Chapter 1

Recap on Digital Signal Processing

Signal processing consists of handling data in order to extract information considered relevant, or to modify them so as to give them useful properties: extracting, for example, information on a plane's speed or distance from a RADAR signal, making an old and decayed sound recording clearer, synthesizing a sentence on an answering machine, transmitting information through a communication channel, etc.

The processing is called *digital* if it deals with a discrete sequence of values $\{x_1, x_2, \ldots\}$. There are two types of scenario: either the observation is already a sequence of numbers, as is the case for example for economic data, either the observed phenomenon is "continuous-time", and the signal's value $x(t)$ must then be measured at regular intervals.

This second scenario has tremendous practical applications. This is why an entire section of this chapter is devoted to the operation called *sampling*.

The acquisition chain is described in Figure 1.1.

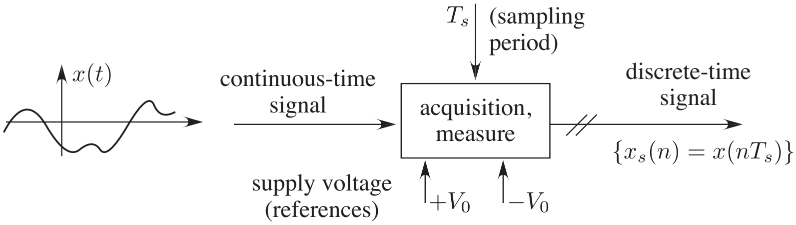

Figure 1.1 – *Digital signal acquisition*

The essential part of the acquisition device is usually the *analog-to-digital converter*, or *ADC*, which samples the value of the input voltage at regular intervals – every T_s seconds – and provides a coded representation at the output.

To be absolutely correct, this coded value is not exactly equal to the value of $x(nT_s)$. However, in the course of this chapter, we will assume that $x_s(n) = x(nT_s)$. The sequence of these numerical values will be referred to as the *digital signal*, or more plainly as the *signal*.

T_s is called the *sampling period* and $F_s = 1/T_s$ the *sampling frequency*. The gap between the actual value and the coded value is called *quantization noise*.

Obviously, the sampling frequency must be high enough "in order not to lose too much information" – a concept we will discuss later on – from the original signal, and there is a connection between this frequency and the sampled signal's "frequential content". Anybody who conducts experiments knows this "graph plotting principle": when the signal's value changes quickly (presence of "high frequencies"), "many" points have to be plotted (it would actually be preferable to use the phrase "high point density"), whereas when the signal's value changes slowly (presence of low frequencies), fewer points need to be plotted.

To sum up, the signal sampling must be done in such a way that the numerical sequence $\{x_s(n)\}$ alone is enough to reconstruct the continuous-time signal. The sampling theorem specifies the conditions that need to be met for perfect reconstruction to be possible.

1.1 The sampling theorem

Let $x(t)$ be a continuous signal, with $X(F)$ its Fourier transform, which will also be called the *spectrum*. The sample sequence measured at the frequency $F_s = 1/T_s$ is denoted by $x_s(n) = x(nT_s)$.

Definition 1.1 *When $X(F) \neq 0$ for $F \in (B_1, B_2)$ and $X(F) = 0$ everywhere else, $x(t)$ is said to be (B_1, B_2) band-limited. If $x(t)$ is real, its Fourier transform has a property called* Hermitian symmetry*, meaning that $X(F) = X^*(-F)$, and the frequency band's expression is $(-B, +B)$. A common misuse of language consists of referring to the signal as a B-band signal.*

Perfect reconstruction

Our goal is to reconstruct $x(t)$, at every time t, using the sampling sequence $x_s(n) = x(nT_s)$, while imposing a "reconstruction scheme" defined by the expression (1.1):

$$y(t) = \sum_{n=-\infty}^{+\infty} x(nT_s)h(t - nT_s) \tag{1.1}$$

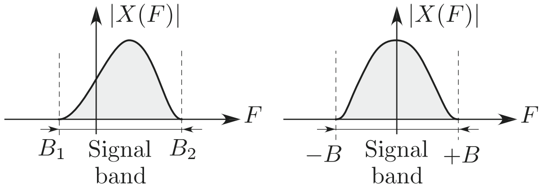

Figure 1.2 – *Spectra of band-limited signals*

where $h(t)$ is called a *reconstruction function.* Notice that (1.1) is linear with respect to $x(nT_s)$. In order to reach this objective, two questions have to be answered:

1. is there a class of signals $x(t)$ large enough for $y(t)$ to be identical to $x(t)$?

2. if that is the case, what is the expression of $h(t)$?

The answers to these questions are provided by the sampling theorem (1.1).

Theorem 1.1 (Sampling theorem)
Let $x(t)$ be a (B_1, B_2) band-limited signal, real or complex, and let $\{x(nT_s)\}$ be its sample sequence, then there are two possible cases:

1. *If $F_s = 1/T_s$ is such that $F_s \geq B_2 - B_1$, then $x(t)$ can be perfectly reconstructed from its samples $x(nT_s)$ using the expression:*

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT_s) h_{(B_1,B_2)}(t - nT_s) \tag{1.2}$$

where the FT of the reconstruction function $h_{(B_1,B_2)}(t)$ is:

$$H_{(B_1,B_2)}(F) = \frac{1}{F_s}\ \mathbf{1}(F \in (B_1, B_2)) \tag{1.3}$$

2. *If $F_s = 1/T_s < B_2 - B_1$, perfect reconstruction turns out to be impossible because of the "spectrum aliasing" phenomenon.*

The proof uses the *Poisson summation formula* which gives the relation between $X(F)$ and the values of $x(t)$ at sampling times nT_s, and makes it possible to determine the expression of the spectrum of the signal $y(t)$ defined by equation (1.1).

Lemma 1.1 (Poisson formula) *Let $x(t)$ be a signal, and $X(F)$ its Fourier transform. Then for any T_s:*

$$\frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X(F-kF_s) = \sum_{n=-\infty}^{+\infty} x(nT_s)\exp(-2j\pi nFT_s) \tag{1.4}$$

where the left member is assumed to be a continuous function of F.

We will use the following definition for the *discrete-time Fourier transform*. We will see another completely equivalent expression of it (definition 1.3), but more frequently used in the case of numerical sequences.

Definition 1.2 (DTFT) *The sum $\sum_{n=-\infty}^{+\infty} x(nT_s)\exp(-2j\pi nFT_s)$ is called the Discrete-Time Fourier Transform (DTFT) of the sequence $\{x(nT_s)\}$.*

We now go back to the sampling theorem. By using the fact that the Fourier transform of $h(t-nT_s)$ is $H(F)e^{-2j\pi nFT_s}$, the Fourier transform of $y(t)$, defined by (1.1), can be written:

$$\begin{aligned} Y(F) &= \sum_{n=-\infty}^{+\infty} x(nT_s)\times H(F)e^{-2j\pi nFT_s} \\ &= \frac{H(F)}{T_s}\sum_{k=-\infty}^{+\infty} X(F-kF_s) \end{aligned} \tag{1.5}$$

Therefore, if $F_s \geq B_2 - B_1$, the different contributions $X(F-kF_s)$ do not overlap, and by simply assuming $H_{(B_1,B_2)}(F) = T_s\mathbf{1}(F\in(B_1,B_2))$, $Y(F)$ coincides exactly with $X(F)$. Figure 1.3 illustrates this case for a real signal. In this case, $B_1 = -B$ and $B_2 = B$.

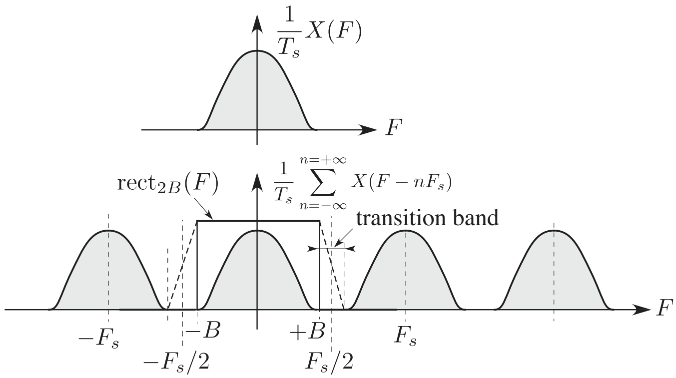

Figure 1.3 – *Real signal reconstruction*

Except if specified otherwise, we will assume from now on that $x(t)$ is real. The *sufficient* reconstruction condition can be written as follows:

$$F_s \geq 2B \tag{1.6}$$

The limit frequency $2B$ is called the *Nyquist frequency*. Still in the same case, the Fourier transform of a possible reconstruction function is $H_B(F) = T_s \mathrm{rect}_{2B}(F)$, and therefore:

$$h_B(t) = \frac{\sin(2\pi Bt)}{\pi F_s t} \tag{1.7}$$

It should be noted that the function $H_B(F) = T_s \mathrm{rect}_{2B}(F)$ is not the only possible function. If F_s is assumed to be strictly greater than $2B$, then we can choose a filter with larger transition bands (see Figure 1.3), making it easier to design.

When there is no possible doubt, we will not indicate the dependence on B, and simply write $h(t)$ instead of $h_B(t)$.

Anti-aliasing filter

The reconstruction formula (1.1), is, according to the Poisson formula (1.4), associated with the periodization of the spectrum $X(F)$ with the period F_s. It follows that, for $F_s < 2B$, the different non-zero parts of the spectrum overlap, making perfect reconstruction impossible. The overlapping phenomenon is called *spectrum aliasing*.

Figure 1.4 illustrates the spectrum aliasing phenomenon for a real signal whose frequential content is of the "low-pass" type, implicitly meaning that it "fills up" the band $(-F_s/2, +F_s/2)$.

Except in some particular cases, we will assume that spectrum signals are of this type, or that they can be modified to fit this description.

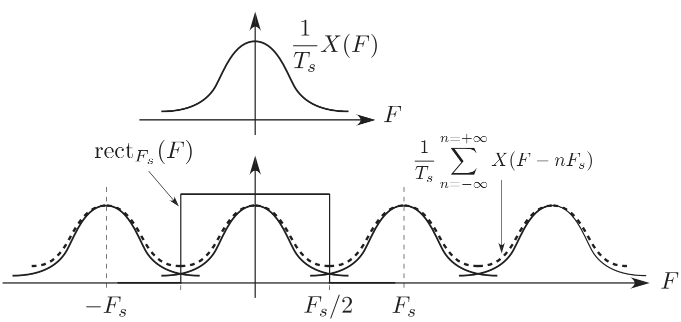

Figure 1.4 – *The aliasing phenomenon*

For a real signal, showing aliasing means that the frequencies beyond the frequency $F_s/2$ can be "brought back" to the $(-F_s/2, +F_s/2)$ band.

In practice, the following cases will occur:

1. the sampling frequency is imposed: if, knowing how the data is used, the aliasing phenomenon is considered to "cause damage", the appropriate procedure for sampling a real signal requires the use of low-pass filtering called *anti-aliasing filtering* which eliminates the components of the frequencies higher than $F_s/2$;

2. the sampling frequency is not imposed: in this case, it can be chosen high enough so that the aliased components of the signal do not alter the expected results. If this is not possible, F_s is set, and the situation becomes the same as in the first case.

Speech signals are a good example. If they are sampled at 8,000 Hz, an extremely common value, high enough to make the person speaking recognizable and understandable, and if no anti-aliasing filtering is done, the reconstructed signal contains a "hissing" noise. This alone justifies the use of an anti-aliasing filter. The irretrievable loss of high frequency components is actually better than the presence of aliasing.

Figure 1.5 illustrates the case of a "low-pass", prefiltered, real signal to prevent aliasing.

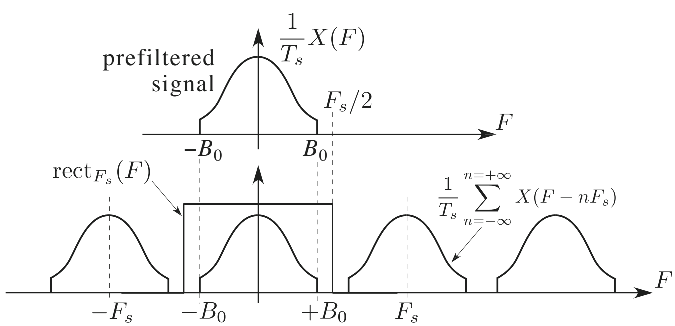

Figure 1.5 – *Absence of aliasing after* $[-B_0, +B_0]$ *filtering*

In general, it is important to understand that anti-aliasing filtering must be done in the band that is considered essential (*useful band*) to the unaliased signal reconstruction. The low-pass filtering mentioned here corresponds to a low-pass sampled signal.

The following general rule can be stated:

> The sampling operation of a signal at the frequency F_s must be preceded by an anti-aliasing filtering with a gain equal to 1 and with a width of F_s in the useful band.

Spectrum aliasing and ambiguity

For a given signal, for any integer k, it is not possible to distinguish F_0 from $F_1 = F_0 + kF_s$, $k \in \mathbb{Z}$, which is called the *image frequency* of F_0 relative to F_s. Hence, $x_1(t) = \sin(2\pi F_0 t)$ and $x_2(t) = \sin(2\pi(F_0 + kF_s)t)$, with $k \in \mathbb{Z}$ take exactly the same values if both are collected at frequency F_s. This is the *ambiguity* due to the spectrum aliasing phenomenon (or generally speaking to the Poisson formula).

1.2 Spectral contents

1.2.1 Discrete-time Fourier transform (DTFT)

The sampling period T_s appears in the DTFT's expression in definition 1.3.

Definition 1.3 (DTFT) *The discrete-time Fourier transform of a sequence $\{x(n)\}$ is the function of the real variable f, periodic with period 1, defined by:*

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n)\exp(-2j\pi nf) \tag{1.8}$$

As you can see, we need only impose $FT_s = f$ and replace $x(nT_s)$ by $x(n)$ to go from (1.4) to(1.8)[(1)].

Definition (1.3) calls for a few comments: it can be proven that if $\{x(n)\}$ is summable ($\sum_n |x(n)| < +\infty$), the series (1.8) converges uniformly to a continuous function $X(f)$. However, if $\{x(n)\}$ is square summable ($\sum_n |x(n)|^2 < +\infty$) without having a summable modulus, then the series converges in quadratic mean. There can be no uniform convergence.

Because of its periodicity, the DTFT is plotted on an interval of length 1, most often the intervals $(-1/2, +1/2)$ or $(0, 1)$.

Starting off from $X(f)$, how can we go back to $x(n)$? One possible answer is given in the following result.

Theorem 1.2 (Inverse DTFT) *If $X(f)$ is a periodic function with period 1, and if $\int_0^1 |X(f)|^2 df < +\infty$, then $X(f) = \sum_n x(n)e^{-2j\pi nf}$, where the $x(n)$ coefficients are given by:*

$$x(n) = \int_{-1/2}^{1/2} X(f)e^{2j\pi nf} df \tag{1.9}$$

[(1)] $X(F)$, which refers to the FT in (1.4) must not be confused with $X(f)$, the DTFT.

As in the continuous-time case, we have the Parseval formula:

$$\sum_{n=-\infty}^{+\infty} |x(n)|^2 = \int_{-1/2}^{1/2} |X(f)|^2 df \tag{1.10}$$

and the conservation of the dot product:

$$\sum_{n=-\infty}^{+\infty} x(n)y^*(n) = \int_{-1/2}^{1/2} X(f)Y^*(f) df \tag{1.11}$$

Because the left member of (1.10) is, by definition, the signal's energy, $|X(f)|^2$ represents the energy's distribution along the frequency axis. It is therefore called the *energy spectral density* (*esd*), or *spectrum*. In the literature, this last word is associated with the function $|X(f)|$. If $X(f)$ is included, this adds up to three definitions for the same word. But in practice, this is not important, as the context is often enough to clear up any ambiguity. It should be pointed out that the two expressions $|X(f)|$ and $|X(f)|^2$ become proportional if the *decibel* scale is used, by imposing:

$$S_{dB}(f) = 20 \log_{10} |X(f)| \tag{1.12}$$

1.2.2 Discrete Fourier transform (DFT)

Definition of the discrete Fourier transform

A computer calculation of the DTFT, based on the values of the samples $x(n)$, imposes an infinite workload, because the sequence is made up of an infinity of terms, and because the frequency f varies continuously on the interval $(0, 1)$. This is why, digitally speaking, the DTFT does not stand a chance against the *Discrete Fourier Transform*, or *DFT*. The DFT calculation is limited to a finite number of values of n, and a finite number of values of f.

The digital use of the DFT has acquired an enormous and undisputed practical importance with the discovery of a fast calculation method known as the *Fast Fourier Transform*, or *FFT*.

Consider the finite sequence $\{x(0), \ldots, x(P-1)\}$. Using definition (1.8), its DTFT is expressed $X(f) = \sum_{n=0}^{P-1} x(n)e^{-2j\pi nf}$ where $f \in (0, 1)$. In order to obtain the values of $X(f)$ using a calculator, only a finite number N of values for f are taken. The first idea that comes to mind is to take N values, uniformly spaced-out in $[0, 1[$, meaning that $f = k/N$ with $k \in \{0, \ldots, N-1\}$. This gives us the N values:

$$X(k/N) = \sum_{n=0}^{P-1} x(n)e^{-2j\pi nk/N} \tag{1.13}$$

In this expression, P and N play two very different roles: N is the number of points used to calculate the DTFT, and P is the number of observed points of the temporal sequence. N influences the *precision* of the plotting of $X(f)$, whereas P is related to what is called the *frequency resolution*.

In practice, P and N are chosen so that $N \geq P$. We then impose:

$$\tilde{x}(n) = \begin{cases} x(n) & \text{for } n \in \{0, \ldots, P-1\} \\ 0 & \text{for } n \in \{P, \ldots, N-1\} \end{cases}$$

Obviously:

$$X(k/N) = \sum_{n=0}^{P-1} x(n) e^{-2j\pi nk/N} = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-2j\pi nk/N}$$

Because the sequence $x(n)$ is completed with $(N-P)$ zeros, an operation called *zero-padding*, in the end we have as many points for the sequence $\tilde{x}(n)$ as we do for $X(k/N)$. Choosing to take as many points for both the temporal sequence and the frequential sequence does not restrict in any way the concepts we are trying to explain. This leads to the definition of the *discrete Fourier transform*.

Definition 1.4 *Let $\{x(n)\}$ be a N-length sequence. Its discrete Fourier transform or DFT is defined by:*

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad k \in (0, 1, \ldots N-1) \tag{1.14}$$

$$\textit{where } W_N = e^{-2j\pi/N} \tag{1.15}$$

is an N-th root of unity, that is to say such that $W_N^N = 1$. The inverse formula, leading from the sequence $\{X(k)\}$ to the sequence $\{x(n)\}$, is:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \tag{1.16}$$

Properties of the DFT

The properties of the DFT show strong similarities with those of the DTFT. However, there is an essential difference. In the formulas associated with the DFT, *all the index calculations are done modulo N.*

Fast Fourier transform

The *fast Fourier transform*, or *FFT*, first published in 1965 by J. W. Cooley and J. W. Tuckey [8], is a fast DFT calculation technique. The basic algorithm, many versions of which can be found, calculates a number of points N, equal to a power of 2, and the time saved compared with a direct calculation is roughly:

$$\text{gain} = \frac{N}{\log_2(N)}$$

To get a better idea, if $N = 1{,}024$, the FFT is about *100 times faster* than the direct calculation based on the definition of the DFT.

1.3 Case of random signals

In the case of a random process, also referred to as a "time series", the notion of "spectral content" needs to be treated with caution. More specifically, the Fourier transform of a single random process trajectory generally does not exist. However, theoretical developments [3] lead us to define the *spectrum* associated with a random process by the DTFT of the covariance function:

$$S_{XX}(f) = \text{DTFT}\left(R_{XX}(n)\right) = \sum_{n=-\infty}^{+\infty} R_{XX}(n) e^{-2\pi jnf} \tag{1.17}$$

$S_{XX}(f)$ is called the *power spectral density* (*psd*).

A fairly simple alternative idea is to use the Fourier transform of that single record of length N. This leads us to the definition of a *periodogram*. A periodogram is the random function of $f \in (0,1)$ defined by:

$$I_N(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} X(n) e^{-2j\pi nf} \right|^2 \tag{1.18}$$

Unfortunately the periodogram is not a good estimator of the spectral density [3]. However a consistent estimate can be derived by *averaging* or *smoothing* periodograms.

The filtering equations also exhibit a fundamental difference in comparison to the deterministic case ($Y(f) = H(f)X(f)$). In the random case, the psds are linked by the relation:

$$S_{YY}(f) = |H(f)|^2 S_{XX}(f) \tag{1.19}$$

Note that there is a second filtering equation which can be used:

$$S_{YX}(f) = H(f) S_{XX}(f) \tag{1.20}$$

where $S_{YX}(f)$, which is called the *interspectrum*, refers to the Fourier transform of $\{R_{YX}(n)\}$.

1.4 Example of the Dual Tone Multi-Frequency (DTMF)

On a Dual Tone Multi-Frequency (DTMF) phone keyboard, each key is associated with the sending of a signal. This signal is the sum of two sines the frequencies (in Hz) of which are given in the correspondence Table 1.1.

Hz	1,209	1,336	1,477
697	1	2	3
770	4	5	6
852	7	8	9
941	$\star$	0	#

Table 1.1 – *Frequency correspondence table*

This means that when you dial “5” on your phone, the signal $x(t) = \cos(2\pi \times 1{,}336 \times t) + \cos(2\pi \times 770 \times t)$ is sent through the phone line.

COMMENTS:

– these frequencies belong to the (300 Hz - 3,400 Hz) band, the phone band for the switched network (fixed phones). The frequencies associated with the columns are all greater than those associated with the lines. This layout can help to find the phone number using the signal. Finally, the frequencies were chosen so as not to have frequency ratios equal to integers. As we have seen, a non-linear operation can cause multiples (harmonics) of the fundamental frequency to appear, causing some confusion;

– the keyboard is designed to always send signals for periods longer than $\tau_1 = 65$ milliseconds. This value was chosen so that the two frequencies contained in the signal could easily be separated. In the worst case, the difference in frequency is $\Delta F_{\min} = 1{,}209 - 941 = 268$ Hz (corresponding to the $\star$ key), therefore τ_1 needs to be such that $\Delta F_{\min}\tau_1 \gg 1$. With the values that were chosen, $\Delta F_{\min}\tau_1 > 17$;

– finally, it must be possible to tell the difference between the number C being sent for a duration of T and the number CC being sent for the same duration. This is why, after each key is released, a zero signal is sent for at least 80 ms (even if you can dial faster than that!).

Exercise 1.1 (DTMF signal processing) (see p. 187)
We are going to try to find a 10 digit phone number using the signal sent

by the phone. We will start by sampling the signal at a frequency of 8,000 samples per second, a speed much higher than twice the highest frequency, that is $2 \times 1{,}477 = 2{,}954$ Hz.

The following program creates such a signal:

```
%===== genekey.m
clear
Fs=8000;              % sampling freq.
tel='0145817178'; lt=length(tel); % seq. of numbers
%===== coding table
keys='123456789*0#'; nbkeys=length(keys);
FreqB=[697 770 852 941]; FreqH=[1209 1336 1477];
Freqskeys=...
   [FreqB(1) FreqH(1); FreqB(1) FreqH(2); % 1 et 2
    FreqB(1) FreqH(3); FreqB(2) FreqH(1); % 3 et 4
    FreqB(2) FreqH(2); FreqB(2) FreqH(3); % 5 et 6
    FreqB(3) FreqH(1); FreqB(3) FreqH(2); % 7 et 8
    FreqB(3) FreqH(3); FreqB(4) FreqH(1); % 9 et *
    FreqB(4) FreqH(2); FreqB(4) FreqH(3)];% 0 et #
%===== constraints
tton=0.065; tsil=0.080;            % in seconds
%==== construction of the seq. of frequencies
Freqs=zeros(lt,2);
for k=1:lt
    ind=find(keys==tel(k));        % test of the number
    Freqs(k,:)=Freqskeys(ind,:);   % associated Freq.
end
freqs=Freqs/Fs;                    % normalized freq.
%===== construction of the signal
y=zeros(100+fix(100*rand),1);      % starting with signal=0
dton=fix(1000*rand(lt,1)+tton*Fs); % number duration
dsil=fix(1000*rand(lt,1)+tsil*Fs); % silence duration
for k=1:lt
    sigu=cos(2*pi*(0:dton(k))'*freqs(k,:))*ones(2,1);
    y=[y;sigu;zeros(dsil(k),1)];
end
%===== some noise is added
lx=length(y); py=y'*y/lx;
SNR=30; % signal to noise ratio
pb=py*10^(-SNR/10); x=y+sqrt(pb)*randn(lx,1);
%===== plotting of the signal
tps=(0:lx-1)/Fs; plot(tps,x); grid
set(gca,'xlim',[0 (lx-1)/Fs])
```

In order to simulate the perturbations on an actual phone call, the program adds noise created by `sqrt(pb)*randn(L,1)`. `SNR` is the signal-to-noise ratio (in dB) chosen equal to 30 dB. The resulting signal is shown in Figure 1.6.

We are going to find the 10 digit number in this signal in two steps. First, we will determine the beginning and the end of the signal's active zones, then

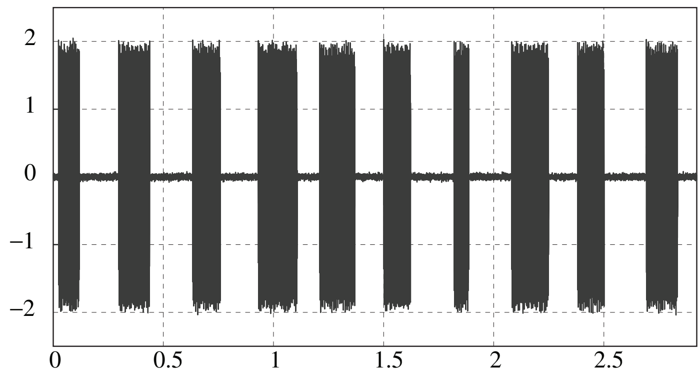

Figure 1.6 – *DTMF Signal*

we will analyze each of the intervals to extract the frequencies and therefore the corresponding digit. To determine the beginning and the end of the active zones of the signal, we are going to estimate the "instantaneous power" and compare it to a threshold value. We will see later on as we study random phenomena what we mean exactly by "estimating the instantaneous power". Here, we will merely be considering the quantity:

$$P_n = \frac{1}{N} \sum_{k=n-N+1}^{n} x_k^2 \tag{1.21}$$

which gives a relevant indication on the signal's fluctuations. The choice of N is done as a compromise. Consider, for example,the signal $x(n)$ represented in Figure 1.6. If N is very small, P_n will be very close to the amplitude x_n^2. The risk would be to make the conclusion that the power is equal to zero whenever the amplitude is close to 0 (which happens every period). If, on the contrary, N is very high, we might include a silence and miss the beginning or the end of an active part. Quantitatively, N must therefore be much greater than the longest of the periods of the active parts, and much smaller than the duration of the wanted signal, that is to say 65 ms. This can be expressed:

$$\frac{F_s}{697} \ll N \ll 65 \times 10^{-3} F_s$$

For $F_s = 8{,}000$, and with $N = 100$, this double inequality is satisfied.

1. write a program that measures the "instantaneous power" and determines the beginning and the end of the signals associated with a digit;

2. write a program that determines the digit associated with each portion of the signal.