Topic 4:
Sampling and analog-digital conversion (ADC)

Aims:

The aim of this TA is to understand:

- the fundamentals of **signal sampling** and the basic rules of signal reconstruction,
- the main **features of an ADC**.

Prerequisites:

Good knowledge of time-frequency representations (spectra) of signals. RC circuits and operational amplifiers.

The main aim of analog-digital conversion (ADC) or digital-analog conversion (DAC) is to match a binary number N to an analog voltage V and vice versa. Input information must be presented to the digital system (e.g. PC) in the binary form EXCLUSIVELY and the same for the resulting information which will be delivered in this form. Traditionally, an acquisition and processing chain consists of the following parts:

- sensor-based information input mostly delivering analog-type magnitude,
- real-time or deferred-time data processing with complex calculation algorithms that cannot be carried out analogically (e.g. digital filtering, DSP calculations, coding, etc.),
- information output based on actuators that need to be controlled on an analog scale.

The diagram in Figure 1 shows the positions of the ADC and DAC within an electronic system.

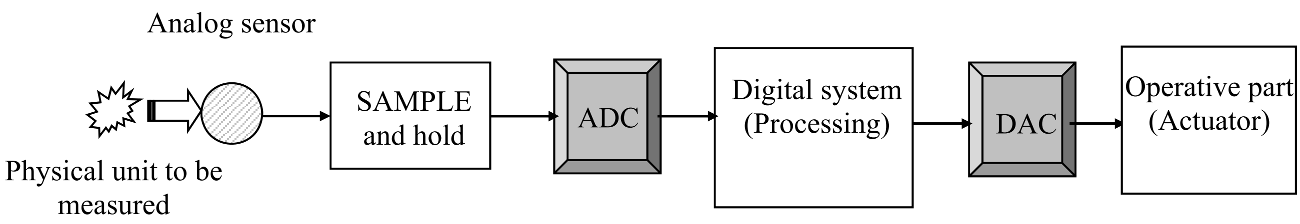


Figure 1: *Block diagram of electronic system processing information*

Sampling is the first stage when carrying out an analog-digital conversion, as shown in Figure 2.

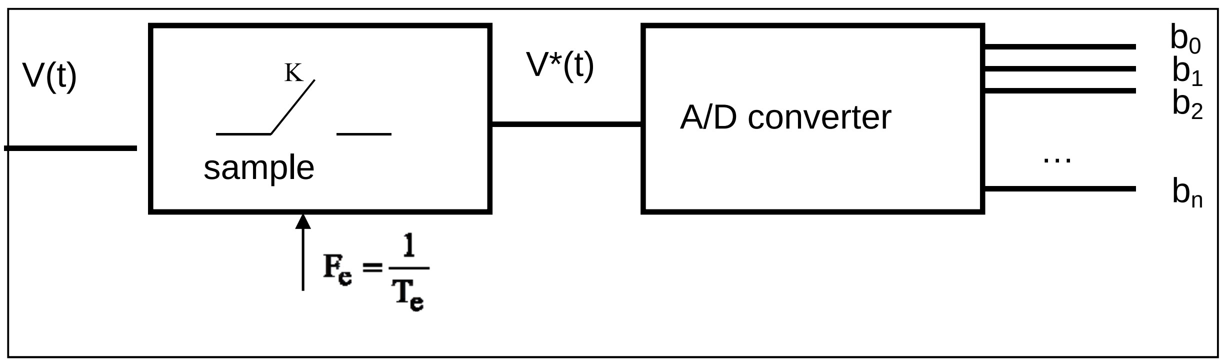


Figure 2: *Principle of analog-digital conversion*

From a mathematical point of view, this operation consists in taking the actual value of the signal V(t) at instants of time separated by a time constant T_e: the sample clock rate. We then obtain the sampled signal V*(t) (which is still an analog signal as it is not yet coded digitally).

I- Ideal sampling

In the case of ideal sampling the signal V*(t) is defined as:

$$V^*(t) = V(t) \text{ if } t = kT_e$$

$$V^*(t) = 0 \text{ if } t \neq kT_e$$

Question 1: Give the time representation of the sampled signal in Figure 3.

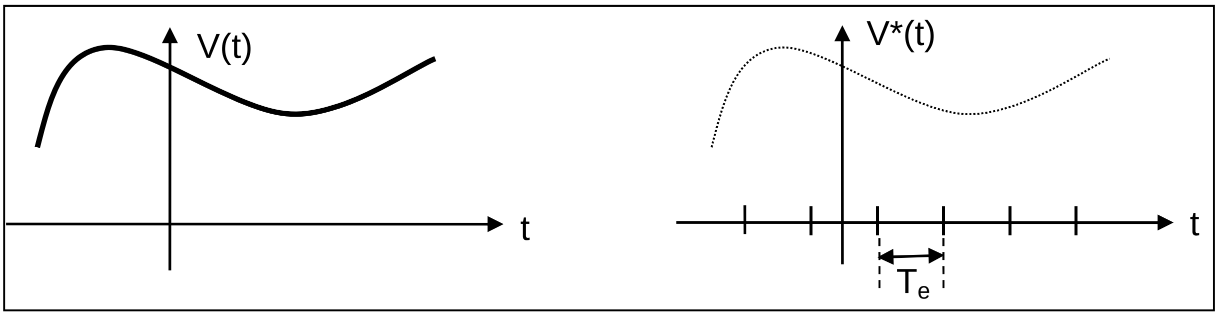


Figure 3: *Time representation of a sampled signal*

II- Practical sampling

Question 2: In practice, the on-off switch K in the sampler is activated by a logic pulsed signal g(t) as shown in Figure 4.
In this case, give the time representation of S(t) in Figure 5 when the signal V(t) is a sinusoid.

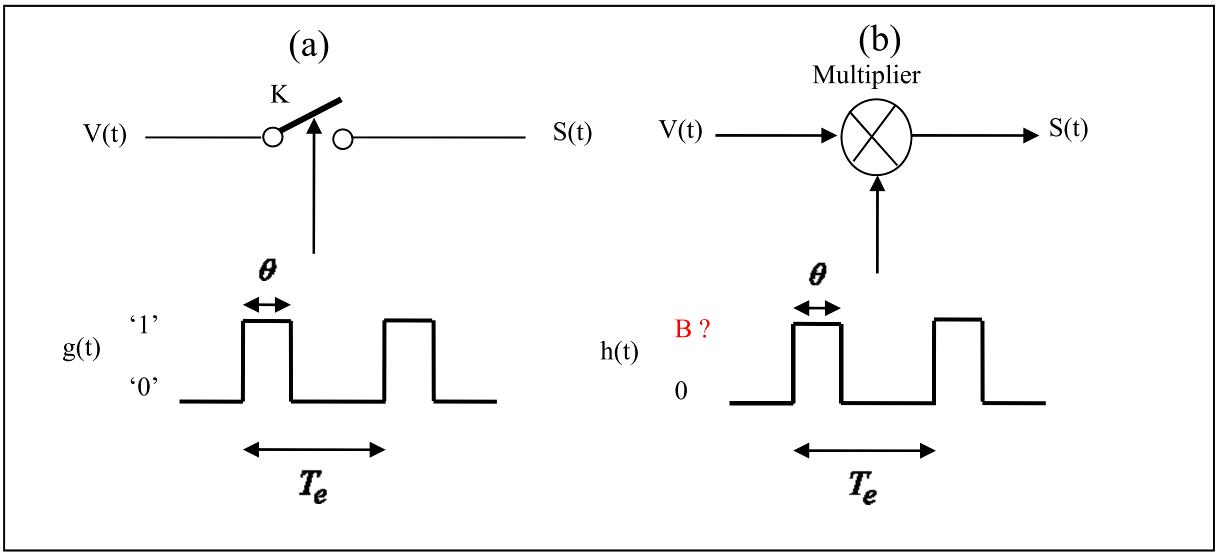


Figure 4: (a) *On-off switch K activated by a periodic pulsed signal, (b) Mathematical model of a real sampler*

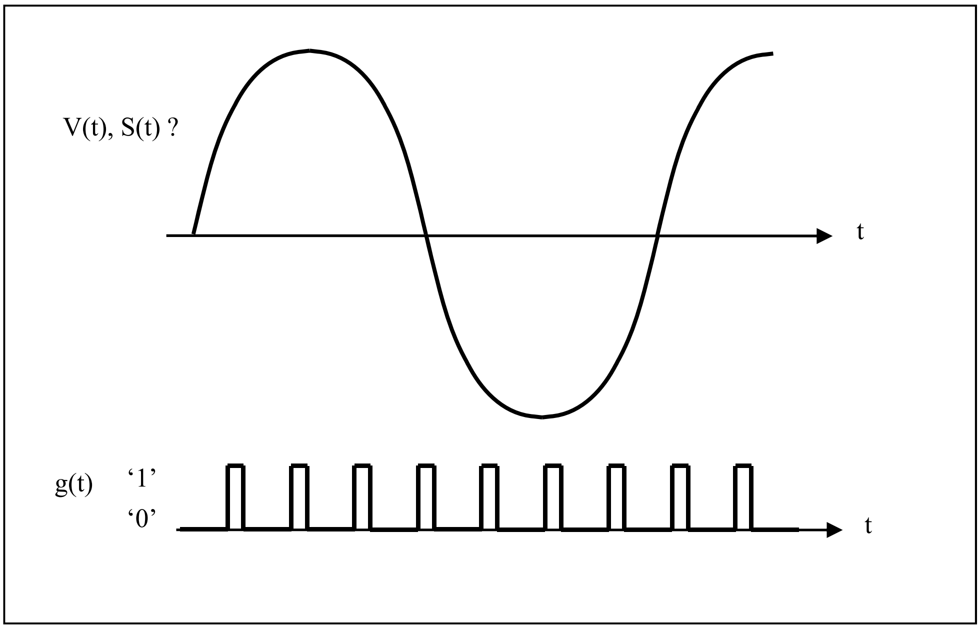


Figure 5: *Time representation of a signal sampled by a pulsed signal*

Question 3: To obtain the spectrum of the sampled signal S(t), the sampler in Figure 4-a is modeled with the mathematical model in Figure 4-b where h(t) is no longer a **logic signal** but a signal with **two algebraic values 0 and B**. What is the value of B?

To understand the effects of a sampling operation in the frequency domain, we look at the specific case of a signal of the type: $V(t) = A\cos(\omega_0 t)$.

The signal h(t) is periodic and admits as a Fourier series:

$$FS \text{ of } h(t) = \frac{B\theta}{T_e} + \frac{2B\theta}{T_e}\sum_{n=1}^{\infty}\frac{\sin\left(\frac{n\pi\theta}{T_e}\right)}{\left(\frac{n\pi\theta}{T_e}\right)}\cos(n\omega_e t) \text{ with } \omega_e = \frac{2\pi}{T_e}$$

Question 4:

- Write the signal S(t) resulting from the product of V(t) by h(t).
- Complete Table I by calculating the effective values of the first five frequencies of the spectrum of S(t).

the following are given: $A = 5V$, $\frac{\theta}{T_e} = 0.45$, $f_0 = 10kHz$ and $f_e = 80kHz$

NB: Remember that $\cos(a)\cos(b) = \frac{1}{2}[\cos(a+b) + \cos(a-b)]$

Frequencies					
Effective value					

Table I

Question 5: Plot the spectrum of S(t) in Figure 6.

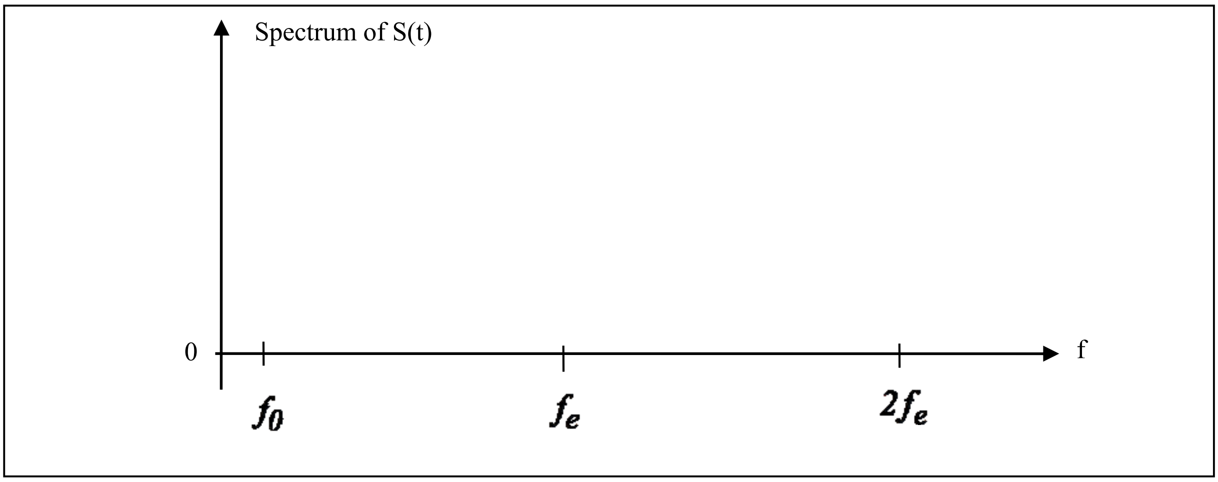


Figure 6: *Frequency representation of the sampled signal*

III- Choice of sampling rate: Shannon's Relation

As we have just seen, when sampling, a frequency f_e has to be set. But how should this be determined?

Question 6: From the spectrum in Figure 6, what relation should f_0 and f_e verify to recover the signal V(t) after filtering the sampled signal S(t)?

Now we replace the purely sinusoidal signal V(t) by a signal containing frequencies between *0* and $f_{\max} = 3kHz$ as shown in the graph in Figure 6. The sampling frequency f_e is $8kHz$.

Question 7: Draw the spectrum of S(t) on Figure 7, looking mainly at the frequency axis.

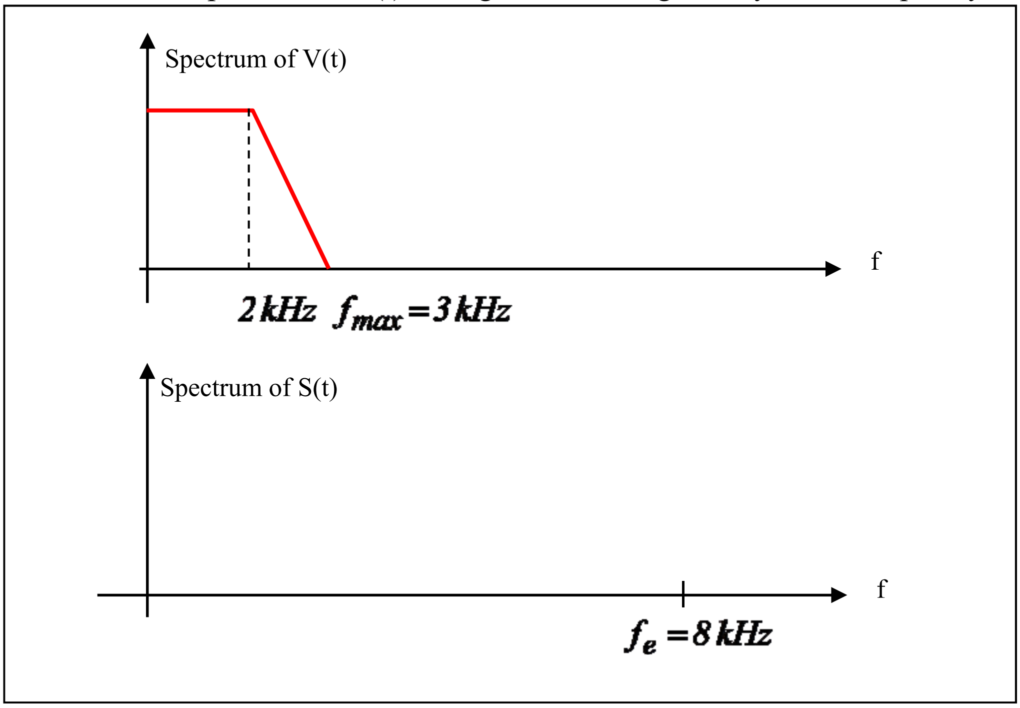


Figure 7: *Spectra of V(t) and S(t)*

What relation should f_{max} and f_e satisfy if we want to recover the signal V(t) by filtering the signal S(t). This is called **Shannon's relation** (American mathematician 1916 – 2001).

IV- Anti-aliasing filter

In Figure 7, we assumed a signal V(t) with no energy beyond frequency f_{max} but it has to be understood that such a signal does not exist. In practice, energy decreases as frequency increases but there is no frequency beyond which energy is zero, thus on the face of it there is a problem in choosing the sampling frequency: it would have to be infinite! In other words, is digital electronics impossible?

Take the case of telephony, the sampling frequency is 8kHz, however, a piece of music can produce sounds with much higher frequencies that are audible to the ear and which have a bandwidth of between 20Hz and 20kHz. **At first sight, Shannon's relation is not verified, yet it works because of a filter called the "anti-aliasing filter".**

To understand the key role played by the anti-aliasing filter in digital signal processing, we look at the spectrum of a signal V(t) like the one in Figure 8.

Question 8: Plot the spectrum of the sampled signal S(t) in Figure 8 when the sampling frequency f_e is 8kHz. Which frequencies of V(t) contribute to the 2kHz frequency of the signal S(t)?

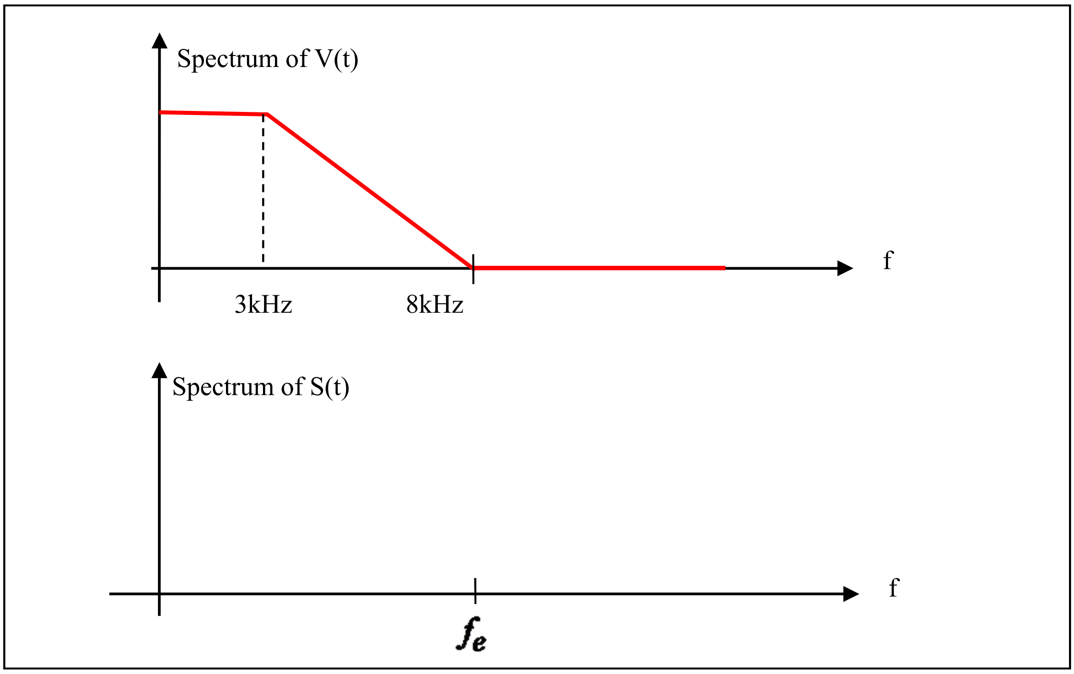


Figure 8: *Spectra of V(t) and S(t), sampling frequency is 8kHz*

Question 9: We want the spectrum of S(t) between 0 and 3kHz to be identical to the spectrum of V(t) in the same frequency range. What should we do?

Question 10: Consider a system carrying out analog-to-digital conversion on a signal. This system is made up of three blocks: an analog digital converter circuit (ADC), an anti-aliasing filter (AAF) and a sample-and hold device (SH).
Place these three elements (ADC, AAF, SH) in the correct order in the block diagram in Figure 9. Show which type of filter to use to produce an AAF. Justify your argument.

Figure 9: *Analog-to-digital conversion chain*

V- Analog-to-digital conversion (ADC-DAC)

V-1. Basic principle of conversion

Consider the processing chain in Figure 1. The sampling and analog-to-digital conversion operations are generally carried out in one and the same integrated circuit (IC).

The ADC delivers the samples $V(nT_e)$ from the analog signal $V(t)$ to the digital processer, coded in N_0 bits, where T_e is the sampling period. The role of the processor is to modify the initial signal by specific digital processing techniques (echo generation, frequency suppression, voluntary distortion, etc.). For example, echo generation will be covered in S4 when we study digital filters.
The processor adjusts another series of samples $S(nT_e)$ to correspond to the first series $V(nT_e)$, for example in the form of an equation such as: $S(nT_e) = \frac{1}{2}\{V(nT_e) + V((n-1)T_e)\}$. To reconstruct the analog signal $S(t)$, a Digital-to-Analog Converter (DAC) is used, as shown in Figure 10.
In the practical assignment (PA) on this topic, the processer carries out the operation $S(nT_e) = V(nT_e)$. In other words, it does nothing and we might expect to obtain: $S(t) = V(t)$. This is not the case, however, and we must try and understand why.

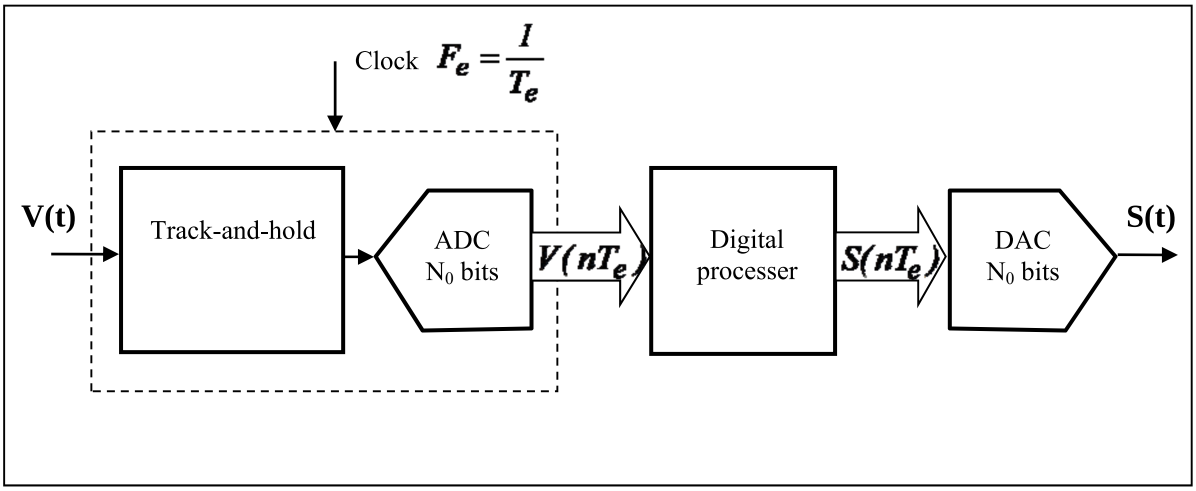


Figure 10: *Digital signal processing chain*

The number, often called the binary word (N), will be characterized by its **number of bits**, written as follows:

$$N = a_{n-1}a_{n-2}\ldots\ldots a_1a_0 \qquad (1)$$

where a_0 represents the least significant bit (LSB) and a_{n-1} the most significant bit (MSB)

The value of the analog voltage V to be converted (ADC) or that has been converted (DAC) is discrete and corresponds to a multiple of a basic value called the **quantum of conversion q** (elementary analog voltage):

$$V = q.(a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \ldots\ldots + a_1 2^1 + a_0 2^0) = q.\ N_{dec} \qquad (2)$$

The two most important features of an ADC or a DAC are thus:

- the number of bits: N_0
- the elementary analog voltage: q

Performing an analog-to-digital conversion involves searching for a digital expression in a code determined to represent analog information (Figure 11).

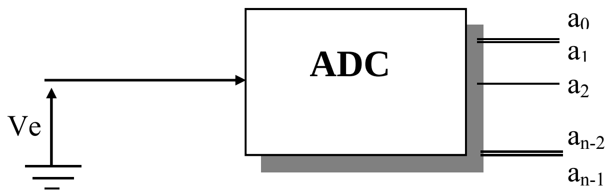


Figure 11*: General role of an ADC*

The quantity q is derived from equation (3):

$$q = \frac{Ve_{max}}{2^{N_0} - 1} \qquad (3)$$

Analog-to-digital conversion is characterized by a **transfer characteristic** usually called a "stairstep graph". See, for example, a 3-bit ADC (Figure 12).

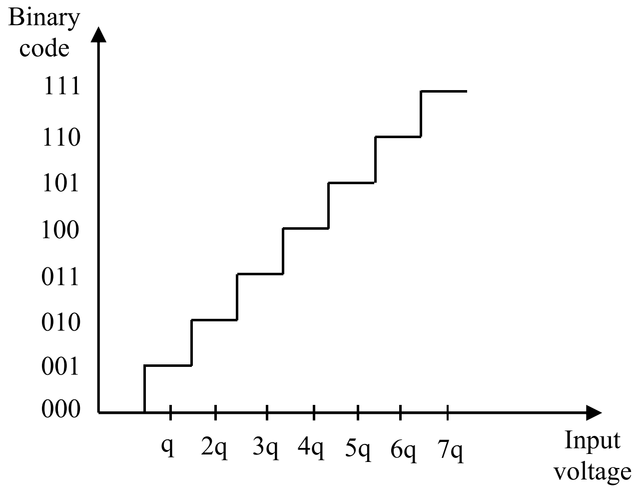


Figure 12: *Ideal transfer of an ADC using 3-bit conversion*

From this graph, the main problem linked with conversion can now be seen. A digital word N can correspond to an input voltage Ve such that:

$$N_{dec}.q - q/2 < Ve < N_{dec}.q + q/2 \qquad (4)$$

where q/2 represents the amplitude of the quantization error

Example: With ADC, a voltage between 0 and 10V can be converted on 3-bit binary codes.

$$q = Ve_{max} / (2^{N}{}_{0} - 1) = 10 / (2^3 – 1) = 10 / 7 \approx 1.43V$$

this means that for the code (100), the corresponding voltage will be between:

$$4q - q/2 = 5V < Ve < 4q + q/2 = 6.42V$$

It is easy to understand the difficulty in coding a binary number on only 3 bits when, for example, the specifications require detection of a fluctuation in voltage to a maximum of 0.1V.

V-2. Usual parameters of an ADC

The usual and functional parameters regarding the choice of ADC are as follows:

- **Full-scale range (or FSR):** This is the maximum acceptable voltage, i.e. $FSR = q.2^{N}{}_{0}$ for a linear converter.
- **Resolution:** This is expressed in number of bits of conversion. The number of binary numbers (or codes) likely to be generated is $2^{N}{}_{0}$.
- **Precision:** To attribute a code to a voltage that is to be converted, it is first necessary to determine the thresholds between which the voltage Ve for conversion is located. In theory, the thresholds are multiples of $FSR/2^{N}{}_{0}$. In practice, thresholds may differ

slightly from these multiples. The difference between the theoretical and the real thresholds is called converter precision and it is often expressed as a fraction of LSB.

• **Quantization error:** All voltages in the same range have the same code attributed (see Figure 13). Thus with only the code, we are unable to find the exact value of the corresponding voltage. If the sampling rate is very high (theoretically infinite), then the code is usually attributed a value from the middle of the range q_k. Quantization error, in the worst case scenario, is then ± q/2.

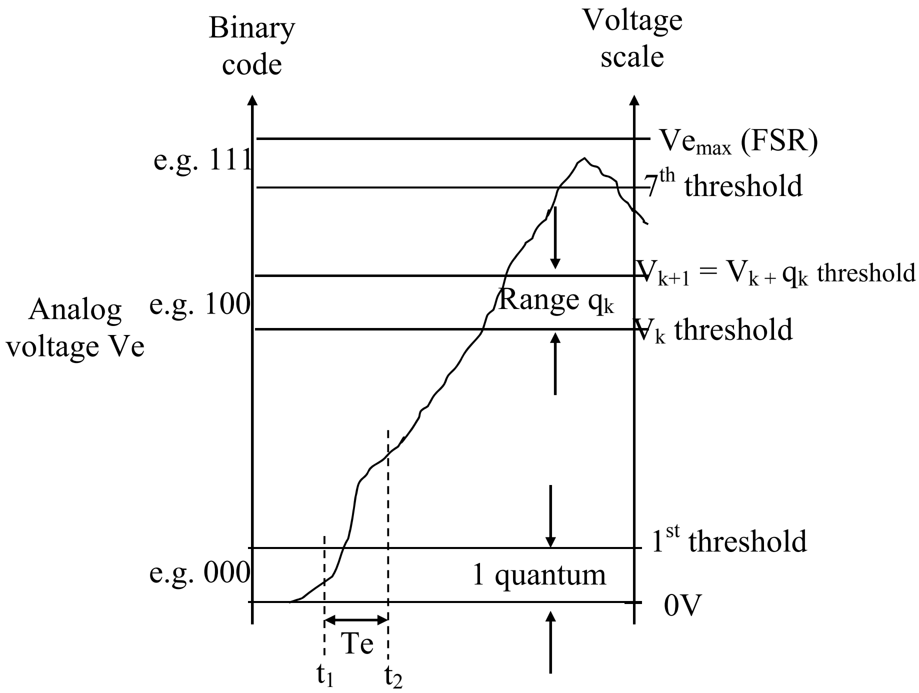


Figure 13: *Illustration of the problem associated with voltage -> code (N_0 = 3 bits) correspondence*

V-3. Track and hold device

Converting an analog sample of V(t) by ADC into a binary code with N_0 bits, requires some time. During this time, the output signal from the sampler must remain constant. To do this, **a track & hold (or sample & hold) device is used** which is shown in simplified form in the diagram in Figure 14.

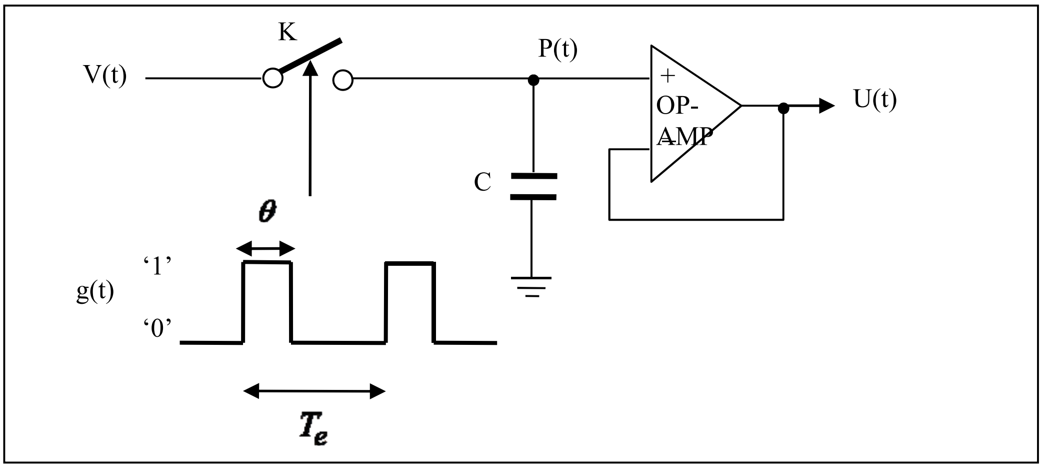


Figure 14: *Diagram of track & hold device*

Question 11: Complete the drawing in Figure 15. What is the advantage of putting an OP-AMP after the capacitor C?

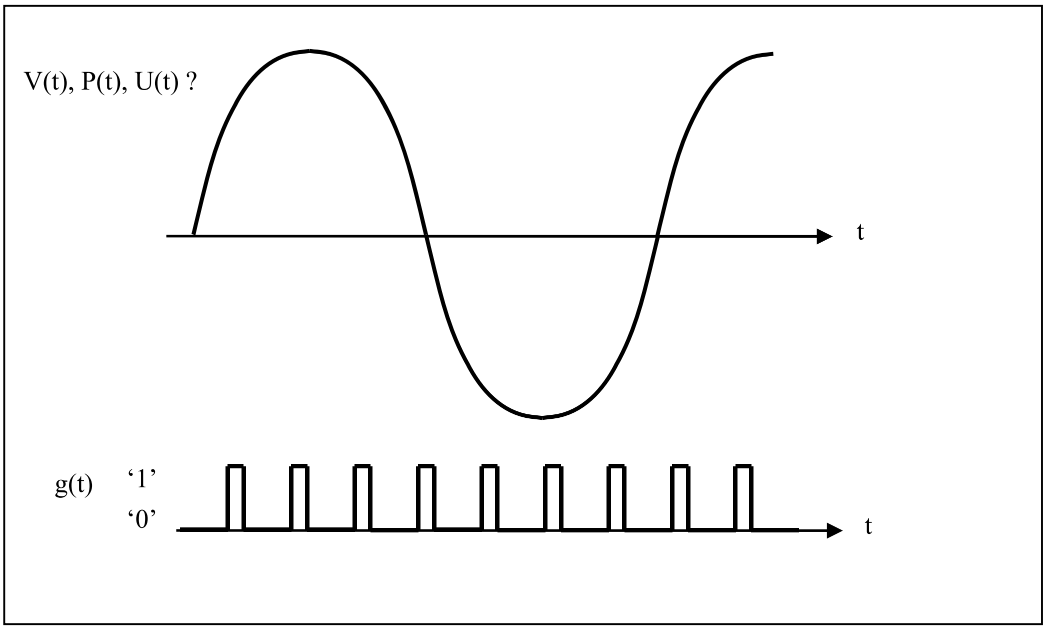


Figure 15: *V(t) input analog signal and U(t) signal tracked and held*

Question 12: How is the U(t) signal modified if the switch K is not perfect and modeled by resistance r?

V-4. Characteristics of an ADC

The ADC used in the PA (AD7819 Analog Device) is a unipolar 8-bit successive approximation ADC and full scale 5V. A track and hold device (T/H) is incorporated into the IC. The functional blocks of the ADC are shown in Figure 16. Analog-digital conversion is initiated on the falling edge of the $\overline{CONVST}$ signal, as shown in the signal timing diagram in Figure 17.

After a request for conversion, the BUSY signal (output) moves into "high" state and the falling edge at BUSY shows that data (DB0 – DB7) are available as outputs. To read them, inputs $\overline{RD}$ and $\overline{CS}$ must switch to "low" state. When the $\overline{RD}$ and $\overline{CS}$ inputs are "high", outputs (DB0 – DB7) are in high impedance (*three state drivers*).

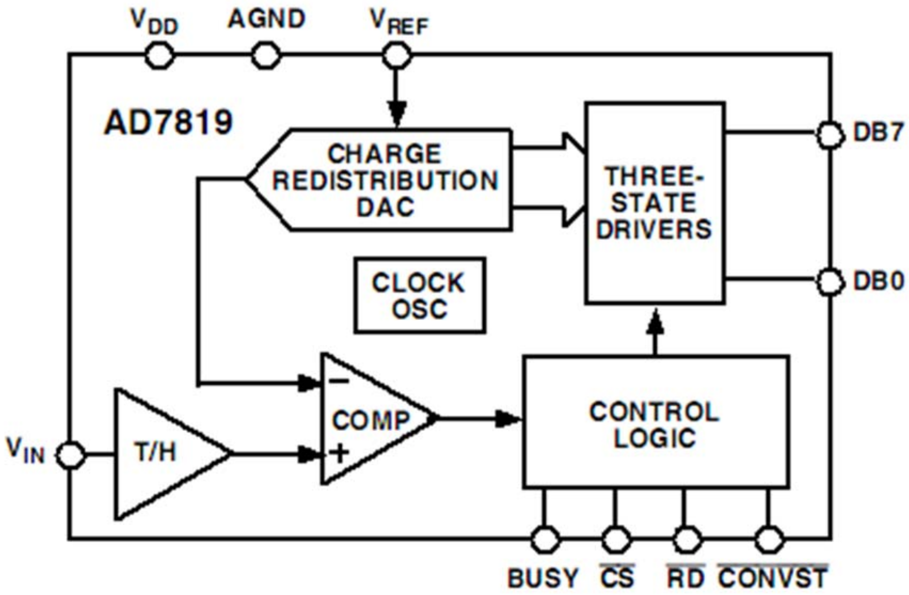


Figure 16: *Functional blocks of ADC AD7819*

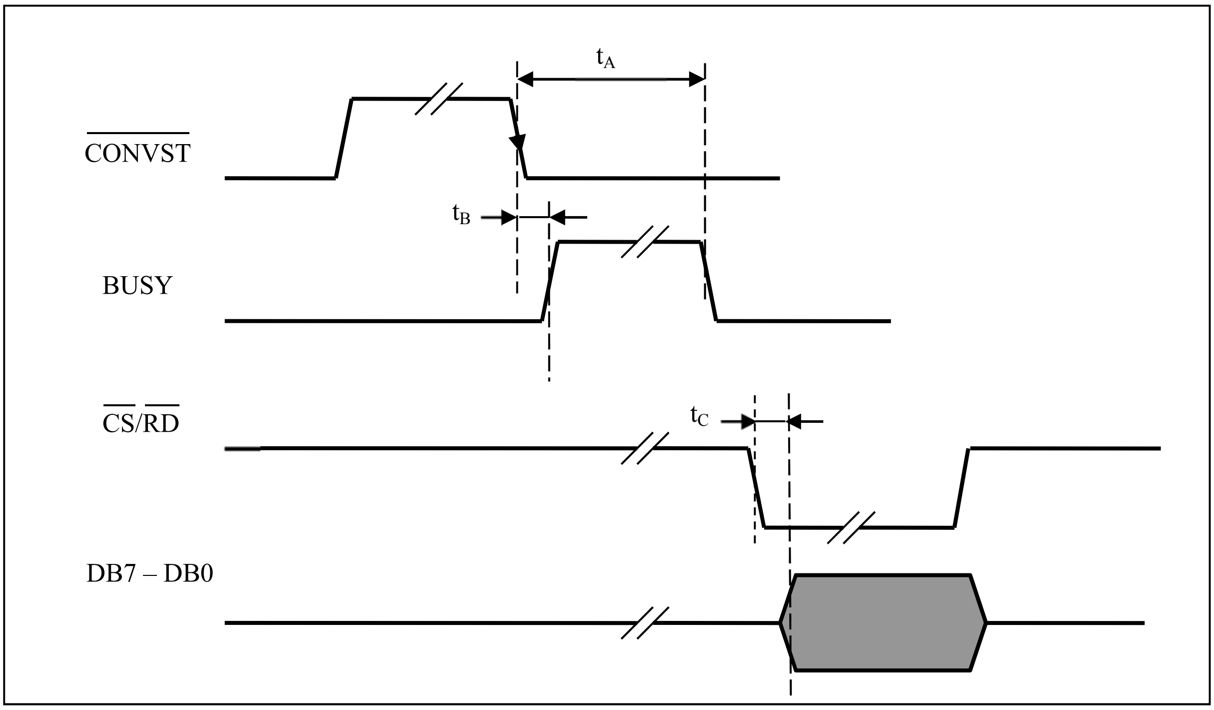


Figure 17: *Timing diagram of the main signals in ADC AD7819*

Question 13: Using Tables II and III to help you, give the values for times t_A, t_B and t_C in the timing diagram in Figure 17. From this, deduce the maximum conversion frequency of the ADC in kSPS (*kilo Samples per Second*).

PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Description
1	V_{REF}	Reference Input, 1.2 V to V_{DD}.
2	V_{IN}	Analog Input, 0 V to V_{REF}.
3	GND	Analog and Digital Ground.
4	$\overline{\text{CONVST}}$	Convert Start. A low-to-high transition on this pin initiates a 1 µs pulse on an internally generated $\overline{\text{CONVST}}$ signal. A high-to-low transition on this line initiates the conversion process if the internal $\overline{\text{CONVST}}$ signal is low. Depending on the signal on this pin at the end of a conversion, the AD7819 automatically powers down.
5	$\overline{\text{CS}}$	Chip Select. This is a logic input. $\overline{\text{CS}}$ is used in conjunction with $\overline{\text{RD}}$ to enable outputs.
6	$\overline{\text{RD}}$	Read Pin. This is a logic input. When $\overline{\text{CS}}$ is low and $\overline{\text{RD}}$ goes low, the DB7–DB0 leave their high impedance state and data is driven onto the data bus.
7	BUSY	ADC Busy Signal. This is a logic output. This signal goes logic high during the conversion process.
8–15	DB0–DB7	Data Bit 0 to 7. These outputs are three-state TTL-compatible.
16	V_{DD}	Positive power supply voltage, +2.7 V to +5.5 V.

Table II: *Description of the different pin functions in the ADC AD7819*

TIMING CHARACTERISTICS[1, 2] (–40°C to +125°C, unless otherwise noted)

Parameter	V_{DD} = 3 V ± 10%	V_{DD} = 5 V ± 10%	Units	Conditions/Comments
$t_{POWER\text{-}UP}$	1	1	µs (max)	Power-Up Time of AD7819 after Rising Edge of $\overline{\text{CONVST}}$.
t_1	4.5	4.5	µs (max)	Conversion Time.
t_2	30	30	ns (min)	$\overline{\text{CONVST}}$ Pulsewidth.
t_3	30	30	ns (max)	$\overline{\text{CONVST}}$ Falling Edge to BUSY Rising Edge Delay.
t_4	0	0	ns (min)	$\overline{\text{CS}}$ to $\overline{\text{RD}}$ Setup Time.
t_5	0	0	ns (min)	$\overline{\text{CS}}$ Hold Time after $\overline{\text{RD}}$ High.
t_6[3]	10	10	ns (max)	Data Access Time after $\overline{\text{RD}}$ Low.
t_7[3, 4]	10	10	ns (max)	Bus Relinquish Time after $\overline{\text{RD}}$ High.
t_8[3]	100	100	ns (min)	Data Bus Relinquish to Falling Edge of $\overline{\text{CONVST}}$ Delay.

Table III: *Timing characteristics of the ADC AD7819*

Question 14: What type of integrated circuit should be connected to the "BUSY" signal on the ADC in Figure 18 in order to read data (DB0 – DB7)?

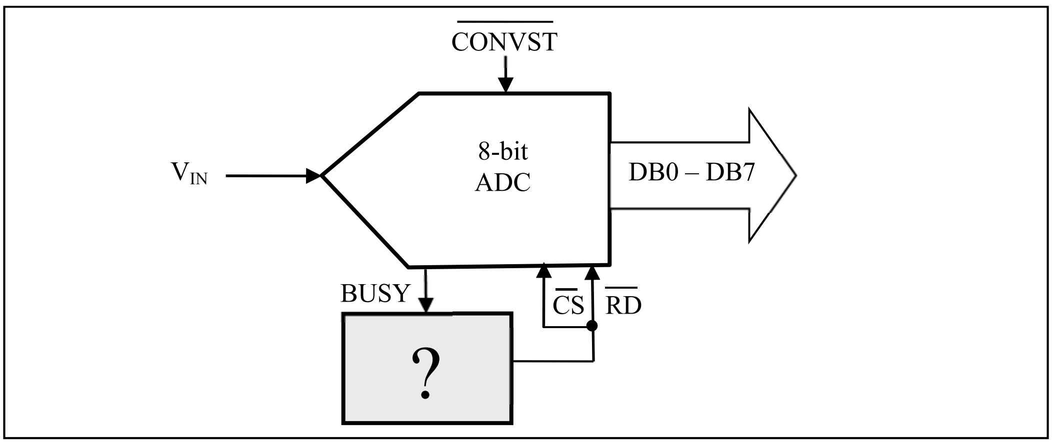


Figure 18: *Complete the assembly with the missing integrated circuit*

V-5. Assignment by DAC – Quantization noise

The truth table for the DAC in Figure 10 is given in Table IV.

Binary code	Output voltage
0000 0000	0
0000 0001	5/256 = 0.0195V
0000 0010	2(5/256) = 0.0390V
0000 0011	3(5/256) = 0.0585V
1111 1111	255(5/256) = 4.980V

Table IV: *Truth table of the DAC*

Whereas the signal V(t) in Figure 10 is a triangle, the signal S(t) consists of a stairstep signal. A sample of V(t) has a corresponding 8-bit code, and when the DAC receives this code it assigns a corresponding voltage from Table IV. The assigned voltage is different from the value of the sample of V(t), as can be seen in Figure 19, since merely knowing the code is no guarantee of knowing the initial value of the sample exactly. **The difference between the initial value and the assigned value is called the quantization noise**. To reduce quantization noise, the number N_0 of conversion bits must be increased.

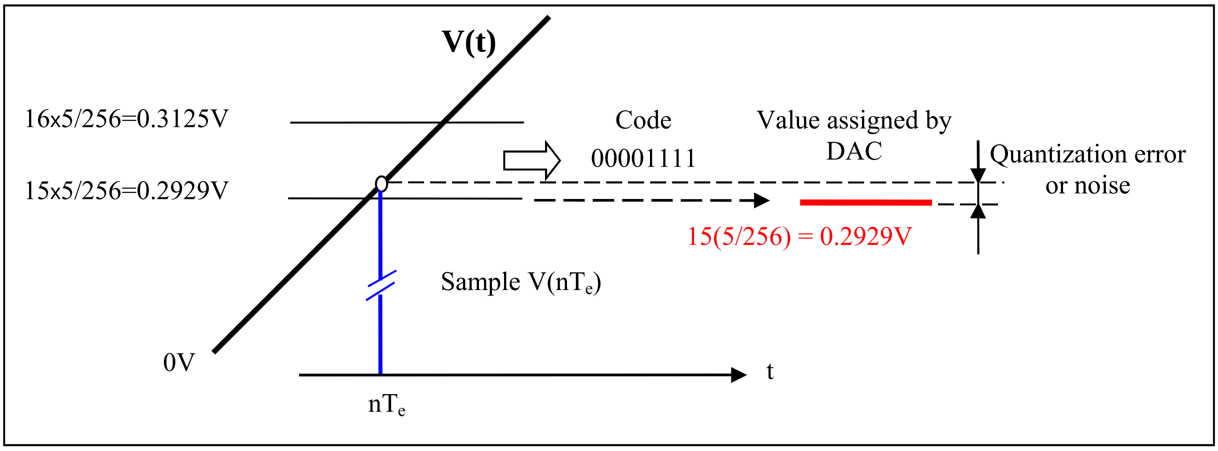


Figure 19: *Illustration of the problem of the assigned value being different from the initial value. The difference is called the quantization noise*

The signal V(t) in Figure 10 is a symmetrical ramp of 5V and frequency 1kHz, as shown in Figure 20.

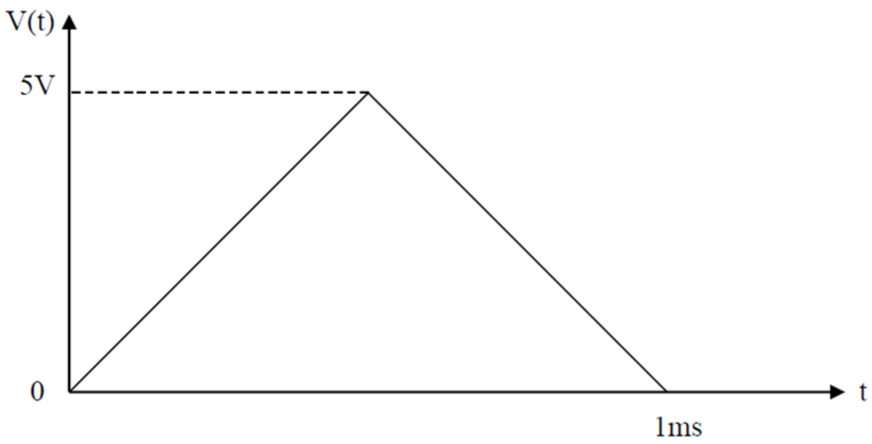


Figure 20: *Timing characteristics of the signal V(t)*

Question 15: Determine the height of the steps (ΔV) of signal S(t) as output from the DAC when the sampling frequency is *80 kHz* . Compare this with the theoretical quantization error (± q/2).

Question 16: What is the new value of ΔV if the rate is divided by 2?

REMEMBER: It is important to remember that when performing an analog-digital conversion of a time-varying signal (e.g. the voltage supplied by a sensor) into a binary code or word (on N_0 bits), a track & hold device must be used then a converter block. The sampling process involves selecting a frequency F_e which must satisfy Shannon's relation ($F_e > 2.\ f_{max}$). It is often combined with an anti-aliasing filter (AAF) so that lines caused by the sampling operation can be eliminated (periodization of frequency F_e of the signal spectrum before sampling) and can be superimposed on the "useful" lines of the initial spectrum. There are four main parameters that guide the choice of ADC or DAC: full-scale range of admissible voltage (FSR), resolution (number of bits), precision (linked with the quantum of conversion) and quantization error, which is important mainly with assignment after DAC.

VI- Applied exercise *(taken from an S3 exam paper)*

A sinusoidal signal $V(t) = 2\cos\left(2\pi 10^3 t\right)$ is sampled with a sampling frequency of $F_e = 10kHz$. The pulses for sampling have a duty cycle of 10%.

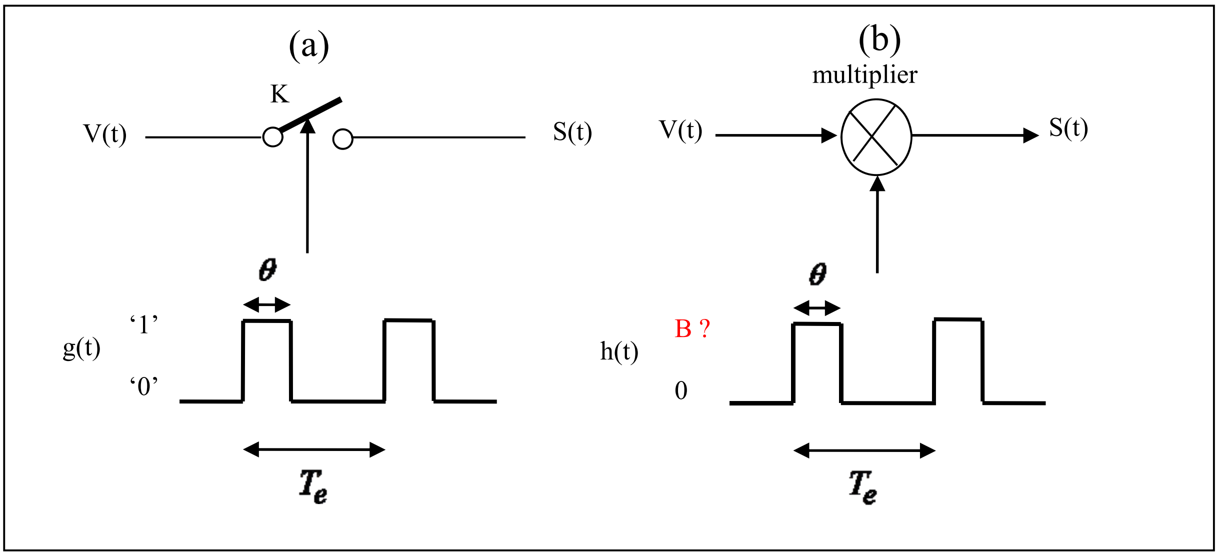


Figure 1: *Sampler and associated mathematical model*

Q1- Give the first three frequencies of the spectrum of *S(t)*.

Q2- Calculate the amplitudes of the first three frequencies.

Q3- The samples of the signal $V(t)$ are coded on N_0 bits. Quantization noise is set lower than $7.9mV$ (difference between the real value of the sample and the value assigned after DAC). Calculate the number of N_0 needed for the chosen converter.

Topic 5:
Time-frequency transformation by FFT

Objectives:

- To learn how to read the results of an FFT
- To learn how to choose an analysis window
- To learn how to calibrate pulse width to obtain the pulse response of a system in order to calculate its frequency response (complex gain).

I - Reminder of the lectures

I-1/ Fourier transform

The Fourier transform $G(\omega)$ of a signal $g(t)$ is defined as follows:

$$G(\omega) = \int_{-\infty}^{\infty} g(t)e^{-j\omega t}dt \qquad (1)$$

A Fourier transform is a **continuous function** of a pulse ω.

I-2/ Digital Fourier transform

Various operations are necessary to calculate a Fourier transform using a processor (see Fig. 1). Those operations are as follows:

- sample the signal at a given frequency $F_e = \frac{1}{T_e}$
- digitize the signal with an ADC of M bits
- cut off the signal at NT_e where N is the number of points stored in the memory
- use a calculation algorithm

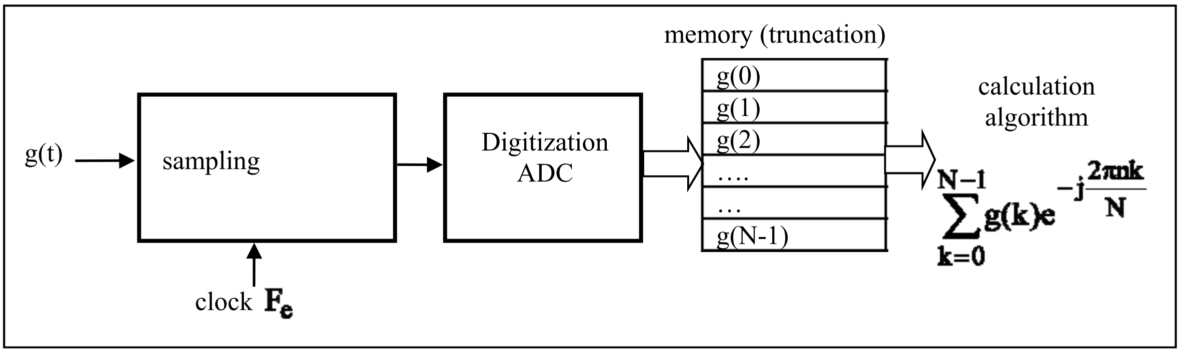


Figure 1: *The different operations required to calculate a Fourier transform using a processor*

Let's put the problem of digitization to one side for the time being, and focus on the effects of **sampling** and **truncation**. Sampling means that we no longer have $g(t)$ but rather the samples $g(kT_e)$ taken at the sampling times kT_e where k is an integer. The Fourier transform (1) thus becomes a **digital Fourier transform** $G_{num}(\omega)$:

$$G_{num}(\omega) = \sum_{k=-\infty}^{\infty} g(kT_e)e^{-j\omega kT_e} \qquad (2)$$

For simplicity's sake we write this as: $G_{num}(\omega) = \sum_{k=-\infty}^{\infty} g(k)e^{-j\omega kT_e}$.

Given the periodicity of the sine and cosine functions, the digital Fourier transform $G_{num}(\omega)$ is a **periodical function** of the sampling frequency F_e as we can see in Figure 2-b. We thus arrive at the result already obtained when studying the sampling theory.

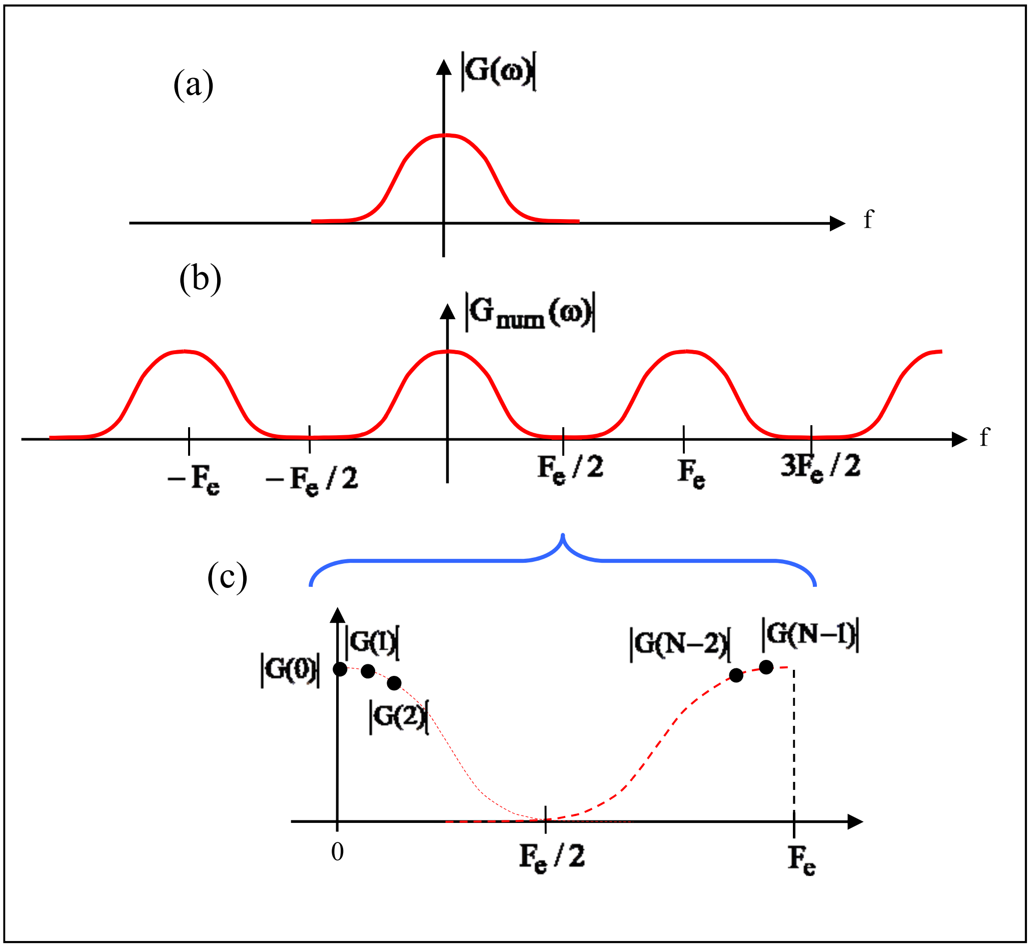


Figure 2: *(a) Fourier transform G(f) of the function g(t), (b) digital Fourier transform* $G_{num}(\omega)$ *of samples* $g(kT_e)$ *and (c) discrete Fourier transform (DFT)*

Given the periodicity of the sine and cosine functions, the $|G_{num}(\omega)|$ module is in the interval $[0, F_e]$ and symmetrical with $\frac{F_e}{2}$. We thus arrive at the result already obtained when studying the sampling theory.

I-3/ Discrete Fourier transform

In practice we cannot store an infinite number of samples $g(kT_e)$, we store only N samples, so **truncation** is required ($G_{num}(\omega) \to \sum_{k=0}^{N-1} g(kT_e)e^{-j\omega kT_e}$). The digital Fourier transform $G_{num}(\omega)$ is a ***continuous function*** of a pulse ω. A processor can only calculate $G_{num}(\omega)$ for certain values of pulse ω. Given the periodicity of $G_{num}(\omega)$, we only calculate $G_{num}(\omega)$ for N frequencies, multiples of $\frac{F_e}{N}$ where N is the number of acquisition points. We thus obtain N values, and these N values collectively represent the **Discrete Fourier Transform**, written as $G(n)$ (Figure 2-c):

$$G(n) = G_{num}(\omega)\Big|_{\omega = n2\pi\frac{F_e}{N}} = \sum_{k=0}^{N-1} g(k)e^{-jn2\pi\frac{F_e}{N}kT_e}$$

Bearing in mind that $F_e = \frac{1}{T_e}$ we end up with:

$$G(n) = \sum_{k=0}^{N-1} g(k)e^{-j\frac{2\pi n k}{N}} \quad \text{where } n = 0, 1, 2, \quad \dots N-2, N-1 \qquad (3)$$

I-4/ FFT and zero padding

Calculating a Fourier transform requires a great many multiplication and addition operations. In order to obtain the value G(n) we must perform N complex multiplications ($g(k)e^{-j\frac{2\pi n k}{N}} = g(k)\left[\cos(\frac{2\pi n k}{N}) - j\sin(\frac{2\pi n k}{N})\right]$) and N additions. As multiplications require more machine cycles than additions, broadly speaking it is the number of multiplications which will determine the calculation time. To obtain the N values required for G(n), N^2 complex multiplications are needed. In the 1950s, mathematicians Cooley and Tuckey demonstrated that if the number of points N was a power of two, $N = 2^p$ (e.g.: $N = 1024 = 2^{10}$, $p = 10$), then an algorithm could be used to rapidly calculate the number of multiplications from N^2 to pN. Consider the example $N = 1024$: without a quick algorithm we would need to perform $\approx 10^6$ multiplications, with the algorithm only $\approx 10^4$ are required, thus cutting the calculation time 100-fold. This rapid calculation algorithm is known as the **FFT** (**F**ast **F**ourier **T**ransform).

If the number of points N is not a power of two, we are left with two options: *i)* we can use a **DFT** (**D**irect **F**ourier **T**ransform) algorithm, requiring N^2 multiplications or *ii)* we can fill out the number of samples N with zeroes to obtain a number of samples equal to a power of two:

$$G(n) = \sum_{k=0}^{N-1} g(k)e^{-j\frac{2\pi n k}{2^p}} + \sum_{N}^{2^p} 0e^{-j\frac{2\pi n k}{2^p}}$$

Adding zeroes does not alter the total and the **digital Fourier transform** remains unchanged, but the calculation step of the discrete Fourier transform which was $\frac{F_e}{N}$ now becomes $\frac{F_e}{2^p}$. This is the technique used by the DSO5032A oscilloscope we use for our practical work. This technique is known as **zero padding**.

I-5/ Truncation effect and window

FFT is an operation involving a finite number of samples. In practice we encounter two types of signal, as seen in Figure 3:

- ***Type I*** signals which are nil before and after the NT_e recording window. The samples $s(kT_e)$ stored in the memory are the product of samples $g(kT_e)$ from samples $p(kT_e)$ taken in an **analysis window** (or weighting window), with $p(kT_e) = 1$ for $k = 0, 1, 2, ...(N-1)$ and $p(kT_e) = 0$ for $k < 0$ and $k > N-1$, it follows that $s(kT_e)$ are equal to $g(kT_e)$ regardless of the value of k. For Type I signals, the **digital Fourier transforms** $S_{num}(\omega)$ and $G_{num}(\omega)$ are thus equal. If the Nyquist-Shannon theorem is respected, i.e. the frequency F_e is at least two times greater than the maximum frequency contained in the spectrum of g(t), then the calculated N values $S(n)$ are identical to the N values of $G(f = n\frac{F_e}{N})$ where G(f), equation (1), is the **Fourier transform** of signal g(t). In conclusion, for Type I signals there is no problem, and we can proceed as if we knew how to calculate the Fourier transform. The only problem which arises is the issue of quantization noise, as the samples are quantized with a finite number of bits.

- ***Type II signals*** are periodic signals ranging from $-\infty$ to $+\infty$. In this case the samples $s(kT_e)$ are equal to the samples $g(kT_e)$ within the window, but they are nil outside the window while $g(kT_e)$ are not equal to zero for $k < 0$ and $k > N-1$, and as such $S_{num}(\omega) \neq G_{num}(\omega)$. In this case we may end up with $S(n)$ values which are different from $G(f = n\frac{F_e}{N})$.

We will come back to this scenario after first studying the frequency response of a system.

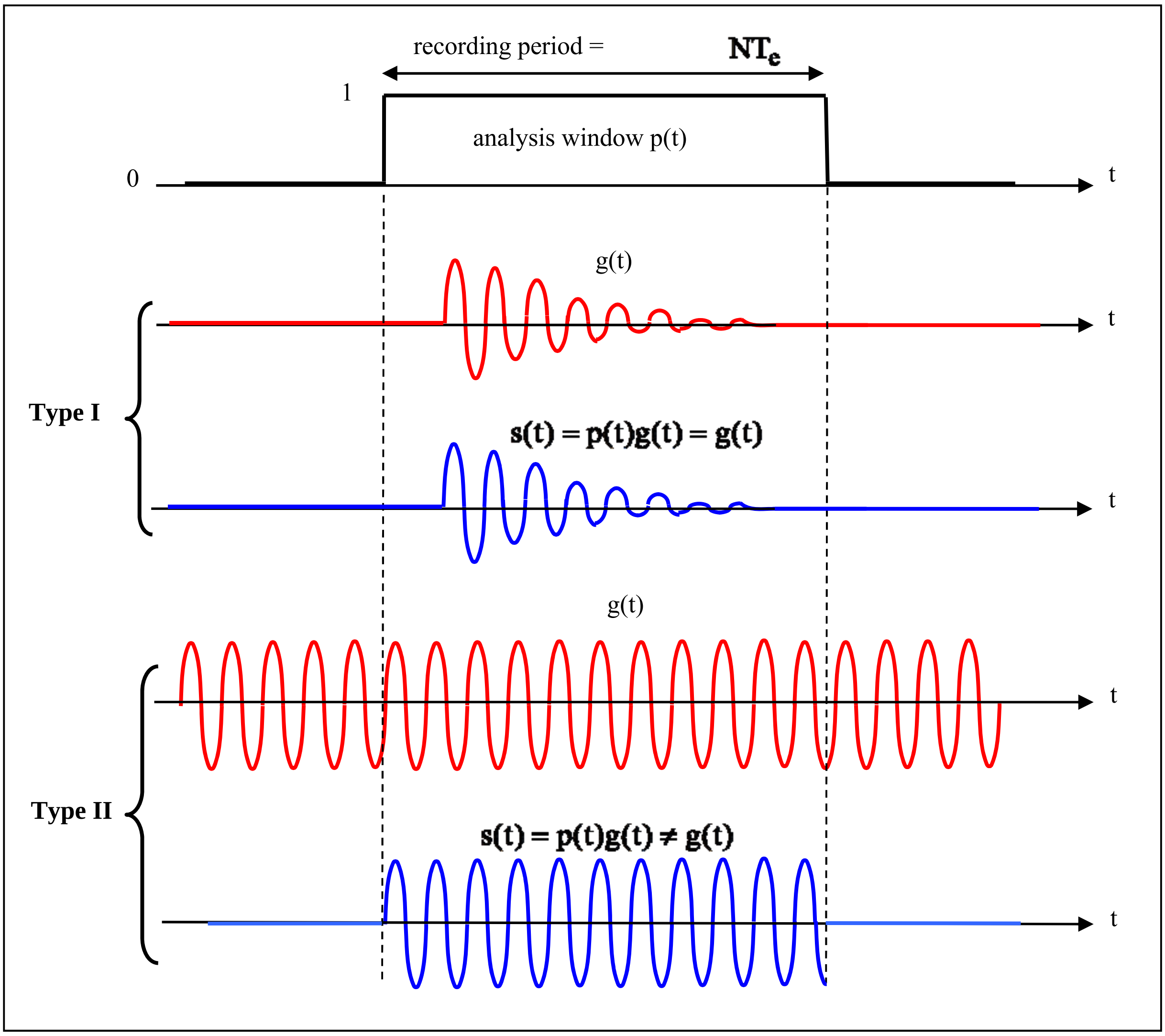


Figure 3: *The samples used to calculate the FFT are obtained by truncating the samples* $g(kT_e)$ *by the samples* $p(kT_e)$ *of an analysis window. There are two signal types, Type I and Type II.*

II - Frequency response of a linear system

The frequency response $H(\omega)$, or complex gain, of a linear system can be obtained (see lecture on Applied mathematics for signal processing) by passing a Dirac pulse $\delta(t)$ through the system and calculating the Fourier transform of the pulse response $h(t)$ as seen in Figure 4-a. As the system is stable, $h(t) \to 0$ when $t \to \infty$, the pulse response is a Type I signal. In practice, the system receives a pulse x(t) of bandwidth θ and amplitude A (figure 4-b), making the frequency response:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} \qquad (4)$$

where $X(\omega)$ and $Y(\omega)$ are the Fourier transforms for x(t) and output y(t) respectively.

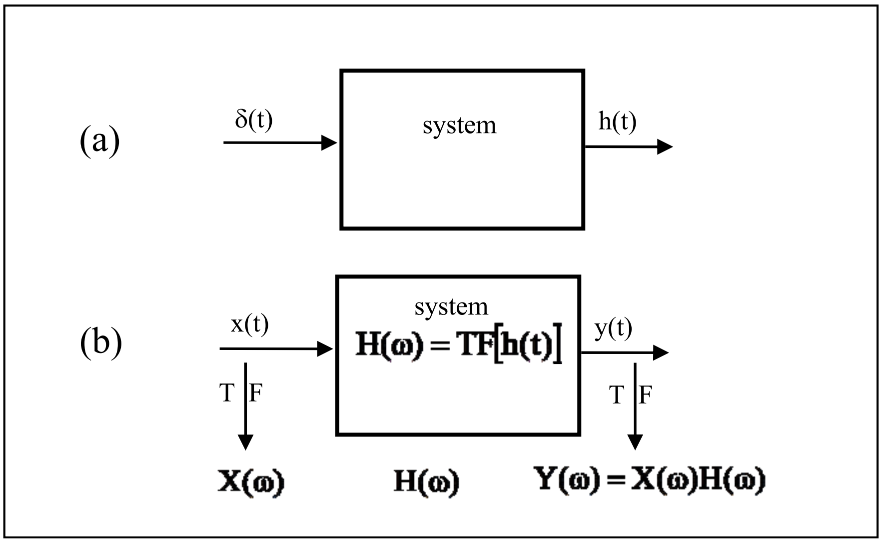


Figure 4: *Frequency response of a system using the pulse method, (a) pulse response h(t), (b) response to excitation x(t).*

The electrical diagram for one of the two systems used in the practical assignment is given in Figure 5. This is the open loop of the oscillator used in the corresponding practical assignment, composed of an amplifier followed by a pass-band filter.

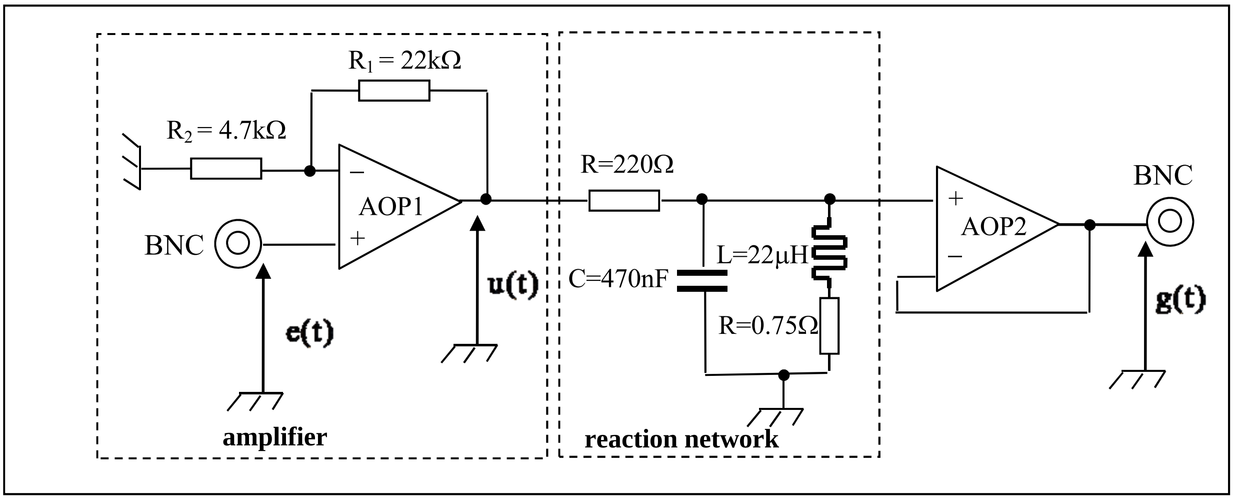


Figure 5: *Electrical diagram showing the amplifier and the pass-band filter*

Question 1: Draw the electrical diagram at zero frequency and demonstrate that the static gain of the loop is equal to:

$$H(\omega)\Big|_{\omega\to 0} = (1+\frac{R_1}{R_2})\left(\frac{r}{R+r}\right) \quad (5)$$

The function of the transfer $H(p)=\frac{G(p)}{E(p)}$ is expressed in the form:

$$H(p)=\frac{G(p)}{E(p)}=(1+\frac{R_1}{R_2})\frac{1}{RLC}\left(\frac{pL+r}{p^2+p\frac{(L+rRC)}{RLC}+\frac{(r+R)}{RLC}}\right) \quad (6)$$

N.B.: *You will not be asked to calculate* $H(p)$ *during this session, but I highly recommend that you do so in your own time.*

Question 2: Check that for $\omega \to 0$ we get the same static gain calculated above.

The standard form of a second-order pass-band filter is written: $A_{BP}\frac{p\frac{\omega_0}{Q}}{p^2+p\frac{\omega_0}{Q}+\omega_0^2}$. The gain at resonant frequency $f_0=\frac{\omega_0}{2\pi}$ is equal to A_{BP}, the bandwidth Δf to $-3dB$ is equal to $\Delta f=\frac{f_0}{Q}$.

Question 3: Around the resonant frequency f_0, $r << L\omega_0$, demonstrate that the expression (6) is correctly expressed as:

$$H(p)\approx A_{BP}\frac{p\frac{\omega_0}{Q}}{p^2+p\frac{\omega_0}{Q}+\omega_0^2} \quad (7)$$

Question 4: Take: $r=0{,}75\Omega$, $R=220\Omega$, $C=470nF$, $L=22\mu H$, $R_1=22k\Omega$ and $R_2=4{,}7k\Omega$, determine A_{BP} i.e. $|H(\omega_0)|$, ω_0 and Q

Question 5: To check the previous calculations, apply a pulse x(t) at the entry to the system and calculate the FFT or response y(t) using a digital scope with FFT (see Fig. 4). At resonant frequency f_0, the scope shows $|X(\omega_0)|=-38{,}50dBV$ and $|Y(\omega_0)|=-36{,}85dBV$ respectively. Calculate $|H(\omega_0)|$.

N.B.: dBV is expressed as follows: $dBV = 20\log_{10}\left(\frac{\text{tension en V}}{1V}\right)$

III- Analysis windows and spectrum leakage

In order to understand the effect of an **analysis window,** consider the example of a signal $g(t) = A\cos(\omega_0 t)$, (see Fig. 3). The Fourier transform G(f) is written (no demonstration):

$$G(f) = \frac{A}{2}\left[\delta(f - f_0) + \delta(f + f_0)\right] \qquad (8)$$

G(f) is nil apart from at f_0 and $-f_0$ where it is equal to $\frac{A}{2}$, as shown in Figure 6-a. We can arrive at the same result in a slightly different manner by noting that $g(t) = \frac{A}{2}e^{j\omega_0 t} + \frac{A}{2}e^{-j\omega_0 t}$, in which form g(t) does indeed appear to be the sum of two complex frequency generators f_0 and $-f_0$ with the same amplitude $\frac{A}{2}$. The samples $s(kT_e) = g(kT_e)p(kT_e)$ resulting from the product of $g(kT_e)$ by the samples $p(kT_e)$ from the window correspond to a digital Fourier transform $S_{num}(f)$ equal to the convolution product of $G_{num}(f)$ by $P_{num}(f)$: $S_{num}(f) = G_{num}(f) \otimes P_{num}(f)$ where the symbol $\otimes$ indicates the convolution product and $P_{num}(f)$ is the **digital Fourier transform** of the samples $p(kT_e)$ from the analysis window.

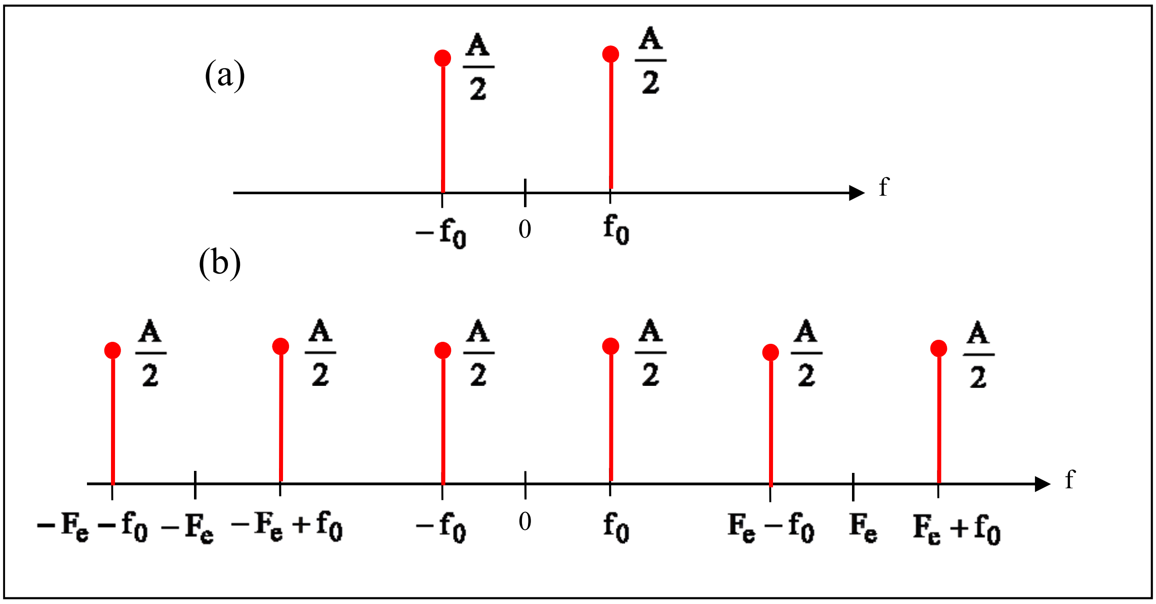


Figure 6: *(a) Fourier transform G(f) of g(t) and (b) digital Fourier transform* $G_{num}(f)$ *of* $g(kT_e)$

The Fourier transform of a pulse of width θ and amplitude A corresponds to the cardinal sine function: $A\theta\frac{\sin(\pi f\theta)}{\pi f\theta}$. To obtain $P(f)$, the Fourier transform for gate p(t), simply replace θ with NT_e from Figure 3. Figure 7 represents the module for $P_{num}(f)$, it is periodical with a period of F_e.

An analysis window is defined by:

- the width of the **main lobe**, $\frac{2}{NT_e}$, for a rectangular window. The width of the main lobe determines the **resolution**: i.e. the capacity to separate two frequency lines which are very close together.
- the amplitude of the **secondary lobes** determines the **dynamic range**: the capacity to identify two lines with very different frequencies and amplitudes.

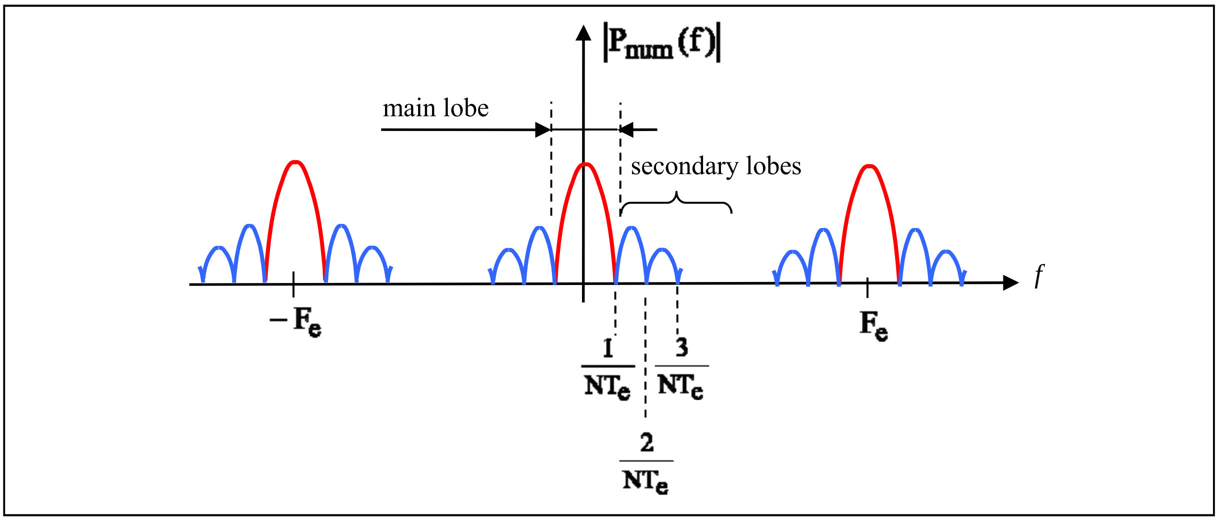


Figure 7: *Digital Fourier transform of the rectangular window*

Convolution $S_{num}(f) = G_{num}(f) \otimes P_{num}(f)$ consists of attaching to each line f_0, $F_e - f_0$, etc. the $P_{num}(f)$ pattern corresponding to the interval $[-\frac{F_e}{2}, \frac{F_e}{2}]$. Figure 8 shows the module $|S_{num}(f)|$ for the interval $[0, F_e]$. The N values of $|S(n)|$ are obtained by calculating $|S_{num}(f)|$ for $\frac{F_e}{N}, \frac{2F_e}{N}, \ldots \frac{(N-1)F_e}{N}$.

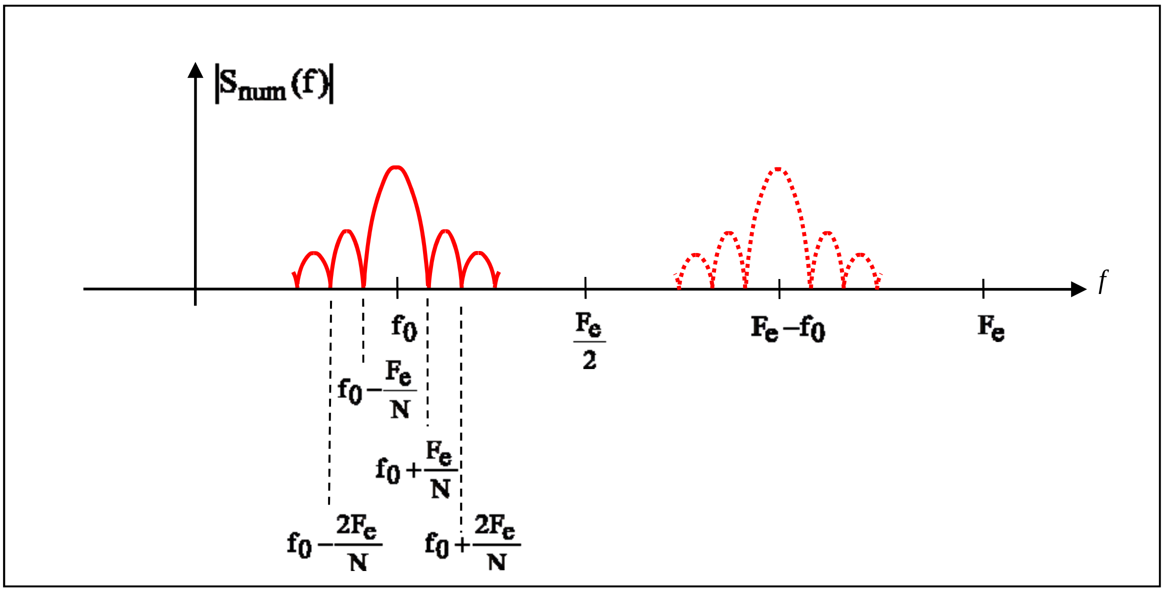
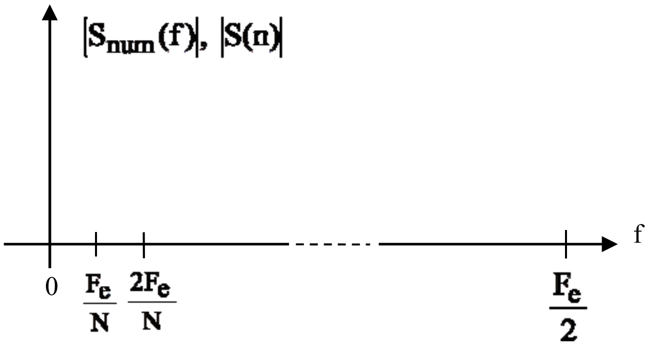
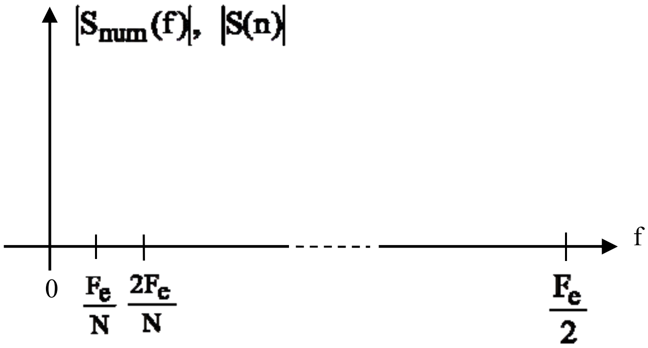


Figure 8: $|S_{num}(f)|$ *interval* $[0, F_e]$

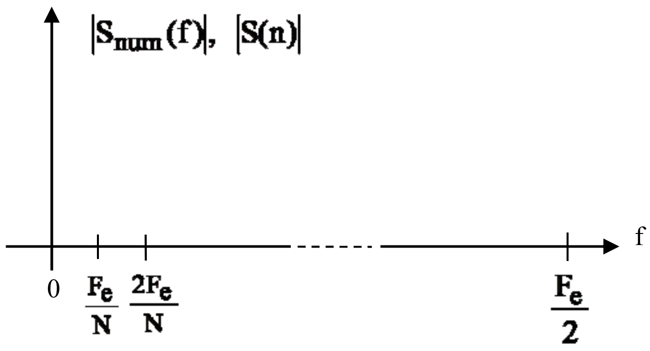
Question 6: The FFT algorithm calculates the values of $S(n)$ for frequencies $\frac{nF_e}{N}$ with $n = 0, 1, 2, ...(N-1)$. If f_0 is a multiple of $\frac{F_e}{N}$, complete this graph by roughly plotting $|S_{num}(f)|$ giving the values of $|S(n)|$.



Question 7: If f_0 is not a multiple of $\frac{F_e}{N}$, complete this graph by roughly plotting $|S_{num}(f)|$ giving the values of $|S(n)|$.



Question 8: The DSO5038A oscilloscope used in the practical assignment is capable of memorizing 1000 points. To use the FFT algorithm it adds 24 zeroes to make up the 1000 points. The calculation step $|S(n)|$ is thus $\frac{F_e}{1024}$. Complete the graph below by roughly plotting $|S_{num}(f)|$ giving the values of $|S(n)|$ for a system where $f_0 = 100kHz$ and $F_e = 2MHz$



If the frequency f_0 is not a multiple of the calculation step $\frac{F_e}{n}$, parasitic lines known as **spectral leaks** may appear. They will be more intense if the secondary lobes of the analysis window are big. The ideal solution is to set an analysis window with a narrow main lobe and secondary lobes with the smallest possible amplitude. For a given number of points (N), a rectangular window will give the tightest main lobe, and thus the highest resolution. But the secondary lobes are large, and may mask lines of lesser amplitude contained in the spectrum. The dynamic range of the rectangular window is therefore low. Other types of analysis window are available, offering smaller secondary lobes and better dynamic ranges, but their main lobes are wider and as such they offer lower resolution. Figure 9 shows the three windows available with the DSO5038A oscilloscope: rectangular, Hanning and Flat Top. The Hanning window has a better dynamic range than the rectangular window, the Flat Top window is used primarily for taking measurements.

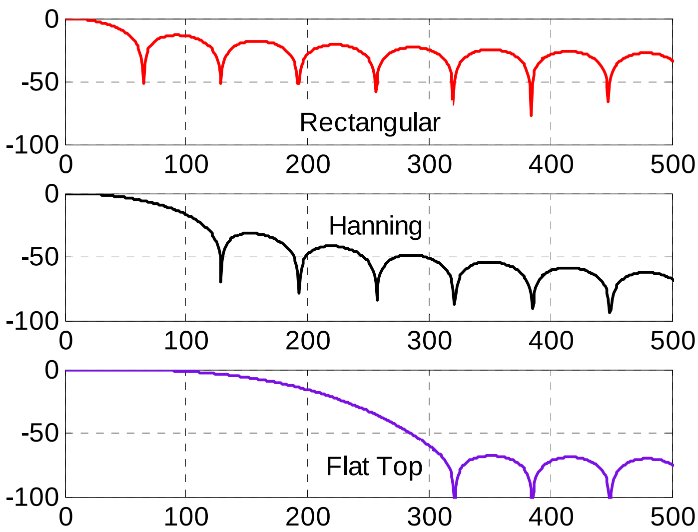


Figure 9: *Different analysis windows: Rectangular, Hanning and Flat Top*

IV - The impact of aliasing

A square wave with a frequency of $f_0 = 990\text{Hz}$, amplitude of $A = 1\text{V}$ and duty report of 50% is sampled without an anti-aliasing filter at a frequency of 100kHz. The Fourier series of this signal before sampling is written:

$$\frac{A}{2} + A\sum_{n=1}^{\infty} \frac{\sin\left(\frac{n\pi}{2}\right)}{\frac{n\pi}{2}} \cos(n\omega_0 t) = \frac{1}{2} + \sum_{n=1}^{\infty} \frac{\sin\left(\frac{n\pi}{2}\right)}{\frac{n\pi}{2}} \cos(n\omega_0 t) \qquad (10)$$

The Fourier transform (1) has the same lines as the Fourier series (10), but the amplitudes are divided by 2. Figure 10 shows the lines for $f = 31x990 = 30690\text{Hz}$ and $f = 33x990 = 32670\text{Hz}$, the line at $f = 32x990 = 31680\text{Hz}$ is nil based on ratio (10).

Question 9: Check that the amplitudes in Figure 10 are correct.

Question 10: After sampling the spectrum contains more lines, between 30690Hz and 32670Hz, for example, there is an anti-aliasing line as shown in Figure 11. Give its frequency and amplitude in dBV.

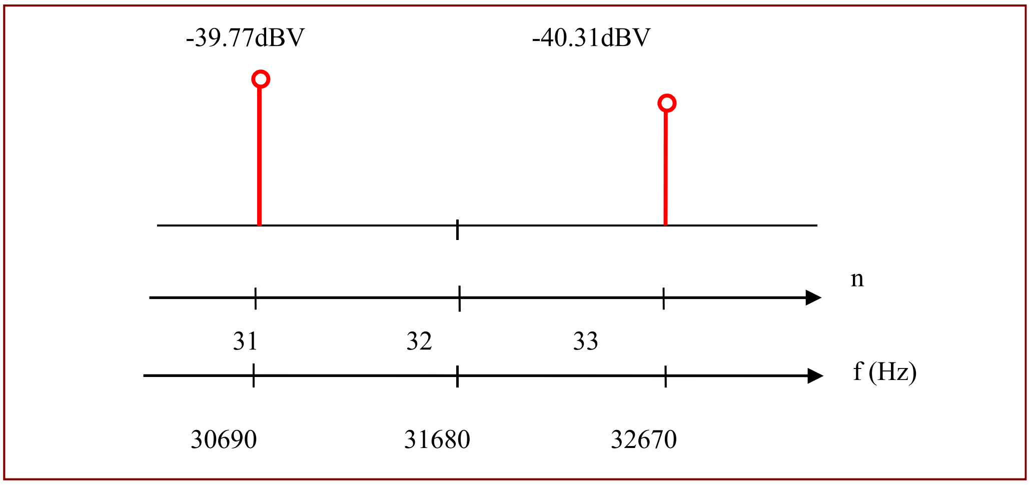


Figure 10: *Harmonics* $n = 31$*, 32 and 33 of a rectangular signal with a duty cycle of 50%*

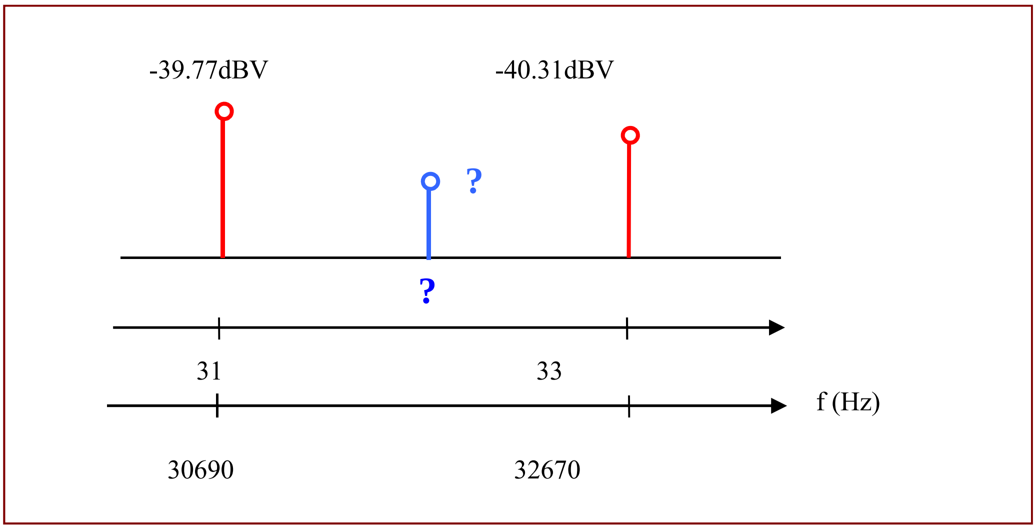


Figure 11: *Frequency and amplitude of the anti-aliasing line between harmonics* $n = 31$ *and 33*