zero. Grades that are more recent are multiplied by increasing factors. As a comparison, the moving average was not taking into account the oldest grades and multiplying by the same factor $1/L$ the most recent ones. In other words, we compute an average where we take into account all the grades, but with different weights. Therefore, we can consider this as an alternative to the moving average.

We show that what we obtain is an IIR filter. In fact, if $x(n) = \delta(n)$ you obtain the impulse response $1, 0.5, 0.25, 0.125, \ldots$. If we consider the generic parameter ρ, we have

$$h(n) = \begin{cases} \rho^n & \text{if} \quad n \geq 0 \\ 0 & \text{if} \quad n < 0 \end{cases}$$

which is actually IIR. Note that we must choose $|\rho| < 1$ in order to have a stable impulse response, that is *an IIR filter may be unstable.*

This is just an example of an IIR filter. You can find many others. The common principle is to express the output as a function of input and output at previous times.

1.2.10 Exercises

1. Answer the following questions:

   (a) Can a finite impulse response filter (FIR) be unstable?

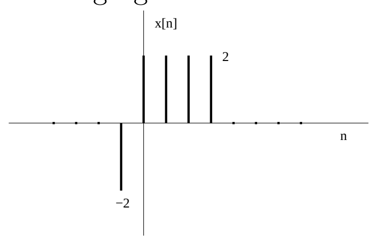
   (b) Consider a system that makes predictions, as an example consider the temperature of a city given by the weather forecast. Is this system causal?

   (c) In this chapter we saw that, when we send a sinusoid as an input of a filter, we will also have a sinusoid with the same frequency at the output. this property is a consequence of the fact that a complex exponential is the solution of

   $$x(n-m) = A(m)x(n) \;\; \forall n \in \mathbb{Z},$$

   for a given $A(m) \in \mathbb{C}$. Can you find other functions that satisfy this equation?

2. The signal $x(n)$ is shown in the following figure:



   Sketch exactly the following signals:

   (a) $x(n-2)$

   (b) $x(3-n)$

   (c) $x(n-1)\delta(n)$

   (d) $x(1-n)\delta(n-2)$

3. Consider a filter that has an impulse response given by h

   $$h(n) = \delta(n) + 2\delta(n-1).$$

   (a) Sketch the impulse response.

   (b) Calculate and sketch the output signal when the input signal is

   $$u(n) = \begin{cases} 1 & \text{if} \;\; n \geq 0 \\ 0 & \text{if} \;\; n < 0 \end{cases}.$$

   (c) Calculate and sketch the output signal when the input signal is

   $$r(n) = \begin{cases} n & \text{if} \;\; n \geq 0 \\ 0 & \text{if} \;\; n < 0 \end{cases}.$$

(d) Calculate the output signal when the input signal is

$$x(n) = \cos(\pi n/2 + \pi/6) + \sin(\pi n + \pi/3).$$

4. Calculate the output of a filter with impulse response:

$$\begin{aligned} &h(0) = 2, \ \ h(1) = 1, \ \ h(2) = -1, \\ &h(n) = 0 \ \ n < 0 \ \text{ or } \ n > 2, \end{aligned}$$

when the input signal is,

$$\begin{aligned} &x(0) = 1, \ \ x(1) = 2, \ \ x(2) = 3, \\ &x(n) = 0 \ \ n < 0 \ \text{ or } \ n > 2. \end{aligned}$$

5. Consider a filter with impulse response

$$\begin{aligned} &h(0) = 1, \quad h(1) = -1, \\ &h(n) = 0 \quad n < 0 \ \text{ or } \ n > 1. \end{aligned}$$

With the help of graphical interpretation of the convolution, calculate the output signal of the filter $y(n)$ when the input signal $x(n)$ is

$$x(n) = \begin{cases} 1 & 1 \leq n \leq 4 \\ 0 & \text{otherwise.} \end{cases} .$$

6. Consider a filter with impulse response

$$h(n) = \begin{cases} 0.8^n & n \geq 0 \\ 0 & n < 0. \end{cases} .$$

Sketch $h(n)$. Is the filter causal? Is it stable and time invariant? Is it an FIR filter?

7. Two filters H_1, H_2 have the following impulse response:

$$h_1(n) = \begin{cases} \frac{1}{n} & 0 < n < 4 \\ 0 & \text{otherwise} \end{cases} , \ \ h_2(n) = \begin{cases} n & 0 < n < 4 \\ 0 & \text{otherwise} \end{cases} .$$

Calculate the impulse response obtained by cascading H_1 and H_2. Is this system also a filter? Is the impulse response finite (FIR)? What happens if we swap H_1 and H_2?

8. Consider a filter with the impulse response

$$h(n) = \frac{1}{L} \sum_{m=0}^{L-1} \delta(n - m).$$

What operation does this filter perform? Is the filter time invariant?Is it causal? Suppose that the signal at the input of the filter is $x(n) = sin(2\pi n/5)$, sketch a few samples of the signal at the output when $L = 3$. Can you say what happens when L increases?

9. Suppose that $x(n)$ and $y(n)$ are the input and the output of a numerical system. Determine if the following systems are linear, stable, time-invariant, or causal:

   (a) $y(n) = 3x(n) - 4x(n-1)$

   (b) $y(n) = 2y(n-1) + x(n+2)$

   (c) $y(n) = nx(n)$

   (d) $y(n) = \cos(x(n))$

10. Imagine you are in a band of amateur musicians and you are responsible for recording using your computer. Because of the limited budget, you cannot afford high quality equipment and noise is constantly present in your recordings. You find out that this noise $\eta(t)$ is actually a sinusoid at 100 Hz that comes from the power supply network. The recorded signal, $s(t)$ is $s(t) = m(t) + \eta(t)$, where $m(t)$ is the desired signal, taken from the microphone. Your computer samples the signal $s(t)$ at $f_s = 8000$ Hz (In other words the sampling period is $T_s = 0.125$ s). You decide to use the techniques that you have learned during the course to filter $s(t)$. First, you apply the moving average of length L. How do you choose the parameter L in order to completely eliminate the component $\eta(t)$? You notice that there are several values of L that make it possible to eliminate $\eta(t)$. What are the effect on the component $m(t)$ if you use a relatively large value for L? A friend of you from the third year proposes you to use a simpler filter which is as follows

$$y(n) = s(n) + a_1 s(n-1) + a_2 s(n-2).$$

    What are the values of a_1 and a_2 if we want to completely eliminate the component $\eta(t)$ at the output of the filter?

11. Consider the averaging example of the grades of an exam that we studied in the course. Suppose that at the end of the year your grades are:

$g(0)$	$g(1)$	$g(2)$	$g(3)$	$g(4)$	$g(5)$
3	4.5	4.2	5.1	5	5.7

    Suppose $y(n)$ is the signal obtained applying the moving average with length $L = 4$.

   (a) What are the values of $y(3)$, $y(4)$ and $y(5)$?

   (b) Suppose that we decompose $g(n)$ in the following way:

$$g(n) = s(n) + e(n)$$

   where $s(n)$ is the note that you deserve and $e(n)$ is an error term, for example due to unfair exams. Can you find an example of a signal $e(n)$ (except for $e(n) = 0$)) so that $y(n)$ is exactly equal to the moving avarage of $s(n)$? In general, what properties should $e(n)$ satisfy?

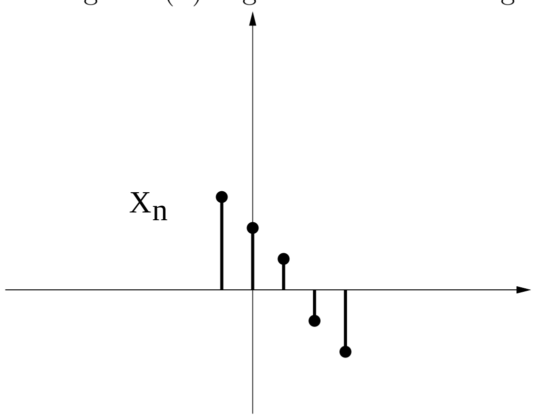
12. Consider a linear system with the impulse response $h(n) = 2\delta(n) - \delta(n-1) - \delta(n-2)$.

   (a) Is the system causal? Why?

   (b) Is the system stable? Why?

(c) What is the output signal of the system when the received signal at the input is: $x(n) = \delta(n) + \delta(n-1) - \delta(n-2) + \delta(n-3)$?

13. The signal $x(n)$ is given in the following figure:



Sketch the following signals

(a) $x(n-3)$

(b) $x(2-n)$

(c) $x(n-1)\delta(n)$

(d) $x(n+1)\delta(n-2)$

14. Consider un filter having the impulse response

$$h(n) = \delta(n) - 2\delta(n-1) + \delta(n-2)$$

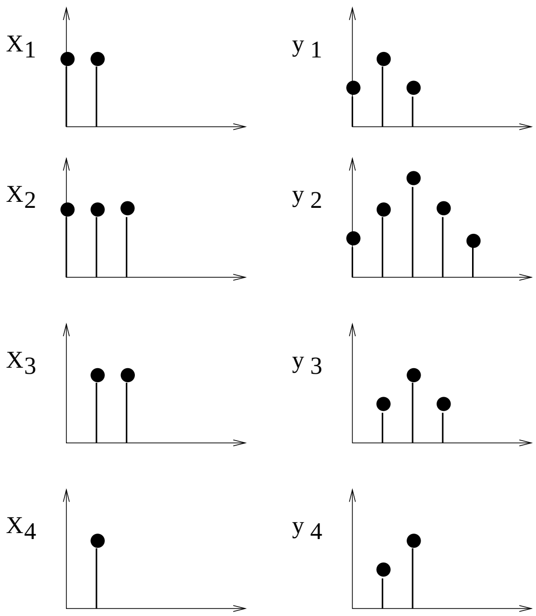
(a) Is this filter causal?

(b) Is this filter stable?

(c) Compute the output if the signal at the input is

$$x(n) = \delta(n-1) - 3\delta(n-2)$$

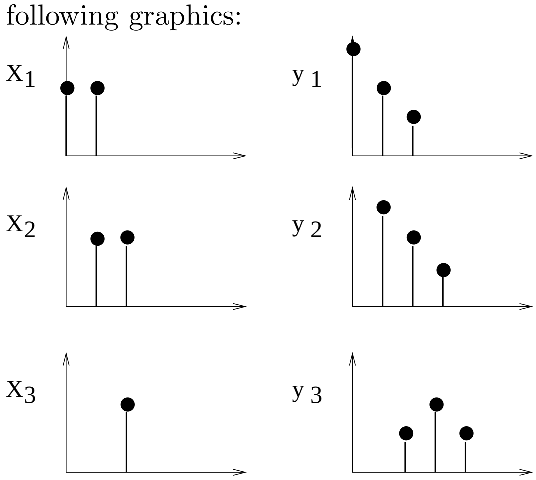
15. For verifying the linearity of a system, we use four signals $x_1(n), x_2(n), x_3(n)$, et $x_4(n)$. The corresponding signals at the output are $y_1(n), y_2(n), y_3(n)$ et $y_4(n)$. The signals at the input and at the output are given by the following pictures:



(a) Are you able to decide wether this system is linear? Explain you answer.

(b) In general, is an finite number of signals is enough to decide if a system is linear or not?

16. Consider a system S and imagine to use three test signals for detecting his components. $x_1(n), x_2(n)$ and $x_4 n$ are the three signals at the input and $y_1(n), y_2(n)$, et $y_3(n)$ are the corresponding output signals. The signals at the input and at the output are given by the following graphics:



Lets call $\bar{h}(n,m)$ the response to the impulse $\delta(n-m)$.

(a) Suppose that S is linear and compute $\bar{h}(n,0)$, which equivalent to the response to $\delta(n)$.

(b) Are you able to decide wether the system is time invariant? Explain your answer!

(c) Are you able to decide wether the system is stable? Explain your answer.

17. We have seen in the cours that the mobile average is an efficient method for reducing errors in a sequence as grades of exams. Lets call $g(n)$ the grade obtained in the n^{ieme} exam and $y(n)$ the corresponding mobile. All mobile averages considered in this exercise are of length $L = 4$.

(a) Suppose that the grades of one student in the end of his first year are given by

$g(0)$	$g(1)$	$g(2)$	$g(3)$	$g(4)$	$g(5)$
3	4.5	4.3	5.2	5	5.5

What are the values of $y(3)$, $y(4)$, $y(5)$?

(b) Suppose that the grades $g(n)$ can be decomposed in the following way:

$$g(n) = s(n) + e(n)$$

with $s(n)$ corresponding to the true value and $e(n)$ to an error component. Show that if

$$e(n) = e_1(n) = \sin\left(\frac{\pi}{2}n\right)$$

ou

$$e(n) = e_2(n) = \cos(\pi n),$$

the error component $e(n)$ is completely eliminated by the mobile average. (hint: Equivilantly, you can use the fact that $\sin(\alpha - \beta) = \sin\alpha\cos\beta - cos\alpha\sin\beta$)

(c) Use the signals $e_1(n)$ and $e_2(n)$ for computing other signals which can be elliminated by the mobile average (at least one!). Which is the property of the system, you use for constructing such a signal?

(d) Suppose that the error signal is exactly one of those we are able to eliminate using the mobile average. Are you able to reconstruct **exactly** $s(n)$, for $ngeq3$, based on the signal $g(n)$ using the mobile average? Justify your answer.

18. Suppose that a linear and time invariant system has the impulse reposponse

$$h(n) = \begin{cases} (-1)^n n & \text{if} \quad n \geq 0 \\ 0 & \text{if} \quad n < 0 \end{cases}$$

(a) Sketch the impulse response $h(n)$. Is the system causal?

(b) Is the system stable?

(c) Compute the output signal if the input signal is
$x(n) = \delta(n) + \delta(n-1)$

1.3 The discrete Fourier transform

1.3.1 Introduction

In section 1.2.7, we have found that sinusoids have a special property with respect to filters. If a sinusoid is sent to a filter, the output (i.e. the convolution with the impulse response) is also a sinusoid with the same frequency of the input and only amplitude and phase are changed. This is an interesting property of a filter that we want to explore better in this chapter. The main idea is to introduce a decomposition of the input signal as a sum of sinusoids. Since filters are linear, we can superpose the effects of each sinusoid at the output; hence, we can write directly the output signal without the need of a convolution. This fact was noticed for the first time by the French mathematician Joseph Fourier (1768-1830). Today, we call *Fourier transform* the decomposition of a signal in a sum of sinusoids.

We have seen that there are different types of signals. First of all, we have seen that signals can be continuous or discrete time. This induces two types of Fourier transform. In fact, we decompose our signal either by using a set of continuous time sinusoids or a set of discrete time sinusoids. Moreover, there is a second subdivision of signals according to periodic and aperiodic signals (either discrete or continuous time). This subdivision implies a difference structure of the Fourier transform for the two cases. In conclusion, there are four types of Fourier transforms, corresponding to the domain of the signal, continuous or discrete, and the periodicity/aperiodicity.

It may be surprising that aperiodic signals could be decomposed as a sum of of periodic signals. However, we recall that even the sum of only two sinusoids give an aperiodic signal when their frequencies do not have a rational ratio. It results that even aperiodic signals can be decomposed by extending the sum to an infinite number of sinusoids. This represent a powerful tool for the analysis of linear systems.

In this chapter, we want to examine the simplest case of Fourier transform. For that, we restrict our analysis to the case of *discrete time periodic signals*. For this case, the Fourier transform takes the name of *Discrete Fourier Transform* (DFT).

1.3.2 A simple example

We start our study with a simple example. Let's consider the signal $x(n)$ periodic with period $N = 4$, depicted in figure 1.21(a). The signal takes the values

$$x(0) = 4, \quad x(1) = 3, \quad x(2) = 2, \quad x(3) = 1.$$

Can we write it as a sum of sinusoids? Since the period is 4, we consider only the sinusoids with period 4 or a divisor of 4, that is 4, 2, 1. Therefore, we look for a decomposition of the form

$$x(n) = P_0 + P_1 \sin\left(2\pi\frac{n}{4} + \phi_1\right) + P_2 \sin\left(2\pi\frac{n}{2} + \phi_2\right), \tag{1.7}$$

where the parameters P_0, P_1, P_2, ϕ_1, and ϕ_2 are the unknowns. There are only 4 constraints, given by the values at the points $n = 0, 1, 2, 3$; hence, we should be able to compute the unknowns.

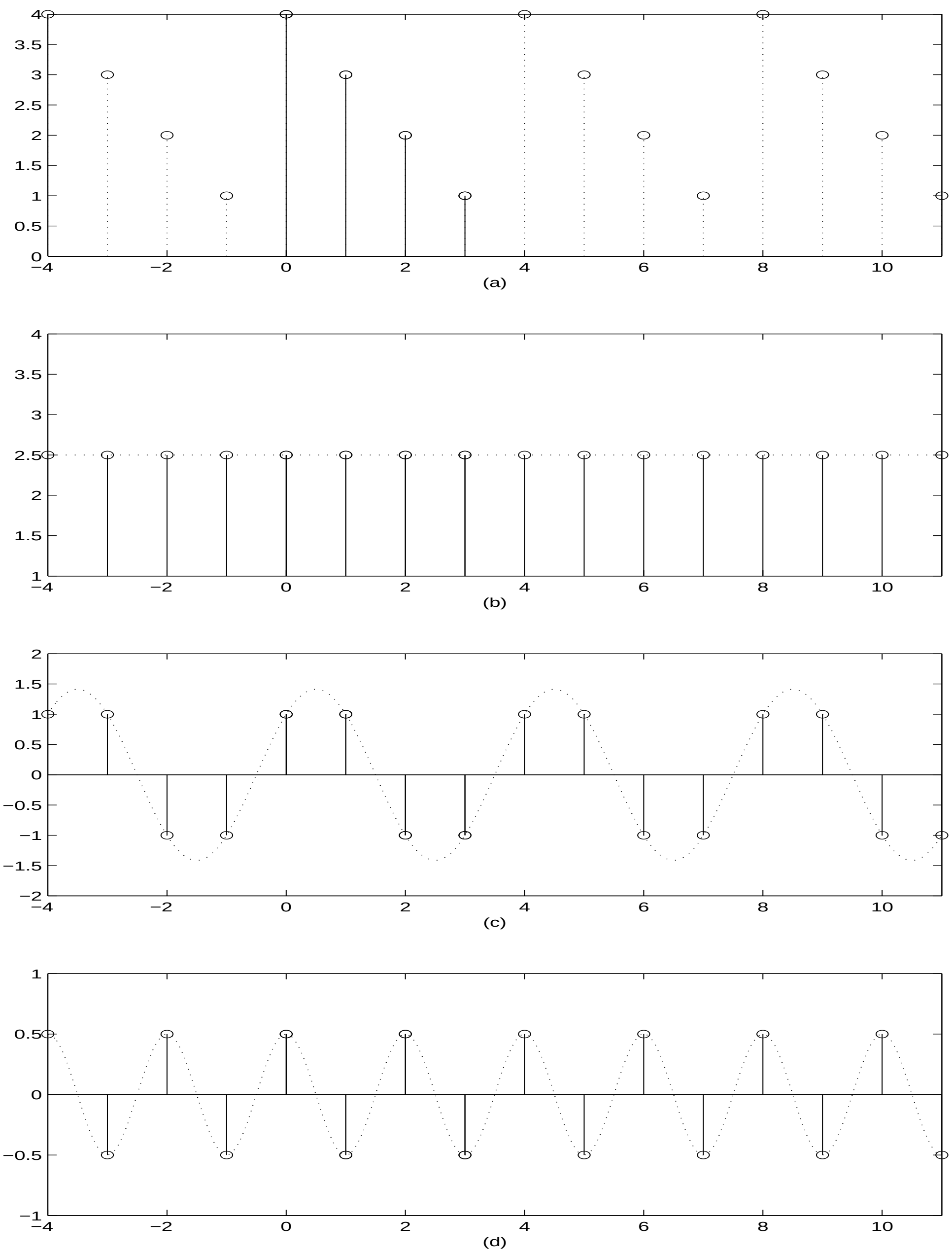


Figure 1.21: Example of Fourier decomposition for a discrete time periodic signal. The decomposed signal has period $N = 4$ (a) and is decomposed as a sum of 3 discrete sinusoids. The first component is a constant, which corresponds to a sinusoid of period 1 (b), the second is a sinusoid of period 4 (c), and the third is a sinusoid of period 2 (d)

To determine the unknowns, we first rewrite the equation (1.7) by recalling the trigonometric formula for the sine of a sum. We obtain,

$$x(n) = A_0 + A_1 \cos\left(2\pi\frac{n}{4}\right) + B_1 \sin\left(2\pi\frac{n}{4}\right) + A_2 \cos\left(2\pi\frac{n}{2}\right) + B_2 \sin\left(2\pi\frac{n}{2}\right),$$

where

$$A_i = P_i \sin(\phi_i), \qquad B_i = P_i \cos(\phi_i), \qquad i = 0, 1, 2, \tag{1.8}$$

(to keep a uniform notation, we represent also the constant X_0 as a sinusoid of period 1). In this way, each sample is written as a linear combination of the unknowns A_0, A_1, A_2, B_1, B_2 and we can write an equation for each sample $x(n)$. We remark that the factor $\sin(2\pi\frac{n}{2})$, that multiplies B_2, is zero for each value of the time index n. Hence, we can discard this term and assume that $B_2 = 0$ (actually, any value of B_2 is compatible with the constraints). In conclusion, we can write a linear systems of equations, that is

$$\begin{cases} A_0 + A_1 + A_2 = 4 \\ A_0 - A_2 + B_1 = 3 \\ A_0 - A_1 + A_2 = 2 \\ A_0 - A_2 - B_1 = 1 \end{cases}.$$

We can solve this system by using the usual tricks. For example, by summing the 4 equations we cancel all variables except A_0; therefore,

$$A_0 = \frac{4+3+2+1}{4} = \frac{5}{2}.$$

Similarly, we can compute the oher unknowns. We obtain,

$$A_0 = \frac{5}{2}, \qquad A_1 = 1, \qquad A_2 = \frac{1}{2}, \qquad B_1 = 1.$$

In conclusion, we can write the decomposition of the input signal as,

$$x(n) = \frac{5}{2} + \cos\left(2\pi\frac{n}{4}\right) + \sin\left(2\pi\frac{n}{4}\right) + \frac{1}{2}\cos\left(2\pi\frac{n}{2}\right).$$

To find the representation with the structure given by (1.7), we have to combine sinusoids and cosinusoids with the same frequency. In this case, the second and third terms are combined, by computing P_1 and ϕ_1 that satisfy equations (1.8). In our example, we have

$$P_1 = \sqrt{A_1^2 + B_1^2} = \sqrt{2}, \qquad \phi_1 = \frac{\pi}{4},$$

and the result is

$$x(n) = \frac{5}{2} + \sqrt{2}\sin\left(2\pi\frac{n}{4} + \frac{\pi}{4}\right) + \frac{1}{2}\sin\left(2\pi\frac{n}{2} + \frac{\pi}{2}\right). \tag{1.9}$$

The terms of the decomposition are depicted in figure 1.21(b-d).

It should be clear, at this point, that for any arbitrary discrete periodic signal, we can repeat the derivation of the example and we can always determine the decomposition. However, it would be tedious to solve a system of equations every time we want to determine a Fourier decomposition. Luckily, we can write directly the solution, for any period N. This is the topic of the next section.

1.3.3 Definition

In order to simplify the representation of the Fourier decomposition it is convenient to replace the sinusoid with a complex exponential. In fact, recall that

$$Pe^{j(2\pi ft+\phi)} = P\cos(2\pi ft+\phi) + jP\sin(2\pi ft+\phi),$$

where $j = \sqrt{-1}$ is the imaginary unit. Therefore, the complex exponential is a way of representing two sinusoids of equal frequency and a phase difference of $\pi/2$. The two components are found on the real and the imaginary part of the complex exponential. In our case, we consider discrete time sinusoids with frequencies k/N, $k = 0, 1, \ldots, N-1$; hence, we consider terms of the form $e^{j2\pi\frac{k}{N}n}$ and we look for a decomposition with the following structure,

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi\frac{k}{N}n}. \tag{1.10}$$

For example, let's rewrite the result of the previous section by using complex exponentials. We recall that,

$$\sin\alpha = \frac{e^{j\alpha} - e^{-j\alpha}}{2j},$$

and, from equation (1.9), we derive

$$x(n) = \frac{5}{2} + \sqrt{2}\frac{e^{j(2\pi\frac{n}{4}+\frac{\pi}{4})} - e^{-j(2\pi\frac{n}{4}+\frac{\pi}{4})}}{2j} + \frac{1}{2}\frac{e^{j(2\pi\frac{n}{2}+\frac{\pi}{2})} - e^{-j(2\pi\frac{n}{2}+\frac{\pi}{2})}}{2j}.$$

As you notice, there are only complex exponentials, but some correspond to negative frequencies. This is not a problem, since a multiplication by the term $e^{j2\pi n} = 1$ allows us to map these negative frequencies back into the range $[0, 1]$. We obtain, after some computations,

$$x(n) = \frac{1}{4}\left(X_0 + X_1 e^{j2\pi\frac{n}{4}} + X_2 e^{j2\pi\frac{n}{2}} + X_3 e^{j2\pi\frac{3n}{N}}\right),$$

with

$$X_0 = 10, \qquad X_1 = 2-2j, \qquad X_2 = 2, \qquad X_3 = 2+2j.$$

The next step is to determine the coefficients X_i directly from the samples $x(n)$, $n = 0, \ldots, 3$. The formula that allows such a computation is

$$X_k = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{k}{N}n}, \qquad k = 0, 1, \ldots, N-1, \tag{1.11}$$

which gives the Discrete Fourier Transform (DFT) of the periodic signal $x(n)$. The equation (1.10) corresponds to the Inverse Discrete Fourier Transform (IDFT) and allows to reconstruct $x(n)$ from the coefficients X_k. Let's verify that the DFT computed with equation (1.11) gives actually the good coefficients for equation (1.10). We do this by replacing (1.11) in (1.10) and we check that we actually reconstruct $x(n)$:

$$x(n) \stackrel{?}{=} \frac{1}{N} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} x(m) e^{-j2\pi \frac{k}{N} m} e^{j2\pi \frac{k}{N} n} = \frac{1}{N} \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} e^{j2\pi \frac{k}{N}(n-m)}.$$

In the last expression, we consider the term

$$\sum_{k=0}^{N-1} e^{j2\pi \frac{k}{N}(n-m)}$$

and we remark that this takes the value N when $n = m$, and is zero when $n \neq m$. In conclusion, we actually reconstruct $x(n)$, which proves that the expression (1.11) gives the right DFT coefficients.

Example 1. *Let's apply the expression (1.11) to determine directly the DFT coefficients. We have that*

$$\begin{aligned} X_0 &= 4 + 3 + 2 + 1 = 10, \\ X_1 &= 4 + 3e^{-j\frac{\pi}{2}} + 2e^{-j\pi} + e^{-j\frac{3\pi}{2}} = 2 - 2j, \\ X_2 &= 4 + 3e^{-j\pi} + 2 + e^{-j3\pi} = 2, \\ X_3 &= 4 + 3e^{-j\frac{3\pi}{2}} + 2e^{-j3\pi} + e^{j\frac{9\pi}{2}} = 2 + 2j. \end{aligned}$$

We remark that these coefficients correspond exactly to those computed earlier (with a much longer computation).

1.3.4 Properties

The Discrete Fourier Transform satisfies some properties that are very useful in order to make computations. Here we give a list of the main ones

Hermitian symmetry

Even if the complex exponential simplifies computations, the signal $x(n)$ is usually real. This implies that the DFT coefficients have a certain structure. This structure is called **Hermitian symmetry** and is given by the equation:

$$X_k = X_{N-k}^*, \qquad k = 0, 1, \ldots, N-1.$$

where the $*$ is the complex conjugate of a complex number (i.e it leaves the real part unchanged and changes the sign of the imaginary part).

Example 2. *This property simplifies the computation of the DFT of real sequences. In fact, consider again the example of section 1.3.2 and take into account the Hermitian symmetry. We have that*

$$X_0 = X_N^* = X_0^*, \qquad X_1 = X_3^*, \qquad X_2 = X_2^*;$$

therefore, only X_0, X_1, and X_2 need to be computed with the expression (1.11, while X_3 can be easily computed from X_1. Moreover, we have that X_0 and X_2 are real quantities.

Linearity

The DFT is a linear operation. In fact, if we take two signals $x(n)$ and $y(n)$ with equal period N, then we can construct the new periodic signal $z(n) = ax(n) + by(n)$, with a and b arbitrary numbers. The linearity of the DFT means that the DFT coefficients of $z(n)$ are given by

$$Z_k = aX_k + bY_k, \qquad k = 0, 1, \ldots, N-1.$$

Time delay

Suppose that the DFT of the signal $x(n)$ is X_k, then the DFT of the signal $y(n) = x(n-M)$, where M is an arbitrary integer, is

$$Y_k = e^{-j2\pi \frac{k}{N} M} X_k, \qquad k = 0, 1, \ldots, N-1$$

Example 3. *For example, take the periodic signal*

$$y(0) = 2, \quad y(1) = 1, \quad y(2) = 4, \quad y(3) = 3, \quad \ldots$$

with period $N = 4$. This is simply the signal $x(n)$ of section 1.3.2 delayed by $M = 2$ samples. Therefore, we can easily write the DFT of $y(n)$ as

$$Y_0 = X_0 = 10, \quad Y_1 = e^{-j2\pi\frac{2}{4}} X_1 = -2+2j, \quad Y_2 = e^{-j2\pi\frac{4}{4}} X_2 = 2, \quad Y_3 = e^{-j2\pi\frac{6}{4}} X_3 = -2-2j.$$

1.3.5 Convolution

The main interest for the Fourier transform comes from the special behavior of sinusoids when they are filtered by a linear time-invariant filter. In fact, remember that, in general, a filter changes the shape of a signal. One of the few exceptions is represented by the sinusoid, which is modified only in amplitude and phase. Since the Fourier transform represents a signal a sum of sinusoid, it is not surprising that the filtering operation has a simpler description if we consider the Fourier domain. In order to see that, let's take a periodic signal $x(n)$ with period N and suppose that we filter it with the filter $h(n)$. As we know from the previous chapter, the output of the filter, $y(n)$ is given by the convolution product between $x(n)$ and $h(n)$. For simplicity, we suppose that $h(n)$ is an FIR filter of length smaller or equal than N. In this case, we can write the output signal as

$$y(n) = \sum_{m=0}^{N-1} h(m) x(n-m).$$

Then, we apply the Discrete Fourier Transform and we write $x(n)$ as in equation (1.10), which gives

$$y(n) = \frac{1}{N} \sum_{m=0}^{N-1} h(m) \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{k}{N}(n-m)}.$$

If we change the order of the sums and we take into account the properties of the exponential, we obtain

$$y(n) = \sum_{k=0}^{N-1} X_k \sum_{m=0}^{N-1} h(m) e^{-j2\pi \frac{k}{N} m} e^{j2\pi \frac{k}{N} n}. \tag{1.12}$$

The inner sum is a quantity that depends only on $h(n)$ and k, we can write it as

$$H_k = \sum_{m=0}^{N-1} h(m) e^{-j2\pi \frac{k}{N} m}, \qquad k = 0, 1, \ldots, N-1,$$

but this quantity is simply the DFT of the sequence $h(n)$, $n = 0, \ldots, N-1$. In other words, we treat $h(n)$ as if it were a periodic sequence of period N and we compute the Discrete Fourier Transform. If we replace it in the expression (1.12), we obtain

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k H_k e^{j2\pi \frac{k}{N} n},$$

and we realize that this is the Inverse Discrete Fourier Transform, i.e. equation (1.10), applied to the sequence $X_k H_k$. This means that, if we take the DFT of $y(n)$, call it Y_k, we have that

$$Y_k = X_k H_k, \qquad k = 0, 1, \ldots, N-1.$$

We can interpret this result by saying that in the Fourier domain, the convolution product becomes a normal product. This result is called the *convolution theorem* and simplifies greatly the computation of the convolution product. Remark that in the version of the theorem that we derived here, the input signal is periodic and the impulse response of the filter is not longer than the period of the input signal. Extension of the theorem to other cases exist, but they are beyond the scope of these notes.

Example 4. *Suppose that we take again the signal $x(n)$ of section 1.3.2 and we send it to the filter*

$$h(n) = \begin{cases} 3 & n = 0, \\ -1 & n = 1, 2, 3, \\ 0 & \textit{elsewhere,} \end{cases}$$

how do we apply the convolution theorem? We already have the DFT of the input signal, we need to compute the DFT of the sequence $h(n)$, $n = 0, \ldots, 3$. As we did previously, we obtain

$$H_0 = 0, \quad H_1 = H_2 = H_3 = 4.$$

Therefore, the DFT of the output signal is

$$Y_0 = X_0 H_0 = 0, \quad Y_1 = X_1 H_1 = 8 - 8j, \quad Y_2 = X_2 H_2 = 8 \quad Y_3 = X_3 H_8 = 8 + 8j.$$

To compute the output signal in the time domain, we apply the Inverse Discrete Fourier Transform and we find,

$$y(0) = 6, \quad y(1) = 2, \quad y(2) = -2, \quad y(2) = -6, \quad \ldots$$

Of course, the output signal is also periodic with period 4.

1.3.6 Exercises

1. Consider the signal $x(n) = \text{rem}(n, 3)$, where $\text{rem}(a, b)$ is the reminder of the division of a by b. Find the period of $x(n)$ and compute the DFT. Write $x(n)$ as the sum of discrete time sinusoids.

2. Suppose that a filter has the impulse response

$$h(n) = \begin{cases} 2 & n = 0 \\ -1 & n = 1 \\ 1 & n = 2 \\ 0 & \text{elsewhere} \end{cases}$$

   and that the signal $x(n) = \text{rem}(n, 3)$ is sent at the input of the filter. Compute the signal at the output by applying the convolution theorem. Verify that the result is correct by computing the convolution in the regular way.

3. Suppose that the DFT of the signal $x(n)$ is $X_k, k = 0, \ldots, N-1$ and you build the signals $y(n) = \sin(\frac{2\pi}{N}Mn)x(n)$, $z(n) = \cos(\frac{2\pi}{N}Mn)x(n)$, with M an arbitrary integer number. Compute the DFT of $y(n)$ and $z(n)$ by using the DFT of $x(n)$.

4. Consider the signal $x(n)$, periodic with period $N = 4$, that takes the values $x(0) = 3, x(1) = 1, x(2) = -3, x(3) = 1, \ldots$. Determine a sinusoidal signal that approximates $x(n)$. (hint: determine the sinusoidal component with the maximum amplitude)

1.4 Sampling and interpolation

1.4.1 Introduction

In this lecture we are going to show the motivations for digital systems. In the first lecture, we saw that physical signals are often continuous-time signals. Digital systems can only perform operations of finite complexity on a finite interval of time. Therefore it seems that digital systems are appropriate only to treat discrete-time signals. Why are digital systems so widely used? We will see that we can use a digital system to process a class of continuous-time signals. The scheme is shown in Figure 1.22. The input signal $x(t)$ is a continuous-time signal, for example it could be an audio signal. The **sampler** transforms it into the discrete-time signal $\bar{x}(n)$ which is processed. Processing includes filtering, as seen in the previous lecture, but also transmission through a digital connection or recording on disks and tapes. The result is the discrete-time signal $\bar{y}(n)$ which is converted to the continuous-time signal $y(t)$ by the **interpolator**.

Why do we want to process a continuous-time signal using such a scheme? As you see this system includes two conversions which potentially introduce errors (we will see that later) and cost money. In electronics, you will study continuous-time systems (also called *analog* systems) that perform operations similar to what you can do with a digital system. For example, you can design analog filters similar to the discrete-time filters that we saw in the previous lecture. Why

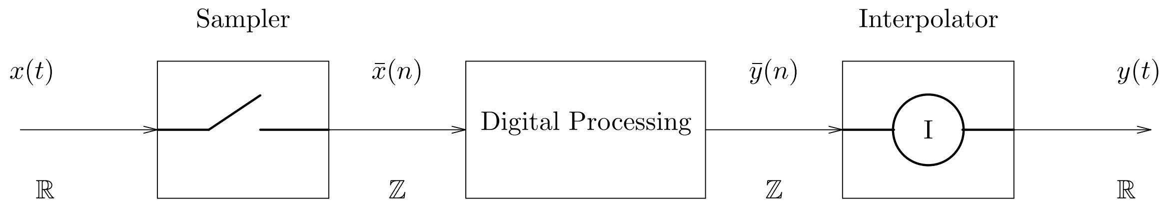


Figure 1.22: Example of a digital system that processes a continuous-time signal. The input signal is transformed to a discrete-time signal and then processed (the domain of the signals is indicated under the arrows that join the blocks). Processing includes filters and also transmission devices such as internet or disk and tape recording. The interpolator transform the result to a continuous-time signal.

do we prefer digital systems? There are several reasons. One is that it is very difficult to obtain good performances for certain media when analog signals are used. An example is the compact disk. Bits are represented on the surface of a disk by means of cavities. The presence or absence of a cavity at a certain position is associated to a binary digit. In principle, one could use a groove with depth proportional to an analog signal, but it would be difficult to measure such a depth. It seems much easier to make the difference between the conditions of presence and absence of the cavity rather than a measure of depth. Another advantage of digital system is their robustness in terms of reliability and stability. In the previous lecture, we saw how we can realize a digital filter. Depending on the parameters of the filter we could see a certain behavior. Such a parameters correspond to numerical values used in computer programs, therefore they do not change over time. On the other hand, an analog system is the result of the connection of some electronic components. The parameters of the components correspond to the parameters of the digital filter, but in this case they may change over time and with temperature. As a result, an analog filter is more sensitive to temperature and ages over time (while a digital system breaks abruptly). Digital systems are also very flexible. Since processing is realized with programs, it is very easy to modify them when there is a new need. Moreover, you can have different programs for different applications. This is exactly what you do with a personal computer. These considerations motivated the replacement of analog systems by their digital counterparts in the last decades. Nowadays, analog systems are limited to few applications, such as the interfaces with digital systems, high power systems and low cost devices.

In the next sections we will examine each block of the chain of Figure 1.22. In particular, we are interested in the problem of recording an audio signal, that is, the processing block is a device that records and then reads from a disk. We will see that for this scheme, under some appropriate conditions, the whole chain is "transparent" to the input signal. This means that the signal at the output of the chain is not simply a good approximation of the input signal, but it is *exactly equal* to the input signal. This result is not intuitive, since one has the intuition that discrete-time signals are "less powerful" than continuous-time signals. We will see that this is only partially true.

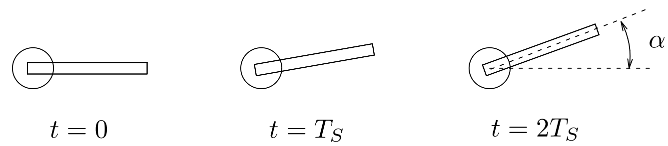


Figure 1.23: Experiment of the rotating bar illuminated by a stroboscope. The light flashes with frequency $f_S = 1/T$. Using the illuminated positions, an observer tries to determine the frequency of rotation f of the bar.

1.4.2 Sampling

A sampler is a system that takes a continuous-time signal and maps it to a discrete-time signal. Mathematically, we can write

$$\bar{x}(n) = x(nT) \quad \forall n \in \mathbb{Z},$$

where $x(t)$ is the continuous-time signal and $\bar{x}(n)$ is the discrete-time signal. The parameter T is the **sampling interval**. The **sampling frequency**, or **sampling rate**, is $f_S = 1/T$, in units of samples per second (or sometimes Hertz).

Sampling a sinusoid

Let $x(t)$ be the sinusoidal signal

$$x(t) = \sin(2\pi f t),$$

where f is the frequency in Hertz. Then, the output of the sampler is the discrete-time sinusoid

$$\bar{x}(n) = \sin(2\pi f n T). \tag{1.13}$$

The sampled sinusoid looks similar to the continuous-time sinusoid. However, there is a fundamental difference. Since n is discrete, the frequency f is undistinguishable from the frequency $f + f_S$ when the discrete-time signal is observed. This phenomenon is called **aliasing**.

Aliasing

Let us consider the following experiment. Suppose that we fix a bar to an electric motor. The bar is fixed on one of the two extremities so that the motor can rotate it (see Figure 1.23). We observe the bar in a dark room, using a stroboscopic light. The light flashes at a regular frequency f_S so you see the position of the bar only when the light flashes. We can change the frequency of rotation of the bar f. An observer tries to measure the frequency of rotation by using only the position of the bar when the light flashes. Is it possible to deduce the correct frequency? Let the frequency f begin at 0 Hz and sweep upwards to at least f_S. Let us fix the convention that a positive frequency corresponds to counterclockwise rotation and a negative frequency to clockwise rotation. When the frequency is very low, the movement of the bar between two consecutive flashes is small and you can follow the movement of the bar. Now, we increase the speed of rotation up to the frequency $f_s/2$ (Figure 1.24). The bar is illuminated

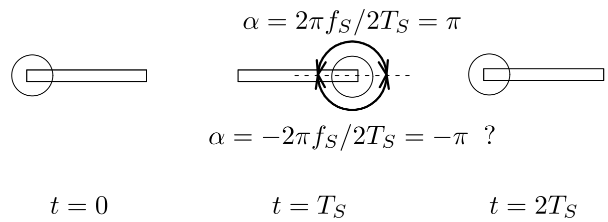


Figure 1.24: Ambiguity of the frequency f of rotation when it corresponds to half the sampling frequency f_S.

only when the angle of rotation α is 0 or 180 degrees. At this point we realize that there is an ambiguity: what would change if the bar rotated clockwise at the same frequency, i.e. $f = -f_S/2$? In both cases the bar appears at the two positions $0°$, $180°$, therefore we do not notice any difference between the cases $f = f_S/2$ and $f = -f_S/2$. If we continue to increase the frequency of rotation, we will probably perceive a clockwise rotation (negative frequency) rather then a counterclockwise rotation. This occurs because our eyes tend to interpolate the movement of the bar in the direction where the angle of rotation between consecutive positions is minimum. Let us call $\alpha = 2\pi f nT$ the angular position of the bar. If f is bigger then $f_S/2$, the angle of rotation between two flashes is $\Delta\alpha = 2\pi fT > \pi$ and there is an ambiguity with the opposite rotation $\Delta\alpha' = \Delta\alpha - 2\pi$. In other words the frequency $f > f_S/2$ is perceived as $f - f_S$. If we continue to increase the frequency of rotation, we reach $f = f_S$ and the light flashes only when $\alpha = 2\pi f_S T = 0$. Hence, we do not see the bar rotating anymore. If we summarize this experiment with a formula, we can write

$$\alpha_P = \alpha + N2\pi,$$

that is, the perceived angular position of the bar α_P has two types of ambiguities. The first one is the unknown number of complete turns during two consecutive flashes. The second one is the direction of rotation of the bar. If we see the bar at the position α_P, we do not know if it is actually $\alpha_P - 2\pi$. In other words, we do not know the correct sign of α. These ambiguities on the angle correspond to ambiguities on the perceived frequency of rotation f_P,

$$f_P = f + Nf_S.$$

This relation is depicted in Figure 1.25. The dotted lines correspond to the frequencies that are indistinguishable from the real one. The frequency $f_S/2$ is called the **Nyquist frequency**, after Harry Nyquist. This frequency represents a threshold in the relationship between discrete-time and continuous-time signals. The intuitive reason is that if the input frequency is below the Nyquist frequency (below half the sampling frequency), then we take more than two samples per turn. In this case, the samples capture the rotation of the bar. Two or fewer samples would not do this. The rotation of the bar appears as one of another frequency.

This experiment with a bar translates directly to the problem of sampling of a sinusoid. The angle of the bar corresponds to the argument of the sine, and the signal $x(t) = \sin(2\pi ft)$ is the vertical position of the extremity of the bar. The flash of the light corresponds to the sampling

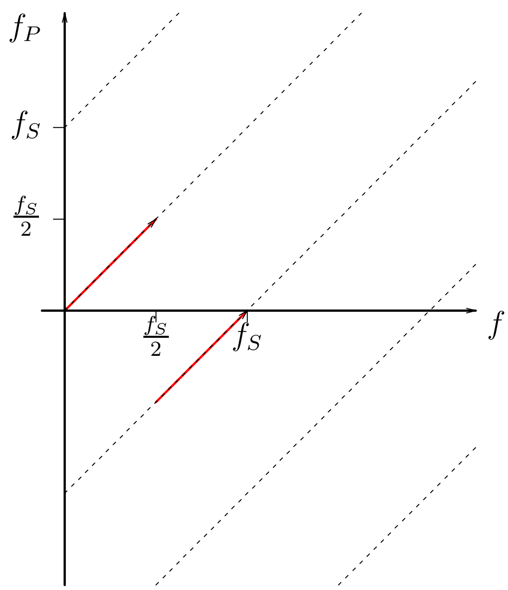


Figure 1.25: Ambiguity of the perceived frequency of rotation f_P as the real frequency f increases. The dotted lines correspond to frequencies indistinguishable from f on the basis of the observations. The red line correspond to the perceived frequency in the rotating bar experiment. Our eyes perceive the movement of the bar in the direction that corresponds to the minimum rotation angle. Therefore, when $f \geq f_S/2$ an observer perceives the frequency $f_P = f - f_S$ instead of f.

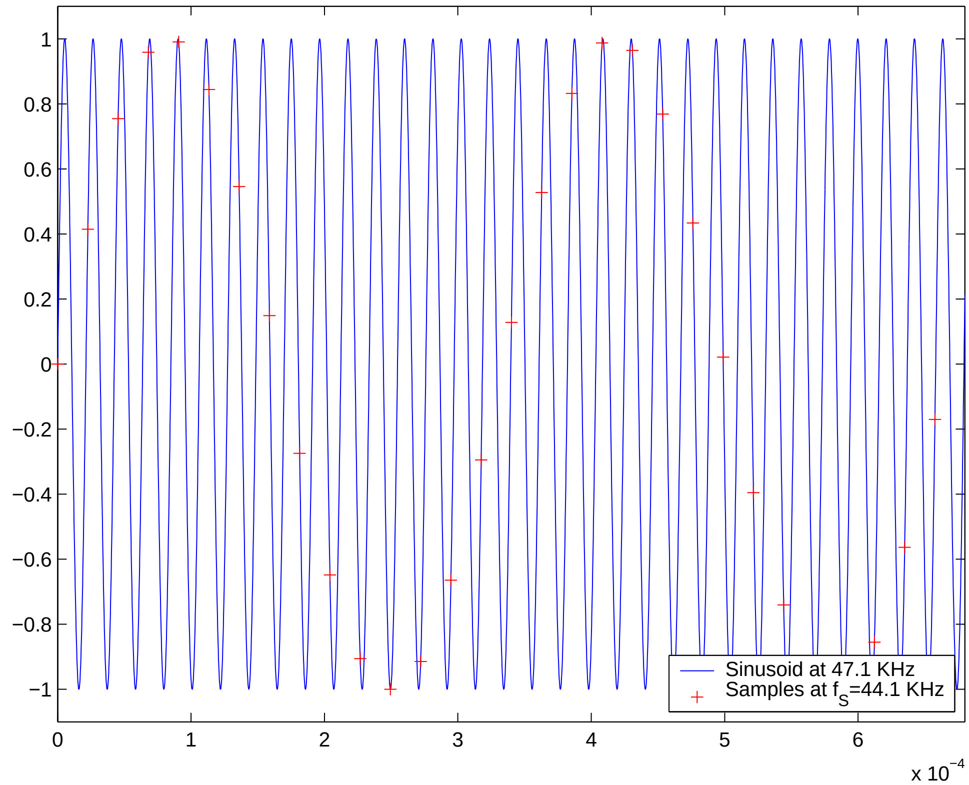


Figure 1.26: A sinusoid at 47.1 KHz and samples taken at 44.1 KHz (sampling frequency used for compact disk recording). The samples are indistinguishable from those taken from a sinusoid at 3 KHz.

of the signal. Therefore, consider the signal

$$x_P(t) = \sin(2\pi f_P t) = \sin(2\pi(f + N f_S)t),$$

where N is some integer and f_S is the sampling frequency. If $N \neq 0$, then this signal is clearly different from $x(t)$. However, when we sample $x_P(t)$ we obtain,

$$\begin{aligned} \bar{x}_P(n) &= \sin(2\pi(f + N f_S)nT) = \sin(2\pi f nT + 2\pi N n) \\ &= \sin(2\pi f nT) = \bar{x}(n) \;\; \forall n \in \mathbb{Z} \end{aligned}$$

because Nn is an integer. Thus, even if $x_P \neq x$, $\bar{x}_P = \bar{x}$, and after being sampled, the signals x and x_P are indistinguishable. This phenomenon is called **aliasing**, because any discrete-time sinusoid has many continuous-time identities.

For example, compact disks are created by sampling audio signals at $f_S = 44.1$ KHz, and so the sampling interval is about $T = 22.7$ μs/sample. A continuous-time sinusoid with a frequency of 47.1 KHz, when sampled at this rate, is indistinguishable from a continuous-time sinusoid with frequency 3 KHz, when sampled at the same rate. The result is shown in Figure 1.26.

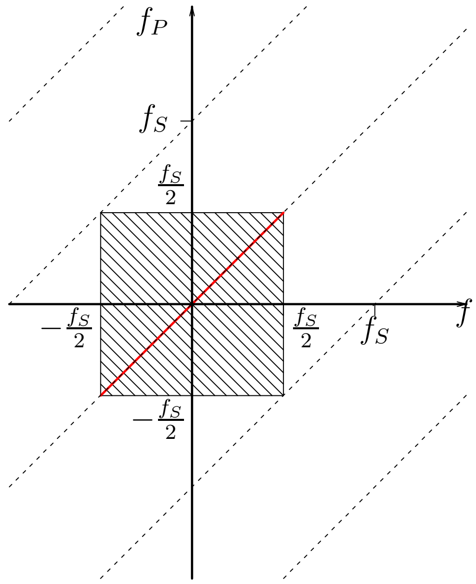


Figure 1.27: The ambiguity on the input frequency is solved by limiting its range. The dashed square shows the region where there is a one-to-one correspondence between the input frequency and the perceived frequency.

Avoiding aliasing ambiguities

Figure 1.25 suggests that even if samples of a sinusoid correspond to an ambiguous frequency, it is possible to construct a uniquely defined continuous-time sinusoid from the samples by choosing the one unique frequency that is closest to 0. This always results in a reconstructed signal that contains only frequencies below the Nyquist frequency in magnitude. In other words, we restrict the frequency range of the input sinusoid to $(-f_S/2, f_S/2)$ in order to avoid the ambiguity (note that the extremes of the interval are excluded). This solution is depicted in Figure 1.27.

How do we limit the frequency range of the input sinusoid? In the previous lecture, we saw that filters are able to attenuate a sinusoid according to its frequency, so we can add a filter to the system before the sampler. Such a filter is called antialiasing filter, because it permits to avoid the ambiguity due to aliasing. When a sinusoid with frequency outside of the range $(-f_S/2, f_S/2)$ is sent to the input of the system, it is simply discarded. This limitation on the frequency range is the price to pay to replace the continuous-time sinusoid with its discrete-time version. However, you see that, if you increase the sampling frequency, you extend the frequency range.

How do we build an antialiasing filter? We note that this filter is continuous-time, since we want to add it before the sampler. In fact, after the sampler the ambiguity on the frequency is already present. Continuous-time filters are very similar to discrete-time filters. Similarly to what was shown in the previous lecture, we can define the impulse response and the properties of time-invariance, stability and causality. The ideal antialiasing filter is a special one. We would like it to be perfectly transparent to sinusoids of frequencies below the Nyquist frequency and stop perfectly sinusoids of other frequencies. It can be proved that such a filter has impulse response:

$$h(t) = \frac{\sin(\pi f_S t)}{\pi t} = f_S \text{sinc}(f_S t),$$

where $\text{sinc}(x) = \sin(\pi x)/(\pi x)$ is called the **sinc** function. It is difficult to show that this filter corresponds to the ideal antialiasing filter. However, you see that the filter is not causal, since the impulse response is not zero for $t < 0$. In practice, this ideal response can only be approximated.

After this discussion on sinusoids, one may think that the problem of aliasing has to be completely reformulated for a generic signal. If we send an arbitrary continuous-time signal to the sampler, under which condition do the samples represent unambiguously the signal? It turns out that what we have seen for sinusoids can be extended to other signals. In fact, it can be proved that any signal can be decomposed into a sum of sinusoids of different frequencies. This is called the **Fourier** decomposition. The sampler is a linear system, so we can imagine that we send each sinusoid at a time and we check if aliasing appears. We call **bandwidth** of the input signal the maximum frequency of its sinusoidal components. Clearly, if the bandwidth is lower than the Nyquist frequency, all the sinusoids can be reconstructed unambiguously and so can the input signal. Conversely, if the bandwidth is higher then the Nyquist frequency, at least one of the sinusoids is reconstructed with a wrong frequency and the reconstructed signal is different from the input signal. The antialiasing filter considered for sinusoids can be used for an arbitrary signal. Again, we take into account the linearity of the filter and we see that signals with bandwidth lower than the Nyquist frequency are not affected by the filter. For the other signals, the filter suppresses the components outside of the range $(-f_S/2, f_S/2)$.

Consider again the example of the compact disk. The antialiasing filter will suppress all components of frequency $|f| \geq f_S/2 = 22.05$ KHz. That solves the ambiguity between the sinusoids at 3 KHz and 41.1 KHz, since the second is discarded by the filter. The limit of 22.05 KHz seems reasonable to record audio, in fact the hearing system is also a low-pass filter and the frequency limit is at about 15 KHz.

1.4.3 Interpolation

Suppose that we design the sampler and the antialising filter according to the concepts described in the previous section. We know that if we apply a sinusoid at the input with frequency smaller then the Nyquist frequency, there is no ambiguity on the frequency but we still do not know how to reconstruct the sinusoid. Moreover, we want to treat signals more complex than a sinusoid. These signals should also be reconstructed from the samples. As anticipated, we call **interpolator** the device that transforms a discrete-time signal into a continuous-time signal. Consider again the diagram of Figure 1.22 and the case of audio recording. Except for a delay and neglecting quantization, the signal $\bar{y}(n)$ is equal to $\bar{x}(n)$. We assume this delay to be zero, to simplify the notation, i.e. we read the CD while we record it. In this case, the problem of interpolation is the reconstruction of the signal $y(t)$ from the samples $\bar{x}(n)$ in order to have $y(t)$ as close as possible to $x(t)$. Due to the definition of the signal $\bar{x}(n)$, it is natural to impose that

$$y(nT) = \bar{x}(n) = x(nT)$$

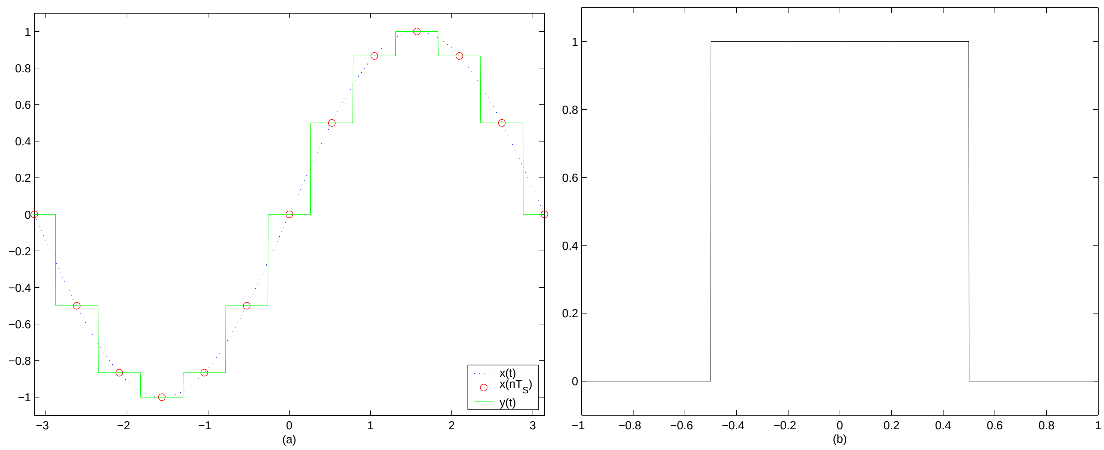


Figure 1.28: Interpolation with zero-order hold. (a) Interpolation of the samples of a sinusoid. Note the discontinuities introduced by this simple scheme. (b) The rect function can be used to describe mathematically the zero-order hold.

i.e. the interpolated signal has to pass through the available points. It remains to decide an interpolation scheme for the other points. Let us show two possibilities.

Zero-order hold

This interpolator approximates the signal $x(t)$ with a piecewise constant function. An example is shown in Figure 1.28(a). As you see, the interpolated signal is held constant on pieces of duration T (the sampling period) centered at the sampling positions. Formally, we can write

$$y(t) = \bar{x}(n) \quad nT - \frac{T}{2} \leq t < nT + \frac{T}{2}.$$

We want to rewrite this definition in a way that can be extended to other types of interpolations. Let us define the function

$$\text{rect}(t) = \begin{cases} 1 & \text{if} \qquad -1/2 \leq t < 1/2 \\ 0 & \text{otherwise} \end{cases},$$

depicted in Figure 1.28(b). The idea is to describe the constant pieces of $y(t)$ with a sum of rect functions. First note that $\text{rect}((t-nT)/T)$ takes the value 1, when t is in the set $[nT - T/2, nT + T/2)$, which corresponds to the generic piece of $y(t)$. Therefore, we can write

$$y(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n) \text{rect}\left(\frac{t-nT}{T}\right).$$

Note that this expression can be generalized to

$$y(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n) h\left(\frac{t-nT}{T}\right). \tag{1.14}$$

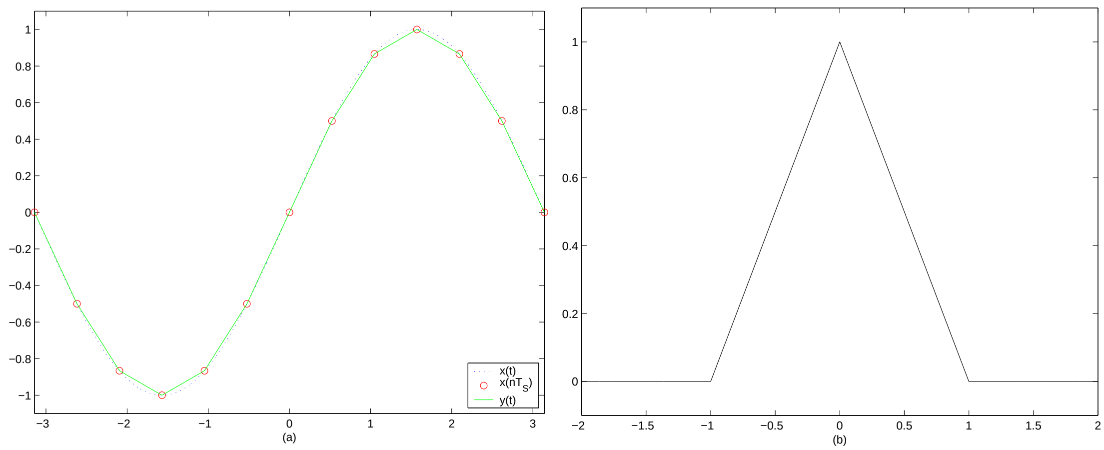


Figure 1.29: Linear interpolation (also called first-order hold). (a) Interpolation of the samples of a sinusoid using linear interpolation. (b) The triangular function is the interpolating function corresponding to the linear interpolation.

In fact, we obtain the zero-order hold by setting $h(t) = \mathrm{rect}(t)$. This is the general expression of the interpolator. By changing the interpolating function h, we can change the type of interpolation and the error with respect to the input signal $x(t)$.

Linear interpolation

A linear interpolator (sometimes called a first-order hold) simply connects the points corresponding to the samples with straight lines. An example is shown in Figure 1.29(a). We see immediately that this interpolator is already a good improvement with respect to the zero-order hold.

Can we put the linear interpolator in the form of equation (1.14)? Let us define the triangular function

$$\mathrm{triang}(t) = \begin{cases} 1 - |t| & \text{if} \quad -1 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

which is shown in Figure 1.29(b). It is easy to show that the triangular function corresponds to the linear interpolation in expression (1.14). In fact, note that the different translations of the triangular function $\mathrm{triang}((t - nT)/T)$ overlap on the straight parts, so that the result is actually made of straight lines.

Can we do something better than linear interpolation? As you have seen, for the zero-order hold, the interpolation at a certain time t was computed on the basis of a unique sample. For the linear interpolation, the result was obtained using two consecutive samples. You can imagine that we can obtain a better interpolator considering more and more samples. This is the main idea in the computation of an ideal interpolator. Before we can find the expression of such an interpolator, we consider a similar problem of interpolation of a finite set of points. We will find the ideal interpolator by taking the limit to infinity of the number of points.

Lagrange interpolation of a finite set of points

In this section we consider a problem slightly different from the interpolation of a discrete-time signal. We see the interpolation of a *finite* set of points. Let us consider the points $(t_1, \bar{x}_1), (t_2, \bar{x}_2), \ldots, (t_N, \bar{x}_N)$. We look for a polynomial $y(t)$ that passes for through all the points. This is called the **Lagrange interpolation** of the points.

We note that the minimum degree of the polynomial is given by $N-1$. In fact, two points define a line, i.e. a polynomial of degree 1, three points a parabola which is a polynomial of degree 2 and so on. Therefore, $y(t)$ is the unique polynomial of degree $N-1$ that passes through all the N points. One could compute the coefficients of the polynomial by writing a system of equations: each equation corresponds to the passage through one of the points. However, there is a simpler way to write directly the solution. In fact, consider the polynomials

$$L_i(t) = \frac{(t-t_1)(t-t_2)\cdots(t-t_{i-1})(t-t_{i+1})\cdots(t-t_N)}{(t_i-t_1)(t_i-t_2)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_N)} \quad i = 1, 2, \ldots, N.$$

We see that each polynomial $L_i(t)$ has degree $N-1$ and

$$L_i(t_j) = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases}.$$

In fact, the numerator has factors corresponding to every point except t_i and the denominator simplifies with the numerator when $t = t_i$. At this point, it is simple to write the expression of $y(t)$ by summing the polynomials $L_i(t)$ after scaling:

$$y(t) = \sum_{i=1}^{i=N} \bar{x}_i L_i(t).$$

In fact, the sum is a polynomial of degree $N-1$ as requested, and passes through all the points. An example is shown in Figure 1.30(a) for 5 points. The polynomials $L_i(t)$ are shown in Figure 1.30(b).

Ideal interpolator

In order to find the ideal interpolator we choose a finite number of points of $\bar{x}(n)$ and we find the Lagrange interpolator. The ideal interpolator is found by taking the limit for the number of points going to infinity. Let us choose the set of points $I^{(K)}$ where we compute the interpolation as:

$$I^{(K)} = \{(-KT, \bar{x}(-KT)), \ldots, (-T, \bar{x}(-T)), (0, \bar{x}(0)), (T, \bar{x}(T)), \ldots, (KT, \bar{x}(KT))\}.$$

As you see, the set of points is centered in 0 and K controls the number of points (which is $2K+1$). As in the previous paragraph, we can write the interpolation as

$$y^{(K)}(t) = \sum_{n=-K}^{K} \bar{x}(n) L_n^{(K)}(t), \tag{1.15}$$

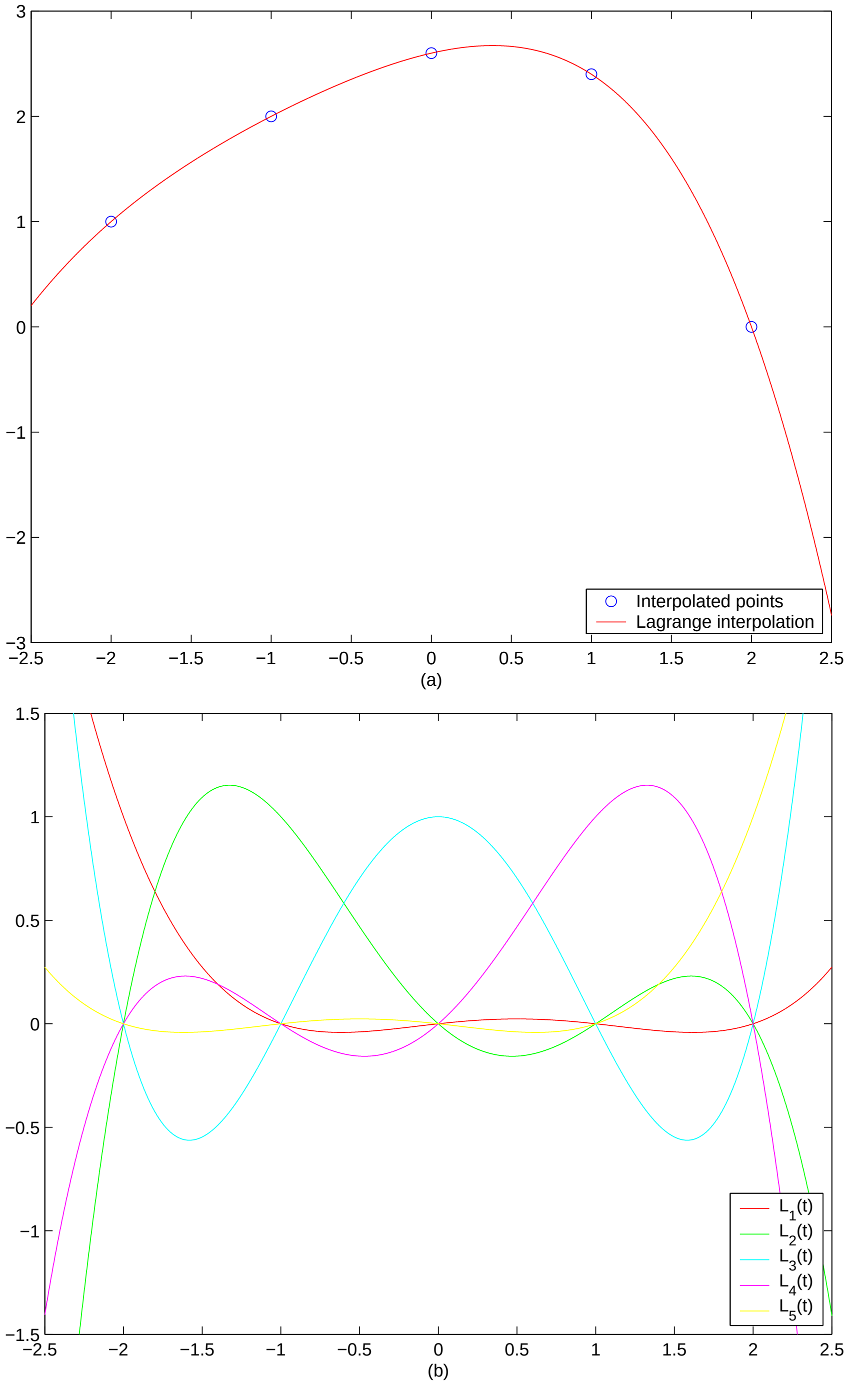


Figure 1.30: Lagrange interpolation. (a) Interpolation using 5 points. (b) The polynomials $L_i(t)$ used to compute the interpolation. Note that each of the $L_i(t)$ is zero for all the abscissas of the points except the point i, where it takes the value 1.

where the "$^{(K)}$" is used to mark the functions that depend on the number of points K. As in the previous paragraph, we can write directly the polynomials $L_n^{(K)}$. For example, $L_0^{(K)}(t)$ is given by

$$L_0^{(K)}(t) = \frac{(t+KT)(t+(K-1)T)\dots(t+T)(t-T)\dots(t-KT)}{KT(K-1)T\dots T(-T)\dots(-KT)}. \tag{1.16}$$

If we take the limit of equation (1.15) for K that goes to infinity, we obtain the ideal interpolator

$$y(t) = \lim_{K\to\infty} y^{(K)}(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n) L_n^{(\infty)}(t),$$

where we defined

$$L_n^{(\infty)}(t) = \lim_{K\to\infty} L_n^{(K)}(t).$$

Since we consider an infinite number of points, all the functions $L_n^{(\infty)}(t)$ are obtained by translation of the same function. For example, we can write

$$L_n^{(\infty)}(t) = L_0^{(\infty)}(t - nT).$$

Therefore, the ideal interpolator takes the form of equation (1.14). Note also that the sampling interval T is simply the scale along the time axis, hence

$$L_n^{(\infty)}(t) = g\left(\frac{t-nT}{T}\right),$$

for an appropriate function $g(t)$. We can compute numerically the function $g(t)$ by considering $L_0^{(K)}(t)$ for $T = 1$ and increasing values of K. The result is shown in Figure 1.31. Surprisingly enough, the limit function is $g(t) = \mathrm{sinc}(t)$. This is confirmed by Figure 1.32 and can be proved formally on equation (1.16) taking into account some formulas on infinite products. It seems very strange that the ideal antialiasing filter corresponds exactly to the ideal interpolator. Actually, this is not simply a coincidence but to understand this phenomenon you would need some advanced analysis.

There is another fact that concerns the ideal interpolator. Suppose that we interpolate the samples of a sinusoid of frequency below the Nyquist frequency. We know that the samples represent the sinusoid with no ambiguity. What is the result given by the ideal interpolator? The ideal interpolator is able to reconstruct the sinusoid exactly. This means that the ideal interpolator is not only a good interpolator but the optimal one. We can have an intuition of why this happens if we think about the Taylor expansion of $\sin(2\pi f nT)$. We see that we can write the samples as a sum of samples of polynomials. We know that the Lagrange interpolator approximates a function with polynomials of a certain degree. Therefore, when we the take limit on the number of points, the interpolator is able to approximate *any* polynomial. In conclusion, the ideal interpolator is able to reconstruct exactly any function for which the Taylor expansion converges on the whole axis, such as the sinusoid. Moreover, the ideal interpolator is, as the sampler, a linear system. Therefore, if the input signal is composed of several sinusoids (such as any signal of practical interest) it is also reconstructed exactly by the system. This important result gives the following **sampling theorem**:

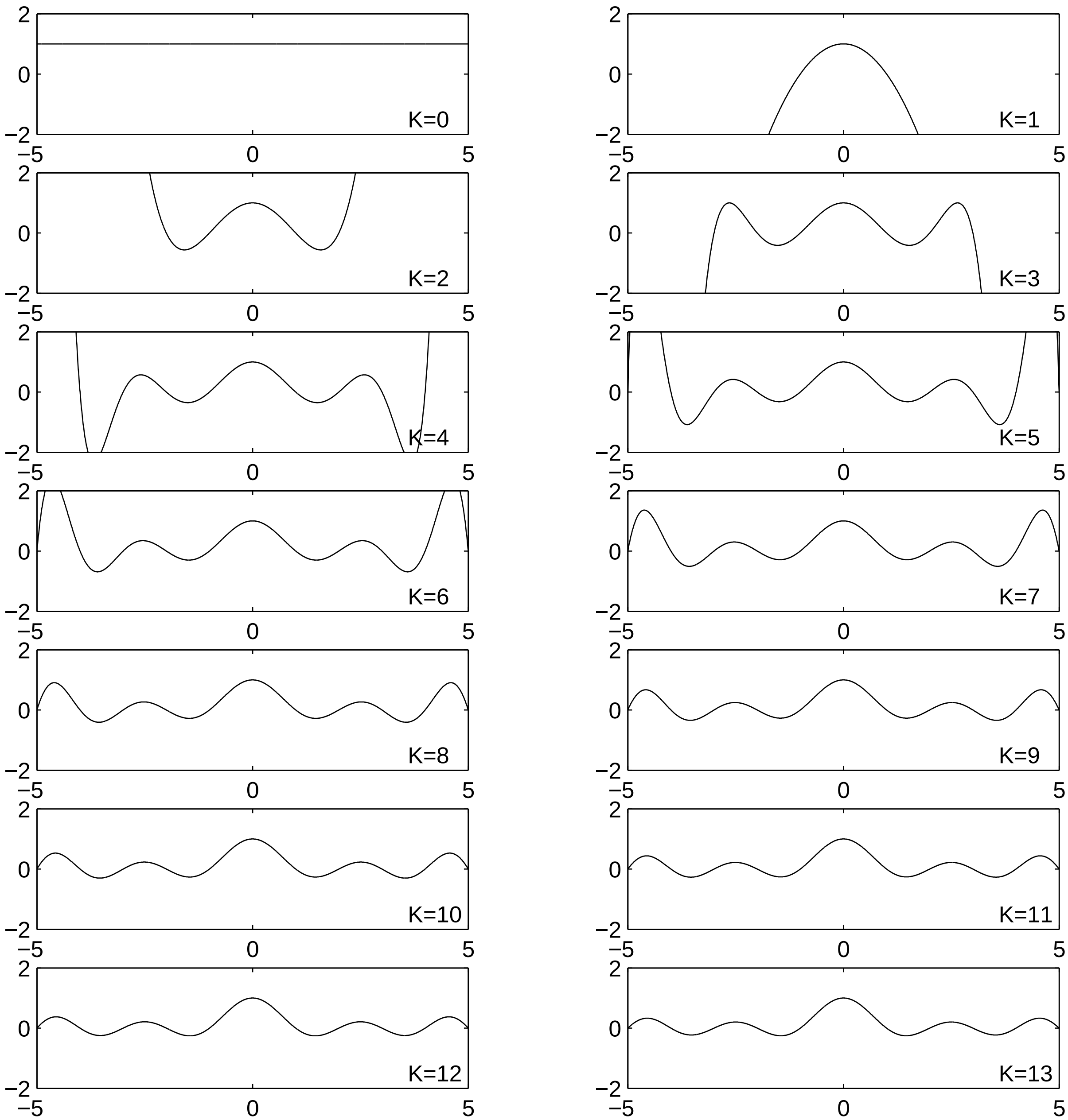


Figure 1.31: Polynomial $L_0^{(K)}(t)$ for different values of K and $T = 1$. As K increases the polynomial converges to the sinc function.

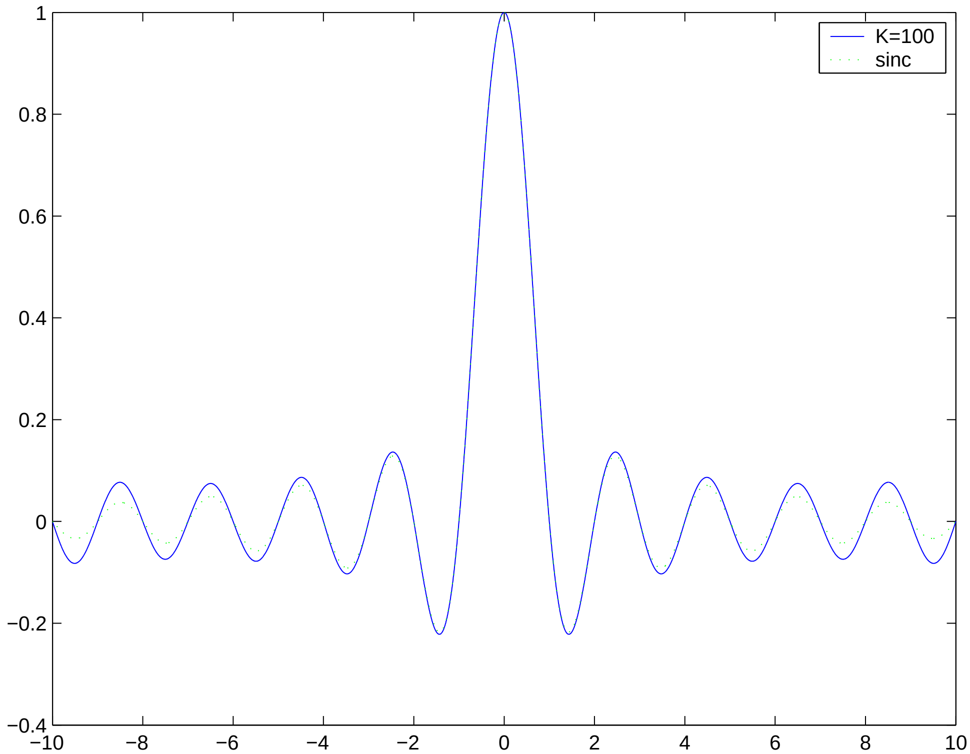


Figure 1.32: Polynomial $L_0^{(K)}(t)$ for $K = 300$ and $T = 1$, superposed to the sinc function.

Theorem 2. *Let $x(t)$ be a continuous-time signal with bandwidth B (i.e. it is composed of sinusoids with maximum frequency B) and $\bar{x}(n) = x(nT)$ be the samples of $x(t)$. If the sampling frequency $f_S = 1/T$ is such that*

$$B < \frac{f_S}{2}$$

then $x(t)$ can be reconstructed from the samples $\bar{x}(n)$ using the interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n) sinc\left(\frac{t - nT}{T}\right).$$

The sampling theorem is an important achievement of the beginning of the previous century. In fact, it shows that a digital system realized with an ideal antialiasing filter and interpolator is exactly equivalent to a continuous-time system with bandwidth equal to half of the sampling frequency. This equivalence motivated the replacement of many analog systems with digital systems, with the advantages that we mentioned at the beginning of the lecture.

1.4.4 Exercises

1. The signal $x(t)$ is a continuous time sinusoid, given by $x(t) = \sin(200\pi t)$.
   (a) Which are the values of the sampling frequency which allow the perfect reconstruction of the signal?
   (b) Can one use a sampling frequency of 200 Hz and have perfect reconstruction? Which would the discrete-time signal $\bar{x}(n)$ be in this case?

2. The signal $\bar{x}(n) = \cos(\pi n/4)$ has been obtained by sampling a continuous-time signal $x(t) = \cos(2\pi f t)$ using a sampling frequency of 1000 Hz.
   (a) Find two values of f that could have given the signal $\bar{x}(n)$.
   (b) Which is the signal $y(t)$ that one would reconstruct by applying the ideal interpolator to the signal $\bar{x}(n)$?

3. Consider the signal $x(t) = \sin(20\pi t) + \cos(40\pi t)$. Which are the sampling frequencies which allow the application of the Shannon-Nyquist theorem?

4. Consider the points $P_0 = (0, 1)$, $P_1 = (1, 0.8)$, $P_2 = (2, 2)$, $P_3 = (3, 4)$.
   (a) Draw the points and the result of piecewise-constant (zero-order hold) and linear interpolation.
   (b) Write the result of the piecewise-constant interpolation as a sum of rect functions.
   (c) Write the expression of the linear interpolation as a sum of triangular functions.
   (d) Write the expression of the Lagrange interpolation.

5. Consider the sinusoidal signal $x(t) = \sin(2\pi f t)$ where the frequency f is unknown. Suppose that $x(t)$ is sampled once using a sampling frequency $f_{S1} = 10$ Hz and once using the sampling frequency $f_{S2} = 12$ Hz. In both cases, the same discrete-time signal $\bar{x}(n) = 0, \forall n \in \mathbb{Z}$ is obtained.
   (a) Which are the possible values of the frequency f? Explain.
   (b) If you had sampled $x(t)$ only once, which sampling frequency would have been given the same ambiguity on the measure of f?

6. The signal $x(t)$ is a continuous-time sinusoid given by $x(t) = \sin(120\pi t)$.
   (a) Which are the sampling frequencies that allows the reconstruction of the signal?
   (b) Is it possible to use a sampling frequency of 120 Hz? Which would the discrete-time signal $\bar{x}(n)$ be in this case? If one uses an ideal interpolator to interpolate $\bar{x}(n)$, which would the reconstructed signal be?

7. The signal $\bar{x}(n) = \cos(\pi n/4)$ is obtained by sampling the continuous-time signal $x(t) = \cos(2\pi f t)$ at the sampling frequency of 1000 Hz.

(a) Determine two values of the frequency f that could have given the signal $\bar{x}(n)$

(b) If one applies the ideal interpolator to the signal $\bar{x}(n)$, which would the reconstructed signal $y(t)$ be?

8. Consider the signal $x(t) = \sin(25\pi t) + \cos(50\pi t)$. Which are the sampling frequencies that allows to avoid the problem of aliasing? If one sampled at the sampling frequency of 40 Hz and used an ideal interpolator to reconstruct the signal, which would the reconstructed signal $y(t)$ be?

9. Consider the points $P_0 = (0, -1)$, $P_1 = (1, 1.8)$, $P_2 = (2, 3)$, $P_3 = (3, 5)$.

   (a) Draw the points, the result of the piecewise-constant interpolator (zero-order hold) and the result of linear interpolator.

   (b) Write the result of the piecewise-constant interpolator as a sum of "rect" functions.

   (c) Write the result of the linear interpolator as a sum of "triang" functions.

   (d) Write the expression of the Lagrange interpolator.

1.5 Solutions to the exercises of the signal processing module

1.5.1 Solutions to the exercises of section 1.1

1. Certainly there are many physical quantities that can be written as one dimensional continuous signals $\mathbb{R} \to \mathbb{R}$. For example, temperature, pressure, speed, acceleration, tension and current.

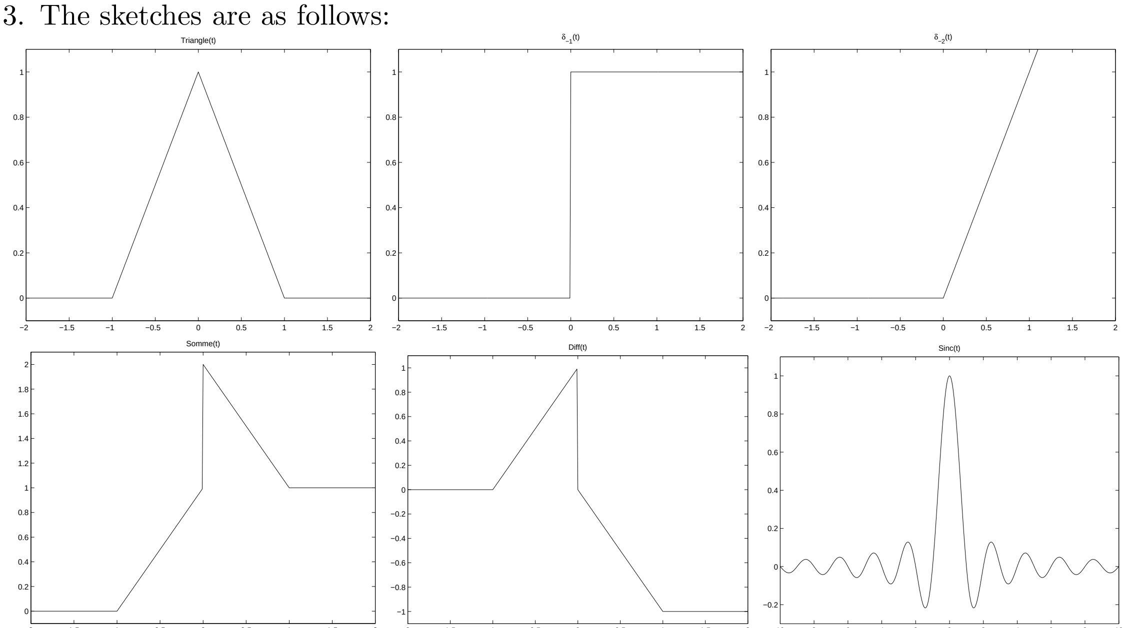
   In the case of two dimensional signals ($\mathbb{R}^2 \to \mathbb{R}$) we can consider the temperature on a surface as a function of position (also the relief of the earth as a function of longitude and latitude).

   For three dimensional signals ($\mathbb{R}^3 \to \mathbb{R}$) we have the physical parameters in the space(again temperature, pressure, etc.).

   For discrete signals we can consider all the preceding examples after sampling.

2. (a) $\mathbb{Z} \to \mathbb{R}$: a series of measurements of temperature, pressure, etc.
   (b) $\mathbb{R} \to \mathbb{R}^2$: an audio stereo signal, the position of a boat(longitude, latitude) as a function of time.
   (c) $\{0, 1, \ldots, 600\} \times \{0, 1, \ldots, 600\} \to \{0, 1, \ldots, 255\}$: a function that represents an image of 600×600 pixels as 256 gray levels (or 256 predefined colors).
   (d) In practice, in a computer all of the images are represented in this form (as an array of integers).

3. The sketches are as follows:



4. We know that $\cos(x) = \sin(x + \pi/2)$, therefore

$$x(t) = 5\cos\left(10t + \frac{\pi}{2}\right) = 5\sin(10t + \pi) = P\sin(\omega t + \phi).$$

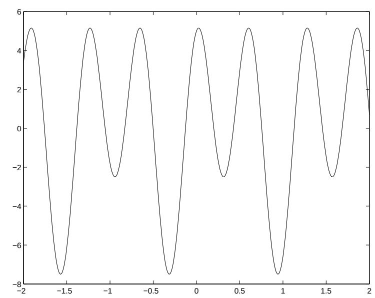
and we have $P = 5$, $\omega = 10$, $\phi = \pi$. The period is $T_P = 1/f = 2\pi/\omega = 1/5\pi s$.

5. We use the definition of a periodic signal:

   $$P_1 \sin(\omega_1 t + \phi_1) + P_2 \sin(\omega_2 t + \phi_2) = P_1 \sin(\omega_1 t + \omega_1 T_P l + \phi_1) + P_2 \sin(\omega_2 t + \omega_2 T_P l + \phi_2),$$

   $l \in \mathbb{Z}, T_P \in \mathbb{R}$. The equation holds when $\omega_1 T_P = 2\pi k_1$ and $\omega_2 T_P = 2\pi k_2$ for appropriate $k_1, k_2 \in \mathbb{Z}$. Therefore, $\omega_1/\omega_2 = k_1/k_2 \in \mathbb{Q}$, meaning that the ratio of frequencies should be rational. We calculate the period T_P by simplifying $\omega_1/\omega_2 = k_1/k_2$ so that k_1 and k_2 are coprime. We have $T_P = 2\pi k_1/\omega_1 = 2\pi k_2/\omega_2$.

6. As seen in the previous exercise, $\omega_1/\omega_2 = 2$ and the sum is periodic. The period is $T_P = 2\pi 2/\omega_1 = 2/5\pi$. The sketch of the function is as follows:



7. We know that there are no more than 16 colors per image, and each color is coded using 24 bits. Therefore, instead of backing up the colors of each pixel, we can only back up the number of the colors used among the 16 possibilities. This information can be coded by 4 bits. Since the 16 colors are different for each image, it is necessary to send the list of colors as well. In total, we have:

   - $768 \times 1024 \times 4$ bits for the numbers,
   - 16×24 bits for the list of colors used,

   this adds up to 393264 bytes. Notice that if we had saved the color of each pixel directly, we would have used $768 \times 1024 \times 24$ bits $= 2359296$ bytes.

8. An interesting example is a musical score. The signals are the sounds produced by the instruments, the notes are the symbols that correspond to the sounds. The notes are grouped in bars and bars in phrases.

9. (a) Each function $a \mapsto b(a)$ from A to B associates the values 0 or 1 to the values x, y, z. We can list the functions in the following table

		$a =$	
	x	y	z
	0	0	0
	0	0	1
	0	1	0
	0	1	1
$b =$	1	0	0
	1	0	1
	1	1	0
	1	1	1

(b) This concept can be applied to list all the functions from one set to the other, when the cardinality of the sets (the number of elements in a set) is finite. Let us suppose that m and n are the number of elements in A and B. In this case, the table has m columns and we have n possibilities to choose each element. Therefore, the table should have n^m lines that correspond to all the functions of S.

(c) In this case $m = 288 \times 720$, $n = 2^{24}$. The number of elements of S is

$$n^m = (2^{24})^{288\times 720} = 2^{24\times 288\times 720} = 10^{24\times 288\times 720\times \log_{10} 2} \simeq 10^{1498118}.$$

1.5.2 Solutions to the exercises of section 1.2

1. Answers to the questions:

   (a) No, a filter with finite impulse response (FIR) cannot be unstable. Actually, if $h(n)$ is the impulse response, we always have,

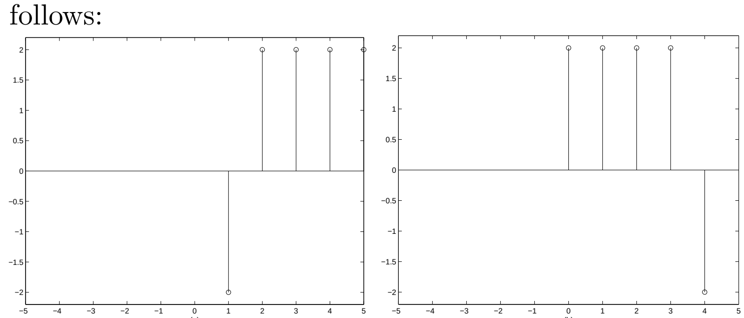
   $$\sum_{n=-\infty}^{\infty} |h(n)| < \infty.$$

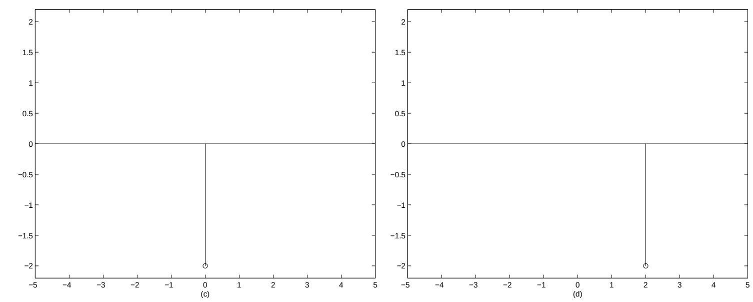
   (b) yes, a predictive system, for example of temperature, is also causal. Making a prediction is simply a form of calculation (even if it might be based on the experience of a human being) that uses the available data at the time of prediction.

   (c) The only solution is the exponential function:

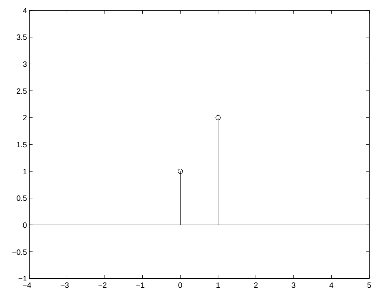
   $$x(n) = e^{sn} \quad s \in \mathbb{C}.$$

2. The graphs are as follows:





3. (a) The following graph is the impulse response:

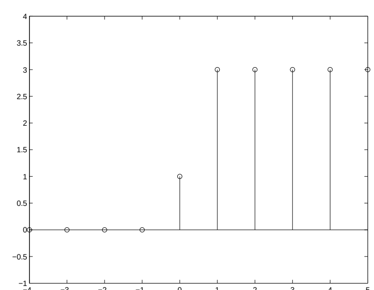


(b) You should pay attention that $x * \delta(n) = x$ (in fact, if the impulse response of a filter is an impulse, the input and the output of the filter are equal). Also, $x * \delta(n-1) = x(n-1)$ (the filter introduces a one sample delay). Therefore, using linearity

$$y(n) = (x * h)(n) = x(n) + 2x(n-1).$$

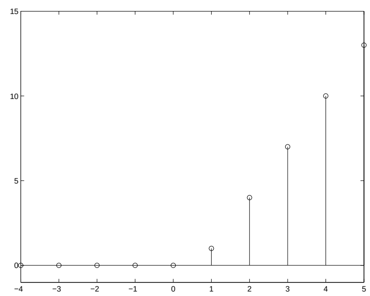
If $x = u$,

$$y(n) = u(n) + 2u(n-1) = \begin{cases} 3 & \text{if} \quad n \geq 1 \\ 1 & \text{if} \quad n = 0 \\ 0 & \text{if} \quad n < 0 \end{cases}$$



(c) As in the previous case, we have:

$$y(n) = r(n) + 2r(n-1) = \begin{cases} 3n-2 & \text{if} \quad n \geq 1 \\ 0 & \text{if} \quad n < 1 \end{cases}$$



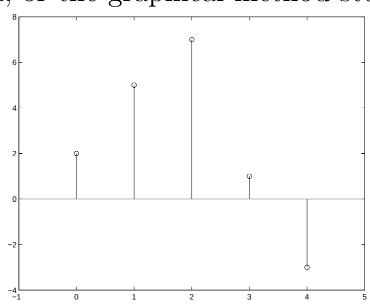
(d) Again we have,

$$y(n) = x(n)+2x(n-1) = \cos(\pi n/2+\pi/6)+\sin(\pi n+\pi/3)+2\cos(\pi n/2-\pi/3)+2\sin(\pi n-2\pi/3).$$

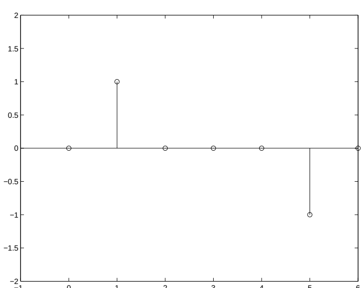
After some simplifications, we get

$$y(n) = (1+\sqrt{3}/2)\cos(\pi n/2) + (\sqrt{3}-1/2)\sin(\pi n/2) - \sqrt{3}/2\cos(\pi n).$$

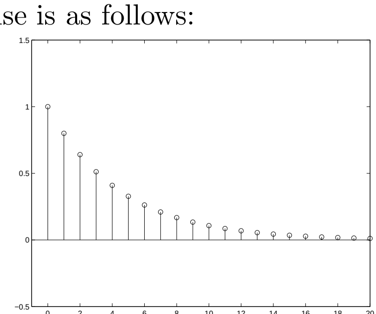
4. By using the convolution formula, or the graphical method studied in the course, we get



5. Like the previous exercise, we get



6. The graph of the impulse response is as follows:

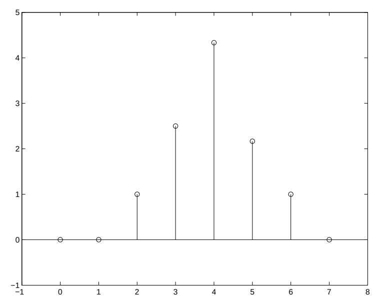


The filter is causal, since $h(n) = 0$ for $n < 0$. It is stable as

$$\sum_{n=-\infty}^{\infty} |h(n)| = \frac{1}{1 - 0.8} < \infty.$$

Il is definitely time invariant, since the impulse response only depends on n. The filter is not an FIR, since the impulse response is different from zero for an infinite number of samples.

7. Let us suppose that we have an impulse function at the input of the cascade of the two filters H_1 and H_2. Suppose that h_1 is the output of the first filter that enters the second system. Now suppose that h_2 is the impulse response of the second filter, therefore the output of the chain is $h = h_2 * h_1$ that corresponds to the impulse response of the cascade of two filters. Here is the result:



This system is also a filter, since we can see that:

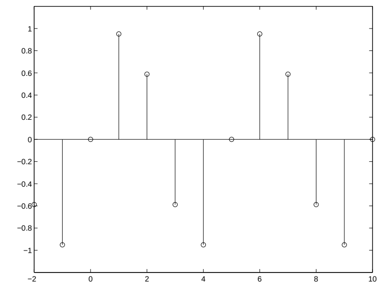
- It is linear (we can easily prove that the cascade of two linear systems is also linear)
- It is time invariant (the composition of two invariant systems is also time invariant)
- The domain of the input and output signals is the same (in this case it is $\mathbb{Z}$)

The resulting filter is an FIR filter (we can verify that by studying h, but in general the cascade of FIR filters is always an FIR filter). If we swap H_1 and H_2 we get exactly the same result since convolution is commutative.

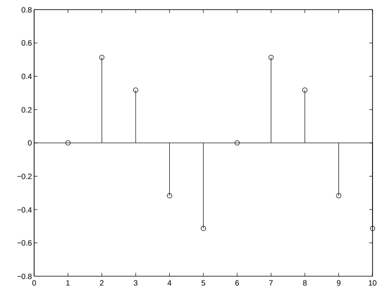
8. We can easily verify that the impulse response is

$$h(n) = \begin{cases} \frac{1}{L} & \text{if} \quad 0 \le n < L \\ 0 & \text{otherwise} \end{cases}$$

Therefore, it is a moving average. The filter is time invariant (the impulse response depends only on n) and causal (the response is zero for $n < 0$). The signal $x(n)$ is periodic and the period is 5:



Mobile averaging over 3 samples gives



We can prove that (see the notes) the output signal is also a sinusoid with period 5. If L increases, the amplitude of the output signal decreases. If L is a multiple of 5, the output is zero, since we add the samples of a multiple of the period.

9. (a) The system is linear, because if y_1 and y_2 are the outputs corresponding to x_1 and x_2, we have

$$y_1(n) = 3x_1(n) - 4x_1(n-1),$$
$$y_2(n) = 3x_2(n) - 4x_2(n-1).$$

If we multiply the two equations by $u_1, u_2 \in \mathbb{R}$ and sum them up, we obtain

$$(u_1y_1 + u_2y_2)(n) = 3(u_1x_1 + u_2x_2)(n) - 4(u_1x_1 + u_2x_2)(n-1).$$

Therefore, $y = u_1y_1 + u_2y_2$ is the output when the input is $x = u_1x_1 + u_2x_2$, which corresponds to the property of linearity. The system is stable. In fact, if $|x(n)| < N$, using the triangular inequality,

$$|y(n)| = |3x(n) - 4x(n-1)| \leq 3|x(n)| + 4|x(n-1)| < 7N.$$

Therefore, one can choose $M = 7N$ to satisfy the definition of stability. Let us set $x(n) = \delta(n-m)$, in other words an impulse at m. We get

$$\bar{h}(n,m) = y(n) = 3\delta(n-m) - 4\delta(n-m-1)$$

Therefore, the system is time invariant, since we can write $\bar{h}(n,m)$ as a function that depends only on $n-m$,

$$\bar{h}(n,m) = h(n-m).$$

(b) The system is linear. For the proof, one can proceed as in the previous case. To check if the system is stable, we compute first the impulse response. We replace $x(n)$ with an impulse function at position m, i.e. $x(n) = \delta(n - m)$. The output is,

$$y(n) = 2y(n-1) + \delta(n - m + 2).$$

The impulse acts only starting $n = m-2$, before the output is zero. When $n = m-2$ the output is 1, since $y(n-1) = 0$. When $n > m - 2$ the input is zero and $y(n) = 2y(n-1)$. Therefore, the output doubles at each step. In conclusion,

$$\bar{h}(n,m) = y(n) = \begin{cases} 2^{n-m+2} & \text{if} \quad n - m \geq -2 \\ 0 & \text{otherwise} \end{cases}.$$

The system is time invariant, since $\bar{h}(n,m) = h(n-m)$. The impulse response is

$$h(n) = \begin{cases} 2^{n+2} & \text{if} \quad n \geq -2 \\ 0 & \text{otherwise} \end{cases}.$$

We check if the system is stable by computing the sum

$$\sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=-2}^{\infty} 2^{n+2} = \sum_{n=0}^{\infty} 2^n = \infty,$$

and the system is unstable. The system is non-causal, since $h(-2) = 1 > 0$.

(c) As in the previous case, the system is linear. Concerning stability, we see that the condition $|x(n)| < N$ is not sufficient to guarantee that y is bounded, i.e. that $|y(n)| < M$ for some $M > 0$. For example, if we take $x(n) = \delta(n - K)$, the input is bounded for any value of K, i.e. $|x(n)| \leq 1$. However, the output is $y(n) = n\delta(n - K) = K\delta(n - K)$, which correspond to a pulse of amplitude K. Since K is arbitrary, we conclude that the output is not bounded and that the system is unstable. To check if the system is time-invariant, we apply the pulse $\delta(n - m)$ at the input to obtain the impulse response,

$$\bar{h}(n,m) = n\delta(n - m).$$

The factor n makes that it is not possible to write the impulse response on the form $\bar{h}(n,m) = h(n-m)$. Therefore, we conclude that the system is not time-invariant. For causality, we have that $\bar{h}(n,m) = 0$ when $n < m$ which implies that the system is causal.

(d) The system is not linear, since the cosine function is not linear. For example, the cosine of the addition of two angles is not the the addition of the cosine of the angles. The properties of stability, causality and time-invariance have been defined for linear systems only and cannot be verified for this system.

10. We should first note that the moving average acts on $s(t)$, so influences both the useful signal $m(t)$ and the noise $\eta(t)$. Since the moving average is a linear system, we can consider

separately the two terms. The only request is that the noise has to be cancelled by the filter. Therefore, there is no need to analyze the effect of the filter on $m(t)$. We know that $\eta(t)$ is a sinusoid for which the frequency is known but amplitude and phase are unknown, i.e. $\eta(t) = P\sin(2\pi ft + \phi)$. After sampling, the noise term is $\bar{\eta}(n) = Psin(2\pi f_d n + \phi)$, where $f_d = fT_s = 1/80$. We see that the signal $\bar{\eta}(n)$ is periodic and the period is $N_d = 1/f_d = 80$. The moving average computes the average of $\bar{\eta}(n)$ on the set $[n-L+1, n]$ for all the values $n \in \mathbb{Z}$. Therefore, to zero mean for all the positions n, L has to be a multiple of the period 80. All the multiples are possible to cancel the noise, but the effect on the signal $m(t)$ is different. As we saw in the lectures, large values of L will reduce the high frequencies of $m(t)$ (the same effect that you have by setting to zero the treble knob of a Hi-Fi chain). For the second filter, we consider again only the term $\eta(t)$ and we obtain:

$$\begin{aligned} y(n) &= \bar{\eta}(n) + a_1\bar{\eta}(n-1) + a_2\bar{\eta}(n-2) \\ &= P\sin(2\pi f_d n + \phi) + a_1 P\sin(2\pi f_d(n-1) + \phi) + a_2 P\sin(2\pi f_d(n-2) + \phi) \\ &= P\sin(2\pi f_d n + \phi)(1 + a_1\cos(2\pi f_d) + a_2\cos(4\pi f_d)) \\ &\quad + P\cos(2\pi f_d n + \phi)(-a_1\sin(2\pi f_d) - a_2\sin(4\pi f_d)). \end{aligned}$$

The last equation has been obtained by applying trigonometric equalities. To have $y(n) = 0,\ \forall n \in \mathbb{Z}$, we need that

$$\begin{cases} \cos(2\pi f_d)a_1 + \cos(4\pi f_d)a_2 &= -1 \\ \sin(2\pi f_d)a_1 + \sin(4\pi f_d)a_2 &= 0, \end{cases}$$

which correspond to a linear system of equations. The solution can be computed by multiplying the two equations by $\sin(2\pi f_d)$ and $\cos(2\pi f_d)$ and taking the difference. This gives, $a_2 = 1$. By substitution of a_2 in one of the two equations, we find $a_1 = -2\cos(2\pi f_d)$.

11. (a) As we saw in the lectures, the moving average of length $L = 4$ is computed in the following way:
$y(3) = \frac{g(0)+g(1)+g(2)+g(3)}{4} = 4.2$ $\quad y(4) = \frac{g(1)+g(2)+g(3)+g(4)}{4} = 4.7$
$y(5) = \frac{g(2)+g(3)+g(4)+g(5)}{4} = 5$

(b) Since the moving average is a linear system, we can treat separately the signals $s(n)$ and $e(n)$. We want to have at the output only the moving average of $s(n)$, hence the moving average of $e(n)$ has to be zero. The moving average compute the average on 4 samples, thus to have zero at the output, we have to choose $e(n)$ periodic of period 4 and such the the average is zero on one period. For examples, we can choose

$$e(n) = \begin{cases} 4 & n = 0, \pm 4, \pm 8, \ldots \\ 1, & n = 1, 1 \pm 4, 1 \pm 8, \ldots \\ -3, & n = 2, 2 \pm 4, 2 \pm 8, \ldots \\ -2, & n = 3, 3 \pm 4, 3 \pm 8, \ldots . \end{cases}$$

12. (a) For the definition of pulse function $\delta(n)$, we have that

$$h(n) = \begin{cases} 0 & n < 0 \text{ et } n > 2 \\ 2 & n = 0, \\ -1 & n = 1, \\ -1 & n = 2, \end{cases}$$

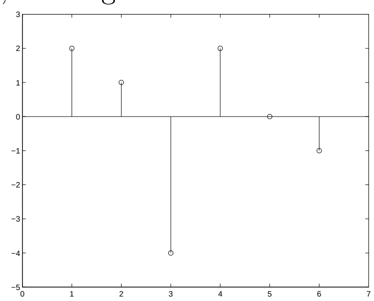
which correspond to a causal impulse response, since $h(n) = 0$ when $n < 0$.

(b) The system is stable, since

$$\sum_{n=-\infty}^{\infty} |h(n)| = 2 + 1 + 1 < +\infty.$$

One could have also remarked that the impulse response has finite duration (FIR) and remember that the FIR responses are always stable.

(c) We have to compute the convolution between $x(n)$ and $h(n)$. We can use the graphic method presented in the lectures, which gives



1.5.3 Solutions to the exercises of section 1.3

TO BE DONE.

1.5.4 Solutions to the exercises of section 1.4

1. (a) The frequency of the sinusoid is $f = 100$ Hz, so to apply the sampling theorem, we have to choose $f_S > 200$ Hz.

   (b) f_S has to be *strictly* larger than 200 Hz. Otherwise, we would have $\bar{x}(n) = 0$ which corresponds to a sinusoid of frequency $f = 0$ Hz.

2. (a) We impose that $\cos(\pi n/4) = \cos(2\pi f_P nT)$, where f_P is the perceived frequency, which differs in general from the real frequency f. As we saw in the lectures, we have

$$\pi n/4 = 2\pi f_P nT + N2\pi, \quad n \in \mathbb{Z},$$

   and the solution is

$$f_P = 1/8 f_S + N f_S.$$

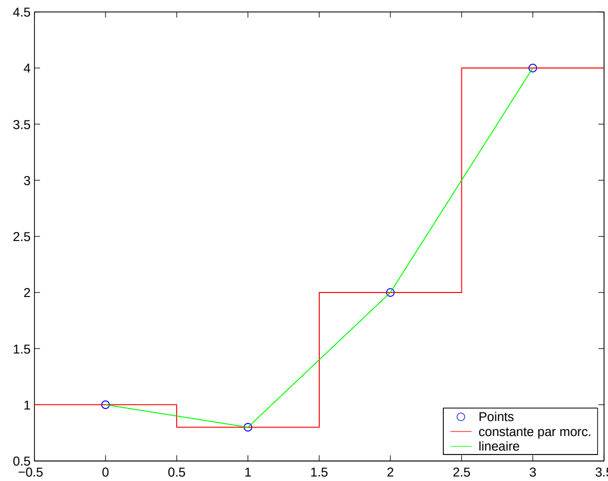
   Two possible values of f_P are for example 125 Hz and 1125 Hz, or -875 Hz and 5125 Hz.

   (b) We saw the ideal interpolator gives, among all the possible values of f_P, the one that in absolute value is smaller than $f_S/2$. In this case, we have $f = f_S/8 = 125$ Hz. The reconstructed signal is

$$y(t) = \cos(2\pi f t) = \cos(250\pi t).$$

3. The signal $x(t)$ is composed by two sinusoids. We defined the bandwidth B as the maximum frequency of the sinusoids that compose the signal, i.e. $B = 20$ Hz. To apply the theorem of Shannon-Nyquist we must choose $f_S > 2B$, hence $f_S > 40$ Hz.

4. (a) The points, the piecewise-constant interpolation and the linear interpolation are shown on the following figure. suivante:



(b) We have

$$y(t) = \mathrm{rect}(t) + 0.8\mathrm{rect}(t-1) + 2\mathrm{rect}(t-2) + 4\mathrm{rect}(t-3),$$

(c) and

$$y(t) = \mathrm{triang}(t) + 0.8\mathrm{triang}(t-1) + 2\mathrm{triang}(t-2) + 4\mathrm{triang}(t-3).$$

(d) As we saw in the lectures, we write

$$L_0(t) = \frac{(t-1)(t-2)(t-3)}{(-1)(-2)(-3)} = \frac{-t^3+6t^2-11t+6}{6},$$

$$L_1(t) = \frac{t(t-1)(t-3)}{1(1-2)(1-3)} = \frac{t^3-5^2+6t}{2},$$

$$L_2(t) = \frac{t(t-1)(t-3)}{2(2-1)(2-3)} = \frac{-t^3+4t^2-3t}{2},$$

$$L_3(t) = \frac{t(t-1)(t-2)}{3(3-1)(3-2)} = \frac{t^3-3t^2+2t}{6}.$$

The expression of the Lagrange interpolator is

$$y(t) = L_0(t) + 0.8L_1(t) + 2L_2(t) + 4L_3(t).$$

5. (a) The two sampled signals are:

$$\begin{array}{l} \sin(2\pi \frac{f}{f_{S1}} n) = 0 \\ \sin(2\pi \frac{f}{f_{S2}} n) = 0 \end{array} \qquad \forall n \in \mathbb{Z}.$$

We recall that the sine is zero on all the multiples of π, so

$$\begin{cases} \frac{f}{f_{S1}} = \frac{k_1}{2} \\ \frac{f}{f_{S2}} = \frac{k_2}{2} \end{cases} \qquad k_1, k_2 \in \mathbb{Z}.$$

To satisfy both the equations *at the same time*, f has to be multiple of both $f_{S1}/2$ and $f_{S2}/2$, hence it has to be multiple of the Least Common Multiple (LCM)

$$f = k \text{ lcm}\left(\frac{f_{S1}}{2}, \frac{f_{S2}}{2}\right) = k30 \qquad k \in \mathbb{Z}.$$

(b) If we had sampled only once at the sampling frequency $f_S = 60$ Hz, we would have obtained in the same way that $f_S/2 = k30$ Hz.

6. (a) The sinusoid frequency is $f = 60$ Hz, hence to apply the sampling theorem, we have to choose $f_S > 120$ Hz.

   (b) f_S has to be *strictly* larger than 120 Hz. Otherwise, we would have $\bar{x}(n) = 0$. We remind that the ideal interpolator reconstructs, among all the sinusoids compatible with the sampled signal, the sinusoid with minimum frequency. In the case of this exercise, the possible frequencies are $f = k120$ Hz, and the interpolator reconstruct a sinusoid at the frequency of 0 Hz,

$$x(t) = \sin(2\pi 0t) = 0.$$

7. (a) We impose $\cos(\pi n/4) = \cos(2\pi f_P nT)$, where f_P is the perceived frequency, which in general differs from the actual frequency f. As seen in the lectures,

$$\pi n/4 = 2\pi f_P nT + N2\pi, \quad N, n \in \mathbb{Z},$$

   and the solution is

$$f_P = 1/8 f_S + M f_S, \quad M \in \mathbb{Z}.$$

   Two values of f_P are for example, 125 Hz and 1125 Hz, or -875 Hz and 5125 Hz.

   (b) The ideal interpolator gives, among all the possible values of the freqeucny f_P, that that is smaller than $f_S/2$. In this case, we have $f_P = f_S/8 = 125$ Hz. The reconstructed signal is

$$y(t) = \cos(2\pi f_P t) = \cos(250\pi t).$$

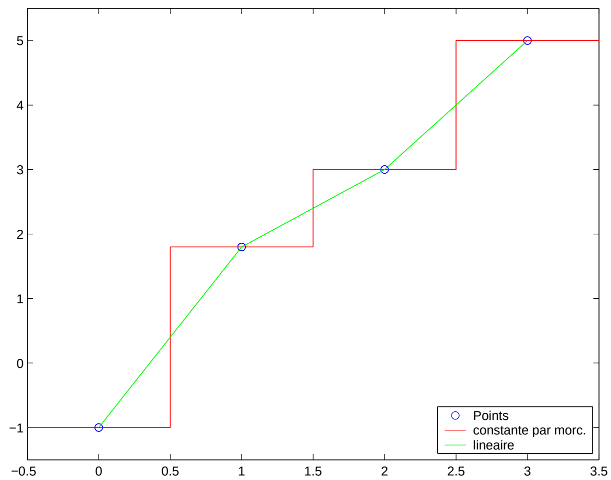
8. The signal $x(t)$ is composed by two sinusoids. In the lecture, the bandwidth B was defined as the maximum frequency of the sinusoids that compose a signal, hence $B = 25$ Hz. To avoid aliasing, we must choose $f_S > 2B$, therefore $f_S > 50$ Hz. If we sample at 40Hz and we use the ideal interpolator, the component at 12.5 Hz is reconstructed exactly, since its frequency is lower than the Nyquist frequency which is 20 Hz. On the other hand, the second component has frequency higher than the Nyquist frequency and is affected by aliasing. The possible frequencies that correspond to the samples of the second component are

$$f_{P2} = 25 + k40 \quad k \in \mathbb{Z}.$$

The ideal interpolator reconstructs the sinusoid corresponding to the frequency that in absolute value is smaller than the Nyquist frequency. In this case, we have that this case corresponds to $k = -1$ and $f_{P2} = -15$ Hz. In conclusion, the reconstructed signal is

$$y(t) = \sin(25\pi t) + \cos(2\pi f_{P2} t) = \sin(25\pi t) + \cos(-30\pi t) = \sin(25\pi t) + \cos(30\pi t).$$

9. (a) The points, the piecewise-constant and the linear interpolation are shown in the following figure:



(b) We have

$$y(t) = -\text{rect}(t) + 1.8\text{rect}(t-1) + 3\text{rect}(t-2) + 5\text{rect}(t-3),$$

(c) and

$$y(t) = -\text{triang}(t) + 1.8\text{triang}(t-1) + 3\text{triang}(t-2) + 5\text{triang}(t-3).$$

(d) As shown in the lecture notes, we write

$$L_0(t) = \frac{(t-1)(t-2)(t-3)}{(-1)(-2)(-3)} = \frac{-t^3 + 6t^2 - 11t + 6}{6},$$

$$L_1(t) = \frac{t(t-1)(t-3)}{1(1-2)(1-3)} = \frac{t^3 - 5^2 + 6t}{2},$$

$$L_2(t) = \frac{t(t-1)(t-3)}{2(2-1)(2-3)} = \frac{-t^3 + 4t^2 - 3t}{2},$$

$$L_3(t) = \frac{t(t-1)(t-2)}{3(3-1)(3-2)} = \frac{t^3 - 3t^2 + 2t}{6}.$$

The Lagrange interpolation is

$$y(t) = -L_0(t) + 1.8L_1(t) + 3L_2(t) + 5L_3(t).$$