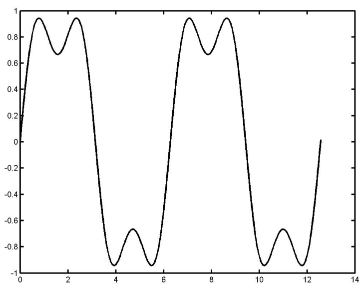
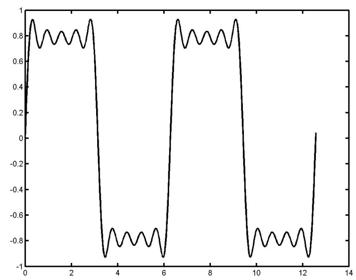
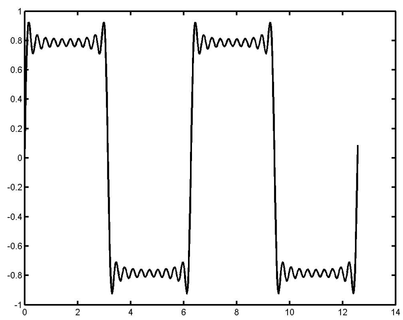
ИТМО

И.А. Коняхин

DIGITAL SIGNAL PROCESSING. BASIC PROCEDURES

Санкт-Петербург
2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

И.А. Коняхин

DIGITAL SIGNAL PROCESSING. BASIC PROCEDURES

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 12.04.02 Оптотехника
в качестве Учебно-методическое пособие для реализации основных профессиональных образовательных программ высшего образования магистратуры

Санкт-Петербург
2023

Коняхин И.А., Digital signal processing. Basic procedures – СПб: Университет ИТМО, 2023. – 61 с.

Рецензент:
Лукьянов Геннадий Николаевич, доктор технических наук, профессор, доцент (квалификационная категория "ординарный доцент") факультета систем управления и робототехники Университета ИТМО.

В учебно-методическом пособии изложены общая структура и методики расчёта параметров двух базовых процедур цифровой обработки сигналов: определения спектра типовых сигналов с помощью дискретного преобразования Фурье и преобразования сигнала (фильтрации) линейными элементами систем. Подробно анализируются эффекты, искажающие результат гармонического анализа при его реализации дискретными вычислительными процедурами, а также расчётные и методические способы минимизации возникающих ошибок. Цифровая обработка сигнала линейными элементами (КИХ фильтрация) рассмотрена в двух вариантах: по методу дискретной свёртки во временной области и методу дискретной свёртки через пространство частот. Подробно рассматриваются особенности вычисления параметров процедур дискретной свёртки при обработке основных видов сигналов: импульсных, периодических, апериодических импульсных. Приводятся примеры расчёта параметров и программной реализации дискретных процедур в технологии Matlab.
Пособие предназначено для студентов, обучающихся по программе магистерской подготовки по направлению 12.04.02 Прикладная оптика, дисциплина «Цифровая обработка сигналов»/“Digital signal processing” (преподавание ведётся на английском языке).

Table of Contents

Table of Contents 3
Glossary of Abbreviations and Notation 5
Introduction 6
1 Discrete Fourier Transform as tool of the signal spectrum analyzing 8
1.1 Discrete Fourier Transform as result of standard transform revising 8
1.2 Sampling the Signal in the Spatial (time) Domain by the Time Quant 10
1.3 Previous value of the Signal Sampling Period 15
1.4 Signal Sampling Period of Infinite Aperiodic Signals. Leakage phenomenon 15
1.5 Sampling Period of the Periodic Signal 16
1.6 Sampling Period of a Single Impulse. Zero padding approach 17
1.7 Calculation the Number *N* of the Samples in the Sampling Period *T*. Rules of the Fast Fourier Transform Algorithm 18
1.8 Calculation the Number of Samples in the Signal Array 19
1.9 Restoration the conventional viewing of the spectrum. Mirror Phenomena 20
1.10 Example of an exercise 20
Test questions 25
2 Discrete Convolution in the Spatial Domain as a procedure of the Signal Transform by Linear Space-Invariant System 26
2.1 Discrete Convolution as a result of standard transform revising 26
2.2 Sampling the Input Signal and the Impulse Response Function by the Time Quant 27
2.3 Calculation of the Sampling Period of the Impulse Response Function. Number of Samples. Fundamental property of Impulse Response Function 27
2.4 Calculation of the Sampling Period of the Impulse Input Signal. Number of Samples 28
2.5 Calculation of the Sampling Period of the Periodic Input Signal. Number of Samples 29
2.6 Calculation of the Sampling Period of th eAperiodic Infinite Input Signal. Number of Samples 31
2.7 Example of exercise 32
Test questions 37
3 Discrete Convolution in the Frequency Domain as a procedure of the Signal Transform by Linear Space-Invariant System 39
3.1 Convolution in the Frequency Domain Algorithm 39
3.2 Convolution Periodic Input Signal in the Frequency Domain 40
3.3 Convolution Aperiodic Infinite Input Signal in the Frequency Domain 42
3.4 Convolution of a Single Impulse Input Signal in the Frequency Domain 43
3.5 Example of exercise 44
Test questions 47

Conclusion...48
References...49
Appendix 50

Glossary of Abbreviations and Notation

DC – Discrete convolution
DFT – Discrete Fourier transformation
DSP – digital signal processing
FDFT – Fast discrete Fourier transformation
LSI system – Linear Space-Invariant system

Introduction

The great advancements in the design of microchips, digital systems and computer hardware over the past 60 years gave rise to digital signal processing (DSP) which has grown over the short time into a universal, multi-faceted, and developed subject of study. The DSP has been applied in many disciplines ranging from engineering to economics and from astronomy to social sciences. A complete analysis of DSP aspects would require many volumes of publications, thus, this guidebook focuses mainly on the fundamental DSP operations, namely, on the representation of signals by mathematical models and on their processing by discrete-time systems. DSP is considered with regard to Linear Space-Invariant (LSI) systems used in many technical applications. Quite a few types of processing are possible for signals, with DSP almost always being a linear operation, involving reshaping, transforming, or manipulating the frequency spectrum of the signal.

Chapter 1 describes Fourier Transform and Discrete Fourier Transform as the principal mathematical instruments for analyzing the spectral characteristics of continuous-time signals. The Discrete Fourier Transform is deduced from the continuous Fourier Transform through a sampling process. Besides, Chapter 1 presents the Discrete Fourier transform (DFT) and the related Fast Fourier Transform (FFT) method as mathematical tools for the analysis of signals and for implementation of digital filters. The properties of the DFT and the relations between the DFT and the continuous Fourier transform are discussed. These relations have many properties that are far from being obvious, and using Discrete Fourier Transform and Fast Fourier Transform is likely to end up with inaccurate spectral representations for the signals if those properties are ignored. The chapter also deals with the "window" and the "zero padding" approaches, which can facilitate processing signals of infinite and finite duration.

Chapter 2 deals with the fundamental properties of Linear Space-Invariant systems as the basis of the non-recursive filters. The topics considered include linearity, time invariance, causality as the basic LSI system properties. Convolution equations are used as the main mathematical model of an LSI system. According to the convolution mathematical model, the LSI system is analyzed as a discrete object. Chapter 2 discusses two discrete convolution models. One of them is the discrete convolution in the spatial domain, and the other one is convolution in the frequency domain. Some properties of the LSI discrete convolution model that are far from being evident, require a detailed consideration of the relation between the input signal and the impulse response of the LSI system.

The most important mathematical tool for representing and processing discrete-time signals is the convolution in frequency domain. The latter form is the subject of Chapter 3. The chapter also deals with the convolution in frequency domain as a discrete mathematical model of LSI systems and of non-recursive filters. According to the convolution in frequency domain the LSI system is analyzed as a discrete object by using the non-recursive algorithms. Chapter 3 discusses three kinds of non-recursive algorithms. One of them is the non-recursive algorithm as a transformation of a periodic input signal. Another non-recursive algorithm class is the transformation of an aperiodic infinite signal. Some properties of the third algorithm that are far from being obvious require a detailed consideration of the relation between the input impulse and the impulse response of the LSI system.

The guidebook is recommended for the students of Digital Signal Processing in Optoelectronics (within the course in Optical Engineering as part of the Master's Program 12.04.02 Applied Optics), and those of Optic-electronic systems simulation and research (within the course in Electronic and Optic-electronic systems of the Optical Engineering Program 12.05.01).

The applied part of this course implies calculating the parameters of digital transformations. It includes three exercises as follows:

1. A digital Fourier transform as the tool of the signal analysts.
2. A digital convolution in Spatial Domain as a tool of the Linear Space-Invariant (LSI) systems simulation and non-recursive filter synthesis.
3. A digital convolution in Frequency Domain as a tool of the Linear Space-Invariant (LSI) systems simulation and non-recursive filter synthesis.

Every exercise is supposed to take 4 academic hours. Exercise 1 requires 1 academic hour for a study of the basic theoretical provisions, 2 hours are allocated for creating an algorithm, writing and debugging the code, and 1 hour is left for analyzing and processing the results. Results are assessed using a student's report in accordance with the requirements outlined in the relevant section of the guide, as well as its approval by the teacher. The purpose of the report defense is to evaluate the competencies and skills acquired and mastered by the student within the course.

1 Discrete Fourier Transform as tool of the signal spectrum analyzing

1.1 Discrete Fourier Transform as result of standard transform revising

The purpose of Fourier Transform is to extract some information from a signal (for example, an image) and to prepare it for a certain task, for example, filtering, transforming, viewing or transmitting.

Signal/noise ratio may be rather small, so the signal may need to be preprocessed. Preprocessing can be done in the Spatial or in the Frequency domain using a variety of techniques. Fourier transform is the main instrument for processing in the Frequency domain.

Many algorithms based on Fourier transform can be applied for processing in the Spatial domain as well.

Wavelet, Cosine, Walsh, Hadamard, and other famous transforms have the Fourier transform properties.

Fourier transform is the fundamental procedure of harmonic analysis..

An unlimited non-periodic signal $g(t)$ (Figure 1.1a) in the time domain t is transformed by a standard (also referred to as *analog*) Fourier Transform, which is defined as an integral procedure:

$$S(f) = \int_{-\infty}^{\infty} g(t)\exp(-i \cdot 2\pi f t)\ dt, \qquad (1.1)$$

where $S(f)$ is the spectrum of the signal (Figure 1.1b), i is an imaginary unit, f is the frequency (Hz). Spectrum $S(f)$ is limited by the interval $[-f_{max},+f_{max}]$, f_{max} is the largest frequency in the spectrum.

Inverse Fourier transform restore the signal $g(t)$:

$$g(t) = \int_{-\infty}^{\infty} S(f)\exp(i \cdot 2\pi f t)\ df\ , \qquad (1.2)$$

Digital processing approach can be implemented only to sampled signals, therefore it is necessary to take the following steps.

1. Sampling the signal $g(t)$ in the Spatial domain by Space Δt (also referred to as *discrete time quant*) (Figure 1.1a)

2. Limiting the signal $g(t)$ by a time sampling period T (Figure 1.1a)

3. Sampling the spectrum $S(f)$ in the Frequency domain by Space Δf (*discrete frequency quant*) (Figure 1.1b).

4. Limiting the spectrum $S(f)$ by a frequency sampling period F (Figure 1.1b).

These actions result in two arrays: $g(k)$, k= 1..N is the array of the sampled signal (Figure 1.1c) and $S(n)$, n = 1..N is the array of the sampled spectrum (Figure 1.1d). Here N is the number of samples in the arrays $g(k)$ of the sampled signal $g(t)$ and $S(n)$ of the sampled spectrum $S(f)$.

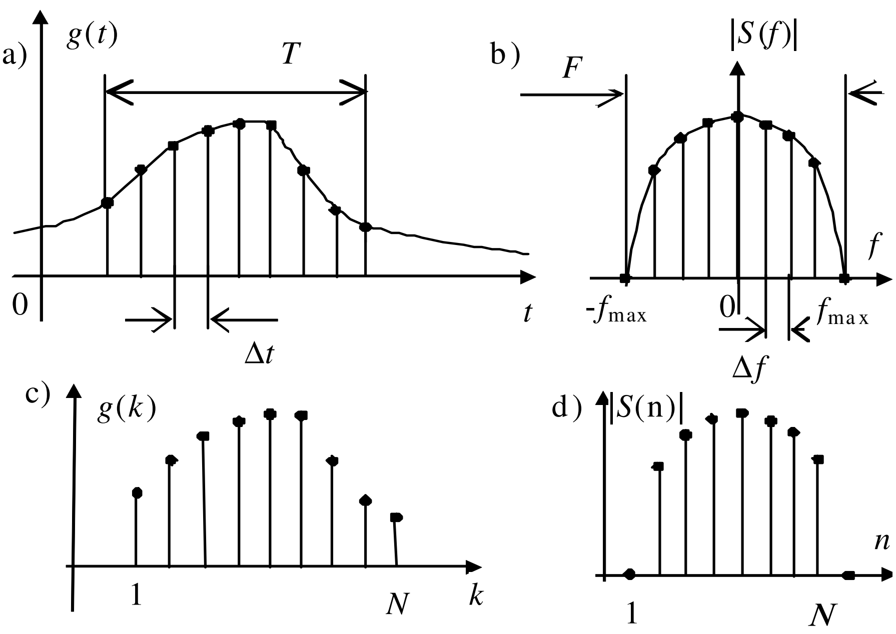


Figure 1.1 – Limitation and sampling of the signal and its corresponding spectrum: a) unlimited non-periodic signal; b) spectrum; c) array of sampled signal; d) array of sampled spectrum

The fundamental formula of the sampling processes is as follows:

$$\Delta t \le \frac{1}{2 f_{\max}}, \tag{1.3}$$

where Δt is the time quant of the sampled signal $g(t)$. This formula is derived from Nyquist-Shannon-Kotelnikov theorem. According to the theorem, signal information is not lost after sampling if the time quant Δt meets the condition (1.3). Parameter $f_{\max}$ of the theorem has a special name: *Nyquist frequency of non-sampling signal* $g(t)$.

Besides, the sampling process is described by formulas:

$$\Delta f = \frac{1}{T}, \tag{1.4}$$

$$F = \frac{1}{\Delta t}, \tag{1.5}$$

$$f_{max} = \frac{F}{2}, \tag{1.6}$$

$$F = (N-1) \cdot \Delta f , \tag{1.7}$$

$$N = \frac{T}{2} + 1. \tag{1.8}$$

The array $g(k)$ of the sampled signal can be transformed into the array $S(n)$ of the spectrum by Discrete Fourier Transform (DFT) which is defined as:

$$S(n)=\sum_{k=0}^{N-1} g(k)\cdot \exp\left(-i\cdot 2\pi\cdot \frac{k\cdot n}{N}\right) \quad (1.9)$$

Inverse Discrete Fourier Transform (IDFT) restores the array $g(t)$ of the sampled signal:

$$g(k)=\frac{1}{N}\cdot \sum_{n=0}^{N-1} S(n)\cdot \exp\left(i\cdot 2\pi\cdot \frac{k\cdot n}{N}\right) \quad (1.10)$$

In formulas (1.9), (1.10) $g(k)$, k= 1..N is the array of the sampled signal $g(t)$, and $S(n)$, n = 1..N is the array of the sampled spectrum $S(t)$.

1.2 Sampling the Signal in the Spatial (time) Domain by the Time Quant

The sampling process transforms the features of the signal and the spectrum.

Firstly, the spectrum is transformed into an unlimited process on the frequency scale (Figure 1.2a). As result, the limit of f_{max} becomes the infinity [1, 2] :

$$lim(f_{max})=\infty . \quad (1.11)$$

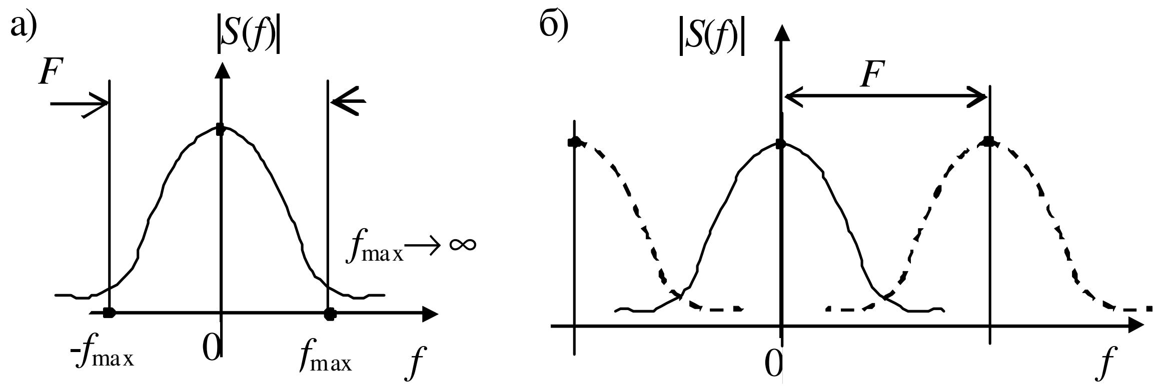


Figure 1.2 – Transformations spectrum as result of sampling: a) f_{max} of sampling spectrum; b) sampling spectrum

Therefore, space Δt cannot be calculated according to the Nyquist-Shannon-Kotelnikov theorem (1.3) in digital processing.

Secondly, the sampled signal $g(k)$ and the spectrum $S(n)$ are processed as periodic objects. In digital processing the sampled signal has the period equal to the sampling period T, and the sampled spectrum has the period equal to the sampling period F, respectively (Figure 1.2b).

The changes considered result in an error in digital processing. The result of the digital processing is distorted due to the fact that the adjacent periods of the discrete spectrum are combined and deformed (Figure 1.3). This phenomenon is called *aliasing* [3, 4].

In signal processing applications the time quant Δt is calculated by the formula:

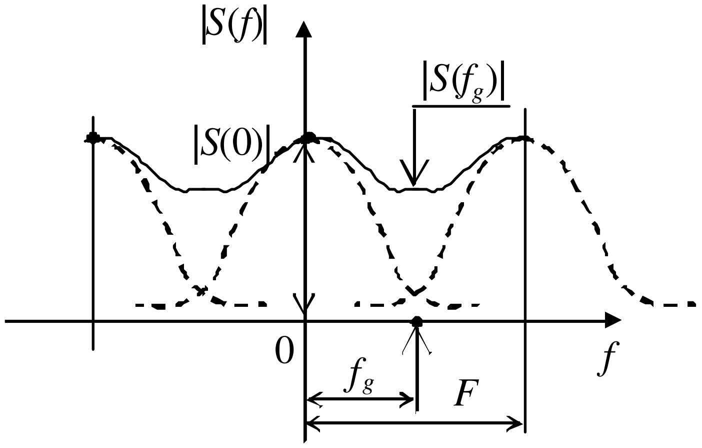


Figure 1.3 – Aliasing error explaining

$$\Delta t = \frac{1}{2f_g}, \qquad (1.12)$$

where f_g is the Nyquist frequency of the sampled spectrum.

Frequency f_g fixes the point of the lager aliasing (Figure 1.3). Then f_g can be defined as:

$$\Delta t = \frac{F}{2}, \qquad (1.13)$$

where F is the sampling period of the spectrum.

The power of aliasing is measured by the coefficient K_{al}, which is defined as:

$$K_{al} = \frac{|S(f_g)|}{|S(0)|}, \qquad (1.14)$$

where $S(f_g)$ is the value of the spectrum if the frequency is equal to the Nyquist frequency f_g, of the sampled spectrum, $S(0)$ is the value of the spectrum at the frequency $f = 0$.

If the spectrum $S(f)$ of the signal at the frequency $f = 0$ has value $S(f) = 0$, then we recommend using the largest value of the spectrum $S(f)$ at the frequency f closest to $f = 0$.

The main steps of the calculations are:

- choosing the acceptable value of the coefficient K_{al};
- calculating the Nyquist frequency of the sampled spectrum f_g as the root of the equation

$$20 \cdot \lg(K_{al}) = B \cdot \lg\left(\frac{2 \cdot f_g}{f_c}\right) + C, \qquad (1.15)$$

-calculating the proper time quant Δt using the formula (1.12).

In Equation (1.15) B and C are coefficients, f_c is the cutting frequency.

The cutting frequency f_c is the point of interest on the frequency axis of the graphic spectrum $S(f)$ in the logarithmic decimal scale.

The cutting frequency f_c is defined by a 2-step algorithm.

1. The graph of the half spectrum $S(f)$ for f > 0 (Figure 1.4a) is re-plotted in another scale. Argument axis becomes the decimal logarithm of frequency lg(f) , where as the function axis becomes the spectrum module $|S(f)|$ in decibel units according to rule (Figure 1.4b)

$$|S(f)|\,dB = 20\cdot\lg\left(\frac{|S(f)|}{|S(0)|}\right), \qquad (1.16),$$

where $S(0)$ is the value of the spectrum at the frequency $f=0$.

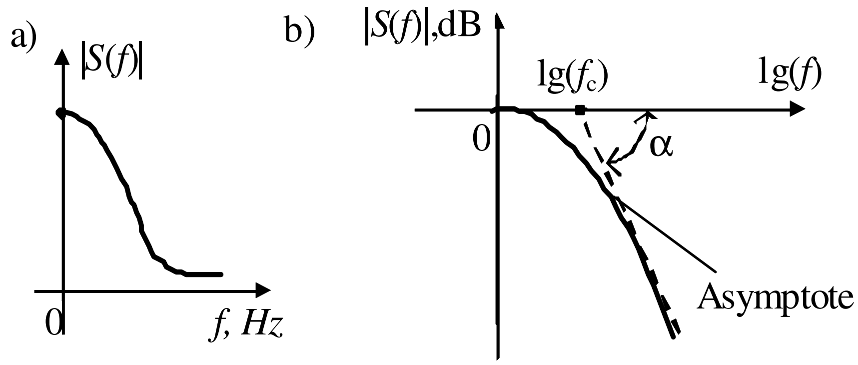


Figure 1.4 – Cutting frequency calculation

2. The graph of the spectrum module $|S(f)|$ in decibel units has a curve part and a pseudo-line part. The line asymptote of the pseudo-line part and the argument axis lg(f) have a point of intersection (Figure 1.4b, dashed line). This point defines the logarithm of cutting frequency lg(f_c).

Angle α of asymptote is calculated by the formula:

$$tan(\alpha) = 20\cdot r\ \frac{dB}{dec}, \qquad (1.17)$$

where r is the rank of the spectrum.

The coefficients B,C in equation (1.15) and formulae for calculating the cutting frequency f_c are the functions of the rank r of the spectrum (Table 1.1).

In Table 1.1 τ_p is the duration of the impulse:

$$\tau_p = t_2 - t_1, \qquad (1.18)$$

where t_1 and t_2 indicate the time of the beginning and the end of the impulse (Figures 1.5, 1.6, 1.7)

Parameter τ_e is the effective duration of the unlimited non-periodic signals (Figures 1.8,1.10,1.11).

Rank r of the spectrum is determined at the first step of the calculation to meet the features of the signal $g(t)$ (Table 1.1).

Table 1.1 – Coefficients B,C and formulae for calculating the cutting frequency f_c

Signal $g(t)$ features	Spectrum rank n_0	B	C	Cutting frequency fc	
				Single Impulse or Periodical signal $g(t)$	Aperiodic infinite impulse signal $g(t)$
1	2	3	4	5	6
Discontinuity of the signal (Figures 1.5,1.8)	1	-19,5	8,58	$\frac{1}{\pi \cdot \tau_p}$	$\frac{1}{2\pi \cdot \tau_e}$
Discontinuity of the signal derivative (Figures 1.6,1.10)	2	-39,55	14,61	$\frac{1}{2\pi \cdot \tau_p}$	$\frac{1}{4\pi \cdot \tau_e}$
"Smooth" signal (Figures 1.7,1.11)	3	-57,62	18,69	$\frac{1}{3\pi \cdot \tau_p}$	$\frac{1}{6\pi \cdot \tau_e}$

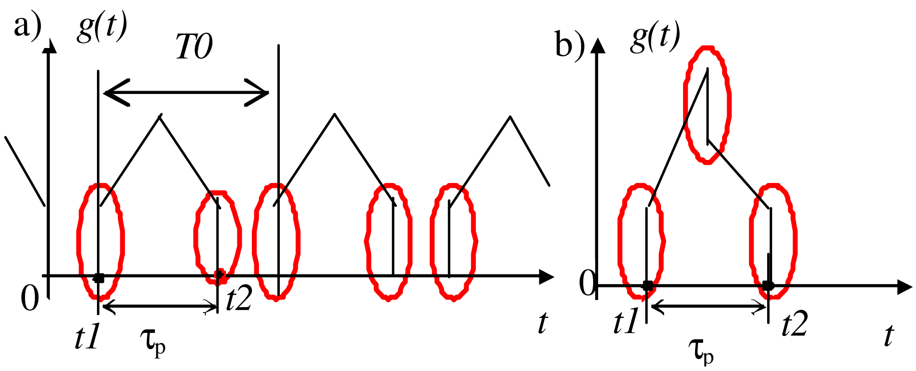


Figure 1.5 – Discontinuity of the signal (circle marks): a) periodic signal, b) impulse

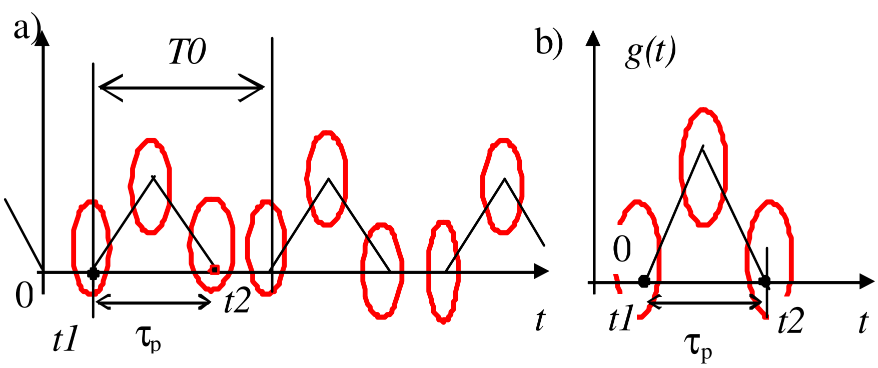


Figure 1.6 – Discontinuity of the signal derivative (circle marks): a) periodic signal, b) impulse

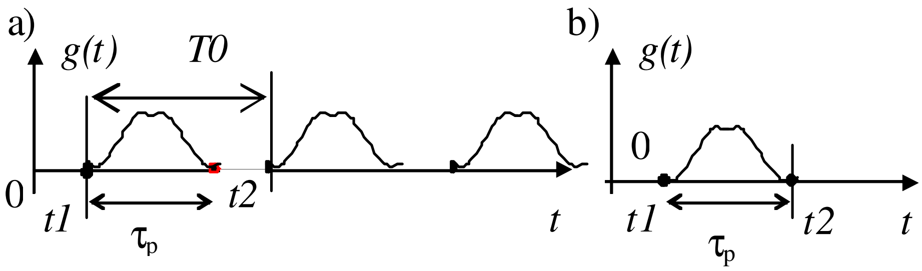


Figure 1.7 – “Smooth” signal: a) periodic, b) Impulse

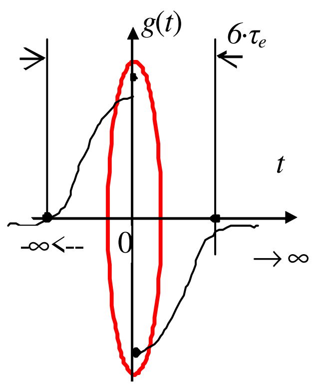


Figure 1.8 – Discontinuity of the infinite signal (circle mark)

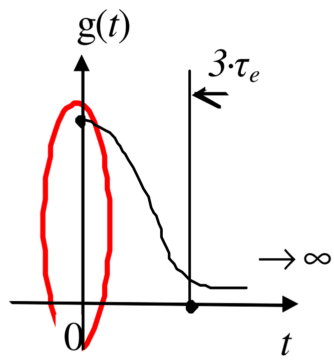


Figure 1.9 – Discontinuity of one-side infinite signal (circle mark)

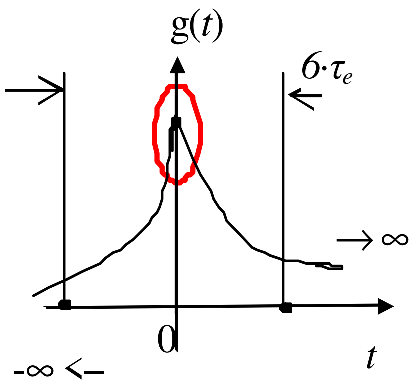


Figure 1.10 – Discontinuity the derivative of infinite signal (circle mark)

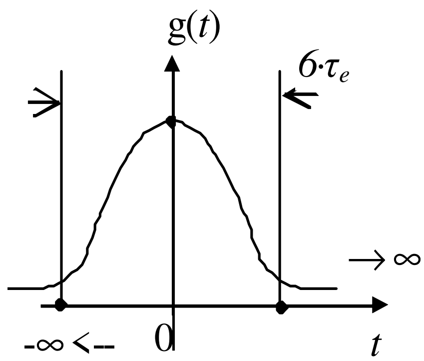


Figure 1.11 – “Smooth” infinite signal

Thus, the calculation of the time quant Δt of the sampling signal $g(t)$ includes 5 stages.

1. Choosing the coefficient K_{al} of the acceptable aliasing
2. Analyzing the graph of the signal $g(t)$ and choosing the rank r of the spectrum.
3. Calculating the cutting frequency f_c using the formulae in Table 1.1 and choosing the values of coefficients B,C.
4. Calculating Nyquist frequency f_g as the root of the equation (1.15).

5. Calculating the time quant Δt using the formula (1.12).

1.3 Previous value of the Signal Sampling Period

The signal sampling period *T* (also referred to as the *interval of signal limiting*) is defined in two stages: first, a preliminary value of *T%* is calculated based on the required value Δf of the sampling interval in the spectrum; second, the preliminary value *T%* is updated according to the type of signal.

Preliminary value of *T%* of the signal sampling period is determined by the formula following (1.4):

$$T\% = \frac{1}{\Delta f}, \tag{1.19}$$

where Δf is the sampling interval of the spectrum, which actually determines the resolution in the frequency domain. As a rule, the required resolution Δf is set based on the envisaged application of the signal processing procedure for a specific project.

Formula (1.19) defines the minimum value of the signal sampling period; if a smaller value is selected, part of the spectrum will be lost as a result of DFT. This phenomenon is referred to as “picket-fence-effect” [5].

1.4 Signal Sampling Period of Infinite Aperiodic Signals. Leakage phenomenon

The updated sampling period *T* has to meet the 3 conditions as follows:

$$T = m \cdot \tau_e, \tag{1.20}$$

$$m \geq 6, \tag{1.21}$$

$$m \geq 3, \tag{1.22}$$

where *T%* is the preliminary value as defined (1.19), τ_e is the effective duration of infinite aperiodic signals (Figures 1.8,1.10,1.11,1.11).

Condition ((1.21) is used for the unlimited signals specified on the time interval (-∞,+∞), and expression (1.22) is used for one-side unlimited signals specified on [0,+∞), for instance, as in Figure 1.9.

Limiting the signal *g*(*t*) by the sampling period *T* results in the generation of some discontinuity in the border points of the interval *T* (Figure 1.12a). These discontinuities distort the DFT result: firstly, decreasing the value of the samples in the low-frequency part of the spectrum and, secondly, generating parasitic oscillation in the high-frequency part (Figure 1.12b). The spectrum in the Figure is shown by a solid line before and a dashed line after the signal limitation. This phenomenon of oscillation generation in the spectrum is referred to as *leakage* [6].

To reduce the error due to the leakage, it is necessary to smooth the discontinuities of the signal by multiplying the original signal *g*(*t*) by the special "window" function *W*(*t*):

$$g(t)_w = g(t) \cdot W(t), \tag{1.23}$$

where $g(t)_w$ is the smoothed signal without any discontinuity at the border points of the limiting interval *T* (Figure 1.12a, dashed line).

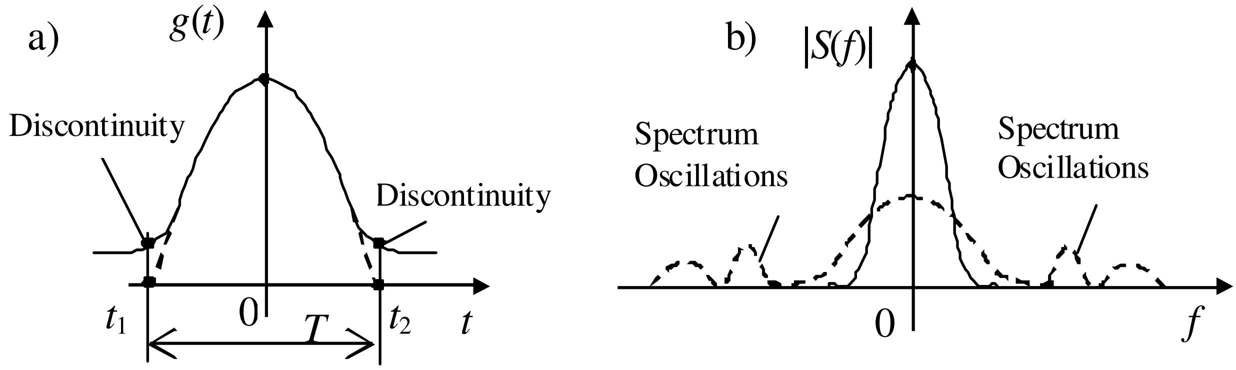


Figure 1.12 – Discontinuity of the signal and the spectrum oscillations as a result of limiting by interval *T*

The “window” function is a smooth function whose values are close to one in the middle of the interval *T*, and gradually decrease to zero at the edges of the interval.

There are many "window" functions, however, the simplest of them are the Tukey "window" , which is a segment of a sine function on the edges of interval *T*, and is close to the value of one in the middle part (Figure 1.13a) and the Hann (Hanning) "window", roudly corresponding to the function $\cos^2(x)$ – Figure 1.13b. Hann “window” is more effective than the Tukey “window" to eliminate parasitic oscillations in the high-frequency part of the spectrum, however, it adds distortions in the low-frequency part.

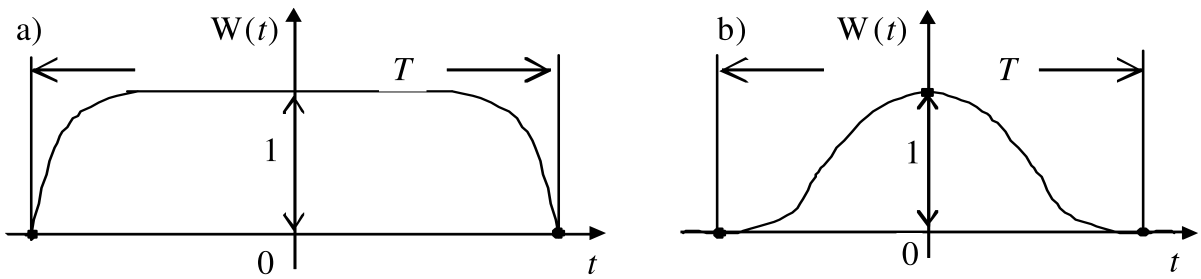


Figure 1.13 – Tukey and Hann “windows”

1.5 Sampling Period of the Periodic Signal

The updated sampling period *T* has to meet the 2 conditions as follows:

$$T = m \cdot T0, \qquad (1.24)$$

where *T%* is the preliminary value ss defined by (1.19), *T*0 is the period of the signal *g*(*t*) (Figure 1.14), *m* is integer.

It is important that the number *m* of periods T0 is integer. Ignoring this condition will lead to a DFT error, because in that case the sampled signal would be processed as a periodic object on the sampling period *T*. It will generate some additional discontinuity of the sampled signal. (Figure 1.14b).

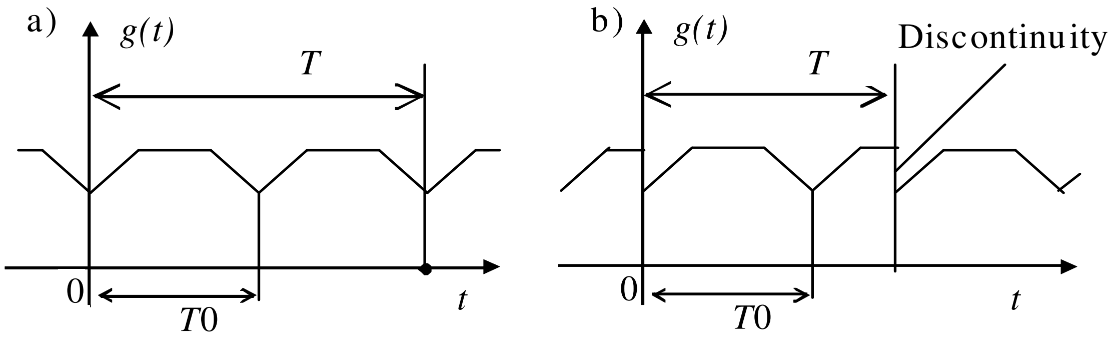


Figure 1.14 – Choosing sampling period *T* of periodic signal *T*

1.6 Sampling Period of a Single Impulse. Zero padding approach

The updated sampling period *T* has to meet the 2 conditions as follows:

$$T \geq \tau_p, \qquad (1.25)$$

where $\tau_p = t_2 - t_1$ is the impulse duration (Figures 1.15,1.16).

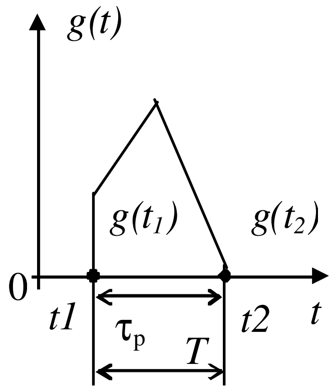


Figure 1.15 – First option of the chosen sampling period

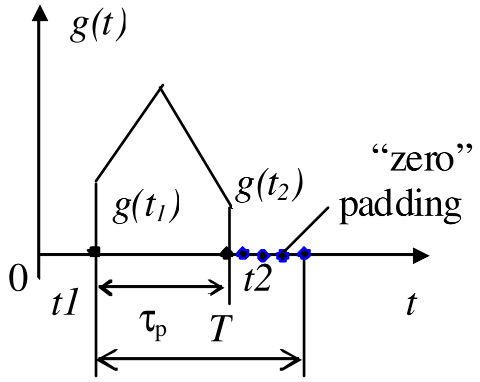


Figure 1.16 – Second option of the chosen sampling period

Condition (1.25) has two options.

Firstly, if the values $g(t_1), g(t_2)$ of signal in edge points t_1 and t_2 meet to condition (for example, Figure 1.15):

$$g(t_1) \cdot g(t2) \leq 0, \qquad (1.26)$$

the sampling period *T* has to be chosen as equal to the impulse duration τ_p ((unless it mismatches the rule (1.19)):

$$T = \tau_p, \qquad (1.27)$$

Secondly, if the values $g(t_1), g(t_2)$ of the signal in the edge points t_1 and t_2 meet the condition:

$$g(t_1) \cdot g(t2) > 0, \qquad (1.28)$$

the sampling period T is to be assembled of two parts. The first part of the signal array is the sampled impulse, and the second part is to be filled with zero samples (Figure 1.16). This method is referred to as "*zero padding*" [7].

The sampling period T in the zero padding procedure is calculated by the formula:

$$T = \tau_p + \frac{\tau_p}{2}, \text{ or } T = \tau_p + \frac{\tau_p}{4}. \qquad (1.29)$$

If zero padding is ignored, the result of DFT will be highly distorted.

Ignoring the zero padding condition will result in a DFT error, because in that case the sampled signal will be processed as a periodic object on the sampling period T. As a result, the natural discontinuity of sampled signal will be radically distorted – compare Figure 1.17 and Figure 1.18.

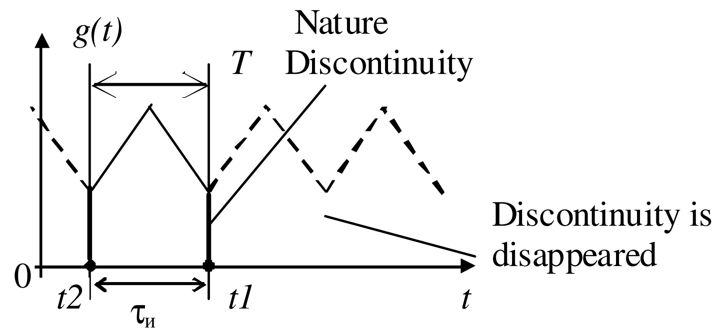


Figure 1.17 – Zero padding rule is ignored

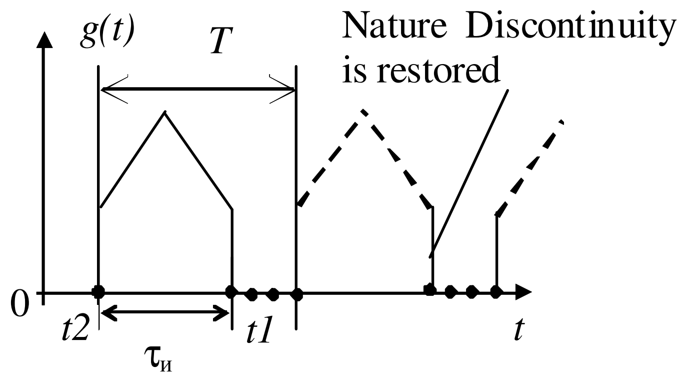


Figure 1.18 – Sampling period *T* matches the Zero padding rule

1.7 Calculation the Number *N* of the Samples in the Sampling Period *T*. Rules of the Fast Fourier Transform Algorithm

The number N of sampling is defined in three stages: first, a preliminary value of the full number $N\%$ is calculated based on the required value T of the sampling period;

second, the preliminary value $N\%$ is updated according to the rule of fast Fourier algorithm; third, the number of sampling in the impulse part of the signal is calculated.

1. The preliminary number $N\%$ of sampling is calculated by the formula:

$$N\% = \frac{T}{\Delta t} + 1, \tag{1.30}$$

where T is the sampling period, Δt is the time space (time quant).

2. The updated number N of sampling is calculated according to the 3 rules of the Couley-Tyukey algorithm of the Fast Discrete Fourier Transform (FDFT)[8]:

$$N \geq N\%, \tag{1.31}$$

$$N = 2^{M}, \tag{1.32}$$

$$N > 2^{M-1}, \tag{1.33}$$

where M integer. For example, if $N\% = 78$, is to be chosen $2^7 = 128$, but not $2^6 = 64$.

1.8 Calculation the Number of Samples in the Signal Array

The array of the infinite aperiodic signal has the number NI of samples, the same as the number N in the sampling period. The number of the samples LB corresponding to its effective duration τ_e is calculated as:

$$LB = \frac{\tau_e}{\Delta t}, \tag{1.34}$$

where τ_e is the effective duration of aperiodic signal, Δt is the space of sampling (the time quant)

The number of samples LI corresponding to the duration τ_p a single impulse is found as:

$$LI = \frac{\tau_p}{\Delta t}, \tag{1.35}$$

where Δt is the space of sampling (time quant)

The periodic signal has addition parameter SKV referred to as *period-to-pulse duration ratio*:

$$SKV = \frac{T0}{\tau_p}, \tag{1.36}$$

where $T0$ is the period, τ_p is the duration of the impulse (Figure 1.5).

The number of samples NI corresponds to one period $T0$ and the number LI corresponds to the duration τ_p of the impulse in one period:

$$NI = \frac{N}{m}, \tag{1.37}$$

$$LI = \frac{NI}{SKV}, \tag{1.38}$$

These formulas are to be used to calculate the number of samples in an array of the signal if the number of samples N in the sampling period of the signal is to be calculated.

1.9 Restoration the conventional viewing of the spectrum. Mirror Phenomena

A feature of the DFT is the unconventional placement of samples in the resulting array of spectrum. In the conventional form, the spectrum values corresponding to negative frequencies are located to the left of frequency $f = 0$, and the values for positive frequencies are located to the right (Figure1.19a).

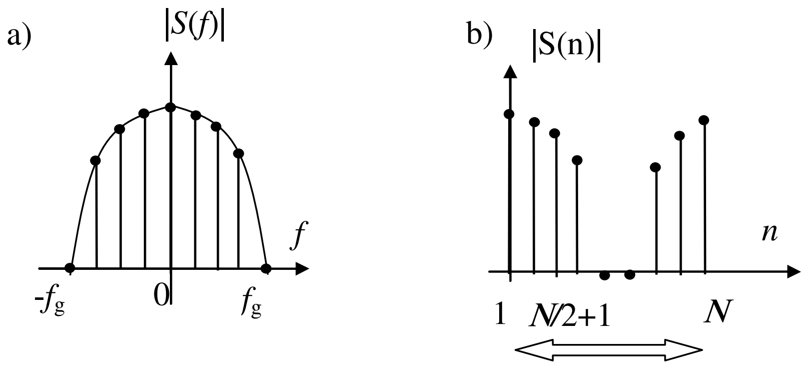


Figure 1.19 – Mirror phenomena in spectrum array

In the spectrum array, as a result of the DFT, the placement of the spectrum samples is different: the spectrum samples corresponding to negative frequencies are located to the right relative to the samples for positive frequencies (Figure 1.19b). This feature is called the *mirror* phenomenon.

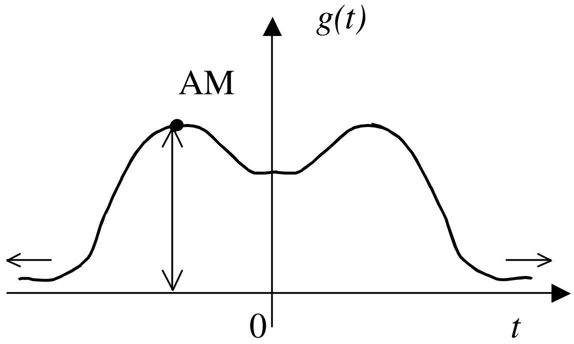
Thus, convention view will be restored if the left and right parts of the spectrum array S(n) is rearranged (see the arrow in Figure 1.19b)).

1.10 Example of an exercise

Example of the task

```
%Digital signal processing in optoelectronics
%Laboratory Exercise 1 (version 3)
%Fundamental properties and main equation for calculate
the parameters Discrete Fourier Transform
%
%  General Instructions
%
% 1.Calculate the parameters of Discrete Fourier Transform
procedure
% 2. Create the program code (in MatLab technology) and
execute this procedure.
```

```
% 3. Getting the following graphics
% 3.1. Sampling Signal in Spatial Domain
% 3.2. Updating Sampling Signal in Spatial Domain
% (optional)
% 3.3. Amplitude Spectrum in Frequency Domain
% 3.4. Phase Spectrum in Frequency Domain
% 3.5. Mirror phenomena correction of Amplitude Spectrum
% 3.6. Mirror phenomena correction of Phase Spectrum
% 3.7. Energy Spectrum
%
%                  Objectives
%     Signal g(t)  is the infinite aperiodic signal ZUSA
```




```
%Signal parameters


% AM = 3
% Effective duration τe, sec 25
%Recognition df in frequency domain, Hz df =< 0,025
% Coefficient Kal "aliasing"                0,00023

% Calculate the parameters and fill the gaps
%1. Spectrum rank                            r  = [ 3 ]
%2. Cutting frequency, Hz                   fc = [0.0021]
%3. Nyquist frequency of the Discrete Fourier
% spectrum                                  fg = [ 0.035]
%4.Space sampling (discrete time quant), sec
%                                          dt = [ 14,28 ]
%5.Preliminary sampling period T% , sec  T%  = [ 40 ]
%  Updated sampling period T, sec          T  = [ 150 ]
%6. Preliminary number of samples       N% = [ 10.5 ]
% Updated number of samples                N   = [ 16 ]
% 8. Discrete frequency quanta, Hz Δf_ = [0.0067]
```

```
% 9. Period of Discrete Spectrum, Hz F = [0.1067]
```

Calculation the parameters of DFT (Example)

1. Spectrum rank r definition (Table 1.1, column 1).

The signal is a "smooth" function, there is no discontinuity of the signal or discontinuity of the derivative t. Therefore, spectrum rank $\boldsymbol{r = 3}$.

2. Calculation of the cutting frequency fc (Table 1.1, column 6, row3):

$$f_c = \frac{1}{6 \cdot \pi \cdot \tau_e} = \frac{1}{6 \cdot \pi \cdot 25} = 0{,}001819 \text{ Hz}$$

3. of Nyquist frequency of the Discrete Fourier spectrum fg .

From the row 3 of the Table 1.1:

B = -57,62 C = 18,69

The solution of the equation (1.15):

$$20 \cdot \lg(K_{al}) = B \cdot \lg\left(\frac{2 \cdot f_g}{f_c}\right) + C$$

$$\lg(2 \cdot f_g) = \frac{20 \cdot \lg(0{,}00023) - 18{,}69}{-57{,}62} + \lg(0.001819) = -1{,}153$$

$$f_g = \frac{10^{-1{,}153}}{2} = 0{,}035 \, \text{Hz}$$

4. Definition of a space sampling (time quant) of the signal, formula (1.12):

$$\Delta t = \frac{1}{2 \cdot f_g} = \frac{1}{2 \cdot 0{,}035} = 14{,}28 \text{ sec}$$

5. Calculation of the preliminary sampling period $T\%$ of limiting, formula (1.19):

$$T\% = \frac{1}{\Delta f} = \frac{1}{0{,}025} = 40 \, \text{sec}$$

6. Calculation of the updated sampling period T, conditions, ((1.20, (1.21).

T>T%

$T = 6*\tau_e$

T = 6*25 = 150 sec

7. Calculation of the preliminary number $N\%$ of the signal and spectrum samples, formula (1.30).

$$N\% = \frac{T}{\Delta t} = \frac{150}{14{,}28} + 1 = 11{,}5 \approx 12$$

8. Update the number N of the signal and spectrum samples, conditions, (1.31), (1.32), (1.34):

$N > N\%$ and $N > 8$

$N = 2^M$

$N = 16$

9. Calculation of parameter *LB* of the signal array, formula (1.34):

$$LB = \frac{\tau_e}{\Delta t} = \frac{25}{14{,}28} = 1{,}75 \approx 2$$

10. Calculation of discrete frequency quant Δf of spectrum, formula (1.4).

$$df = \frac{1}{T} = \frac{1}{150} = 6{,}67 \cdot 10^{-3} \text{ Hz}$$

11. Calculation of the period of the Discrete Spectrum *F* formula (1.7)

$$F = (N\text{-}1)\cdot df = 15\cdot 6.67\cdot 10^{-3} = 0{,}1 \text{ Hz}$$

Notes: *F* cannot be less than $2\cdot f_g$.

Example of program code; Matlab technology [9]

```
%1. Array of the sampling signal.
%Sampling signal array in spatial domain
%Program-Function  ZUZA  is  used  (look  at  chapter
"Appendix")
    LB = 2
    N = 16
    AM = 3.
    A = ZUZA(N,AM,LB)
%2. Graphic of the sampling signal A (Figure 1.20).
    stem(A)
    pause
```

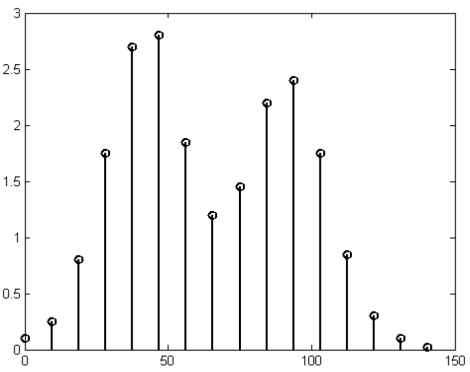


Figure 1.20 – Signal sampling array

```
%3. Creating the array of the Tukey "window".
       W = TUKEY(16)
%4. Avoiding Leakage phenomenon:
       B = A.*W
```

Notes: For the impulse signal the **ZEROF** function is preferable.
For a periodic signal the **SIGM** should be implemented

%5. Graph of the signal after "window" approarch.

```
stem(B)
pause
```

%6. Discrete Fourier Transform:

```
X = fft(B)
```

%7. Amplitude Spectrum in the Frequency Domain:

```
D = abs(X)
```

%8. Phase Spectrum in the Frequency Domain

```
E = angle(X)
```

%7. Graphs of the Spectrum in Frequency Domain and Phase Spectrum in the Frequency Domain (Figure 1.21):

```
stem(D)
pause
stem(E)
pause
```

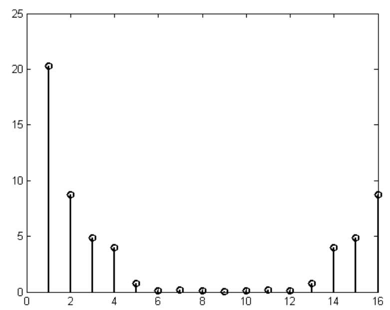


Figure 1.21 – Amplitude spectrum

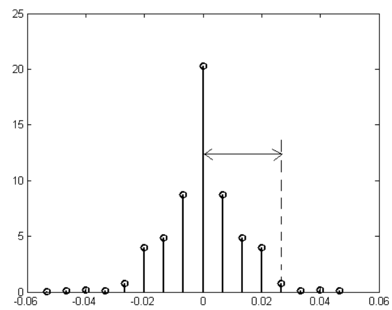


Figure 1.22 – Mirror corrected Amplitude spectrum

%8. Mirror correction of Amplitude and Phase Spectrums

```
FA = fftshift(D)
EA = fftshift(E)
```

%9. Graphs of the Amplitude Spectrum and of the Phase Spectrum after Mirror correction (Figure 1.22, 1.23):

```
stem(F,FA)
pause
stem(F,EA)
pause
```

%10. Energy Spectrum (Figure 1.24)

```
EN = ENG(FA,16)
stem(F,EN)
```

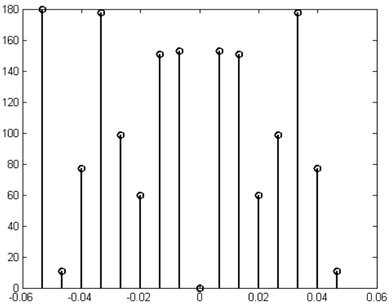


Figure 1.23 – Mirror corrected Phase spectrum

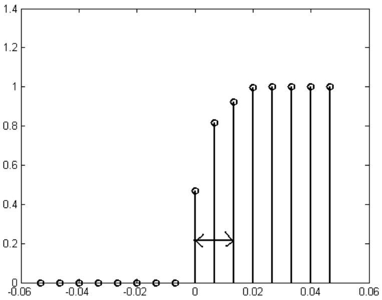


Figure 1.24 – Energy spectrum

Test questions

1. What is the difference between standard (analog) and discrete Fourier transform?

2. Explain what is the Nyquist frequency of the sampled spectrum?

3. Explain what is aliasing phenomena?

4. What approach can be used to eliminate the influence of leakage?

5. What is the main rule of the Fast Fourier transform algorithm?

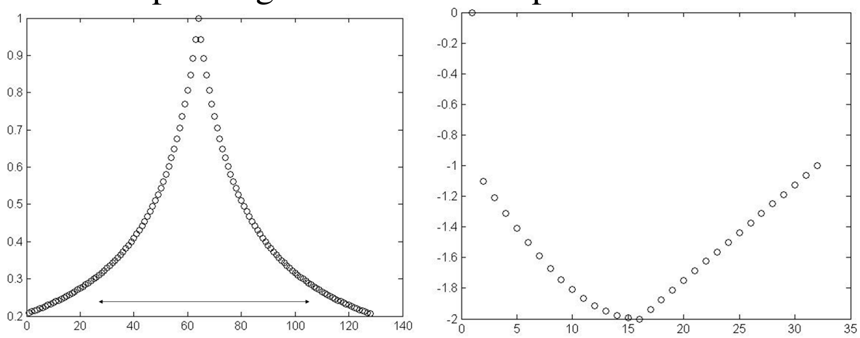
6. Why does the periodic signal sampling period have to consist of an integer number of periods?

7. For which kind of signals should the zero padding be implemented?

9. What is the reason for using the “window” approach?

10. Explain why it is necessary to calculate sampling period in two stage?

11. Tick the corresponding boxes for DFT operation for these two signals



Spectrum rank						Actions to avoid Phenomena DFT					
1		2		3		«window»		Zero padding		Nothing	
1		2		3		«window»		Zero padding		Nothing	

2 Discrete Convolution in the Spatial Domain as a procedure of the Signal Transform by Linear Space-Invariant System

Digital Discrete Convolution is an approach to computer design, research, and simulation of Linear Space-Invariant (LSI) systems. Besides, discrete convolution is a digital processing procedure that is based on the transformation of a signal by a linear-space invariant system. For example, a discrete digital convolution is the main procedure of digital filtering by non-recursive filters [1,2].

2.1 Discrete Convolution as a result of standard transform revising

Transformation of an input signal $g(t)$ into an output signal $y(t)$ by LSI system is defined by standard convolution:

$$y(t) = \int_{-\infty}^{\infty} g(\tau) \cdot h(t-\tau) \cdot d\tau \ , \qquad (2.1)$$

where $g(t)$ is an input signal in the Spatial (time) domain; $y(t)$ is the output signal in the Spatial domain, $h(t)$ is the impulse response function of the LSI, it describes the LSI properties (Figures 2.1).

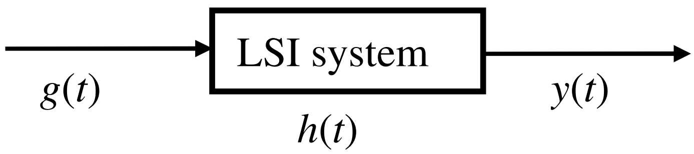


Figure 2.1 – Transformation the input signal by LSI

The impulse response function of LSI is output signal if input equals to Dirac delta function, or $h(t) = y(t)$ if $g(t) = \delta(t)$ [10].

In digital processing a convolution procedure transforms a sampled input signal $g(t)$ into a sampled output signal $y(t)$.

In the Spatial domain a Discrete Convolution is the sampled revising of the integral convolution (2.1):

$$y(m) = \sum_{i=1}^{N_2} g(i) \cdot h(m-i) \ , \qquad (2.2)$$

where $g(i)$ is the sampled signal $g(t)$ by space Δt (discrete time quant Δt) in the Spatial domain; $h(i)$ - is the sampled impulse function $h(t)$, $y(m)$ is the sampled output signal, m is number of sample in the array of the output signal (Figure 2.2)

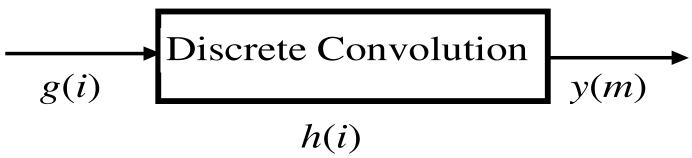


Figure 2.2 – Transformation the sampled input signal by discrete convolution

2.2 Sampling the Input Signal and the Impulse Response Function by the Time Quant

Calculation of the space Δt (time quant) of sampling in a discrete convolution procedure concludes two stages.

1. The space Δt_1 of the sampling input signal $g(t)$ and the space Δt_2 of sampling impulse response function $h(t)$ are calculated according to th eDiscrete Fourier Transform rules (Chapter 1.2).

2. The space Δt of the convolution procedure (2.1),(2.2) is chosen according to the following rule:

$$\Delta t = min(\Delta t_1, \Delta t_2), \tag{2.3}$$

Ignoring this condition would result in the *time scale mismatch* phenomena. Violation of the "duration of the input signal/duration of the impulse response " ratio would result from this phenomenon.

2.3 Calculation of the Sampling Period of the Impulse Response Function. Number of Samples. Fundamental property of Impulse Response Function

The impulse response function $h(t)$ as an operand of a discrete convolution can be defined by only two kinds of functions:

- approximation of a single impulse (Figures 1.15, 1.16, for example);
- approximation ofan infinite aperiodic impulse (Figures 1.8,1.9,1.10,1.11, for example).

Therefore, the sampling period T_2 for impulse response function $h(t)$ can be calculated according to rule (1.27) if $h(t)$ is a single impulse and (1.20),(1.21) or (1.20),(1.22) if $h(t)$ is an aperiodic infinite impulse, respectively.

Number N_2 of samples in an array $h(i)$ of an impulse response function (Figure 2.2) is calculated by the formula:

$$N_2 = \frac{T_2}{\Delta t} + 1, \tag{2.4}$$

where Δt is the space (time quant) of a convolution procedure (see the formula (2.3)).

The impulse response function $h(t)$ of LSI is the output signal if the input equals Dirac delta function $\delta(t)$.Therefore, according to th efundamental property of $\delta(t)$ [10]:

$$\int_{-\infty}^{\infty} h(t) \cdot dt = \int_{-\infty}^{\infty} \delta(t) \cdot dt = 1 , \tag{2.5}$$

The discrete impulse response function has the same properties:

$$\sum_{i=1}^{N_2} h(i) = 1 , \tag{2.6}$$

where $h(i)$, $i = 1,\ldots,N_2$ are the samples of the impulse response function.

2.4 Calculation of the Sampling Period of the Impulse Input Signal. Number of Samples

The sampling period T_1 for the impulse input signal $g(t)$ is calculated according to formula (1.25).

The number N_1 of samples in the array $g(i)$ of the input impulse signal (Figure 2.2) is calculated by the formula:

$$N_1 = \frac{T_1}{\Delta t} + 1, \qquad (2.7)$$

where Δt is the space (time quant) of a convolution procedure (see at the formula(2.3)).

Implementation of the algorithm convolution (2.2) in the space domain involves 3 main stages, which are executed in Matlab automatically [10].

1. Array $g(i)$, $i = 1,\ldots,N_1$ of the sampled input signal $g(t)$ is transformed into array $g_0(k)$, $k = 1,\ldots,(N_1+2\cdot N_2 - 2)$, where both N_2-1 starting samples, and N_2-1 final samples have the zero value (Figure 2.3a, top part). N_2 is the number of samples in the array $h(l)$, $l = 1,\ldots,N_2$ of response function.

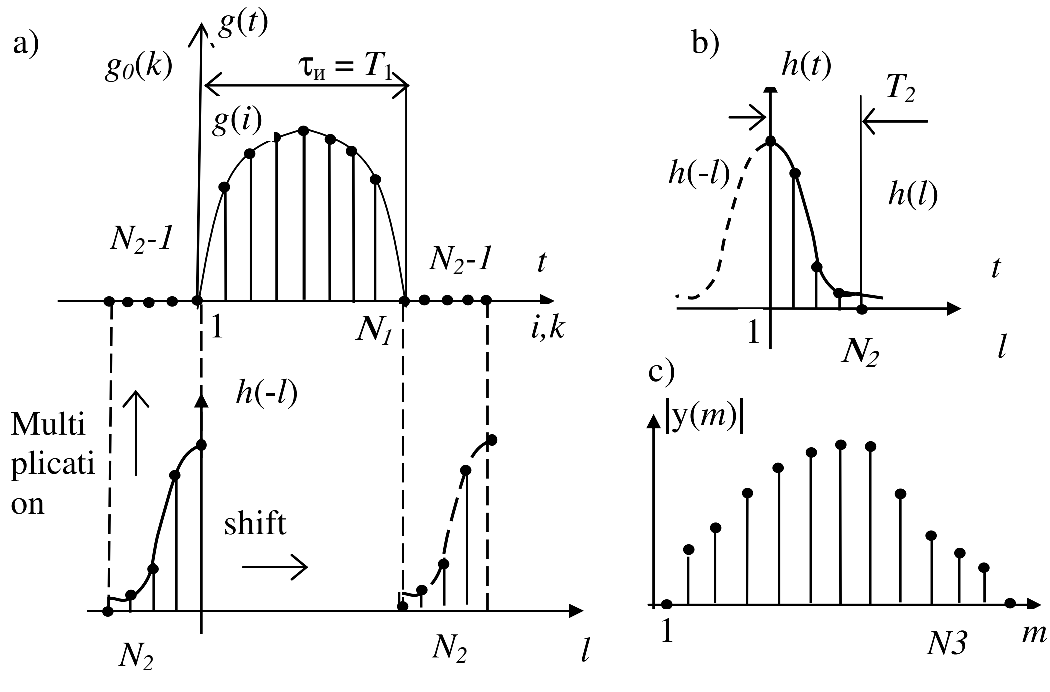


Figure 2.3 – Discrete convolution in the space domain

2. The array of the impulse response function $h(l)$ is transform into the array $h(-l)$, see the solid and the dashed lines in Figure 2.3b, respectively.

3. Convolution according to the algorithm (2.2). In the first step, the samples of the array $h(-1)$, and the starting samples of the array $g_0(k)$ are multiplied (Figure 2.3a, bottom part). In the second step the results of multiplications are summed up. In the third step this sum is transferred to the sample of the output signal array $y(m)$. The number of the sample $m = 1$. In the fourth step the samples of the array $h(-l)$ are

“shifted” by 1 number. These four steps are repeated, and the samples number m = 2,3,4… are calculated sequentially (Figure 2.3c). The computational process will be terminated when the array *h*(*-1*) moves beyond the right finishing sample of the array *g*(*i*).

As result of discrete convolution (2.2) the array *y*(*m*), *m*=1,…,N_3 of the output signal includes N_3 samples:

$$N_3 = N_1 + N_2 - 1, \qquad (2.8)$$

where N_1, N_2 are the number of samples in the arrays *g*(i) of the input signal and *h*(*l*) of the impulse response function

2.5 Calculation of the Sampling Period of the Periodic Input Signal. Number of Samples

If the input signal is a periodic signal, the convolution algorithm implemented in Matlab technology creates error samples in the output signal array.

Let sampling period *T* be equal to one period T_0 of the input signal.

The number of sampling in the array *g*(*i*) of one sampled period of the input signal is *NI* (*i* = 1,…,*NI*).

According to the Matlab algorithm, the array *g*(*i*), *i* = 1,…,*NI* of the sampled period of the input signal is transformed into the array $g_0(k)$, *k*= 1,…,($NI+2\cdot N_2$ - 2), where N_2-1 starting samples and N_2-1 final samples have zero value (Figure 2.3a, top part). N_2 is the number of samples in the array *h*(*l*), *l* = 1,…,N_2 of the response function.

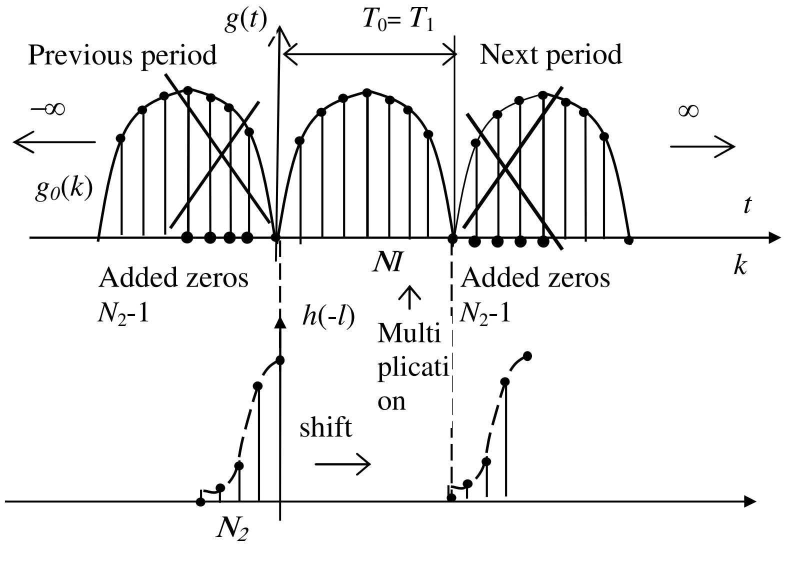


Figure 2.4 – Discrete convolution of periodic input signal

However, the previous period and the next period of the input signals have been set on the positions of the added zero samples. These periods will be damaged. As a result, the samples in the output signal array will be incorrect.

The sampling period T_1 of the periodic input signal has to meet 3 conditions to avoid this error:

$$T_1 = p \cdot T_0, \tag{2.9}$$

$$p \geq 3, \tag{2.10}$$

$$T_1 = 2 \cdot T_2 + T_0, \tag{2.11}$$

where p is integer, T_2 is the sampled period of the impulse response function, T_0 is the period of the input signal (Figure 2.5).

The number of samples in the array of the output signal is calculated as:

$$N_3 = p \cdot NI + N_2 - 1, \tag{2.12}$$

where N_3 is the number of samples in the array of output signals, NI is the number of sampling in one period of input signal, N_2 is the number of samples in the array of the impulse response function:

$$NI = \frac{T_0}{\Delta t} + 1. \tag{2.13}$$

If the sampling period T_1 of the input signal has been chosen according to rules (2.9), (2.10), (2.11), the array of the output signal includes correct samples – see Figure 2.5, where p =3. Only N_2-1 starting samples and N_2-1 final samples in the array of the output signal Y will be incorrect (Figure 2.6).

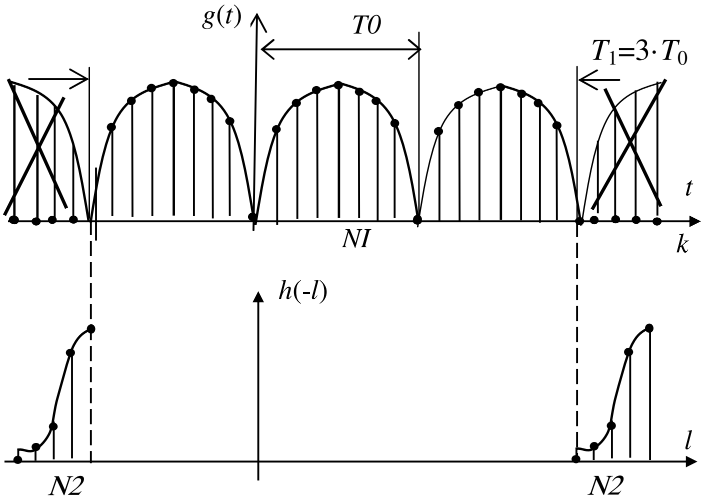


Figure 2.5 – Choosing the sampling period of the periodic input signal

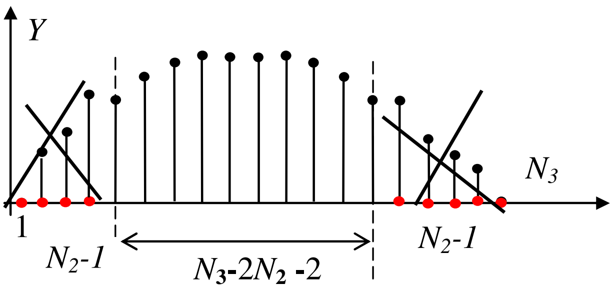


Figure 2.6 – Output signal for the periodic input signal

The correct samples of the output signal array have to be cut into a separate array $y(m)$, $m = 1,\ldots,N_4$ where N_4 is the number of correct samples in the output signal array (Figure 2.7).

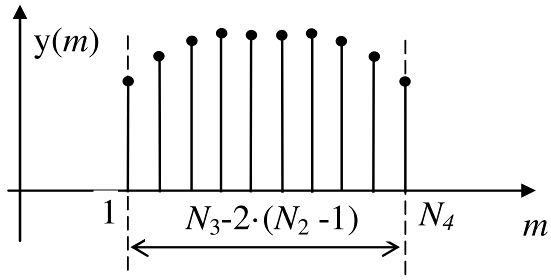


Figure 2.7 – True samples of the output signal

The number N_4 of correct samples in the output signal array is calculated by the formula:

$$N_4 = N_3 - 2\cdot(N_2 - 1) = p\cdot NI - N_2 + 1, \qquad (2.14)$$

where p is the number of periods of the input signal, NI is the number of samples in one period, N_2 is the number of samples in the array of the impulse response function..

2.6 Calculation of the Sampling Period of th eAperiodic Infinite Input Signal. Number of Samples

If the input signal is an aperiodic infinite signal, the convolution algorithm implemented in the Matlab technology creates incorrect samples in the output signal array. The reason for these errors is similar to that of a periodic signal (see paragraph 2.5).

The sampling period T_1 of the input signal is selected according to the following condition:

$$\tau_e \ge T_2, \qquad (2.15)$$

where τ_e is the effective duration of the input infinite signal, T_2 is the sampling period of the inpulse response function of the LSI system.

If τ_e of the input signal meets the condition (2.15), the sampling period T_1 of input signal is calculated by the formula:

$$T_1 = 8\cdot\tau_e, \qquad (2.16)$$

The number N_3 of samples in the array of the output signal is calculated as:

$$N_3 = 8\cdot LB + N_2 - 1, \qquad (2.17)$$

where N_2 is the number of samples in the array of output signals, LB is the number of samples that is equal to the effective duration of the infinite signal (see formula (1.34)).

If the sampling period T_1 of the input signal has been chosen according to rules (2.16) the array of the output signal includes correct samples – see Figure 2.5. Only N_2-1 starting samples and N_2-1 final samples in the array of output signal Y will be incorrect (Figure 2.8).

The correct samples of the output signal array have to be cut into a separate array $y(m)$, $m = 1,\ldots,N_4$ where N_4 is the number of correct samples in the output signal array (Figure 2.8):

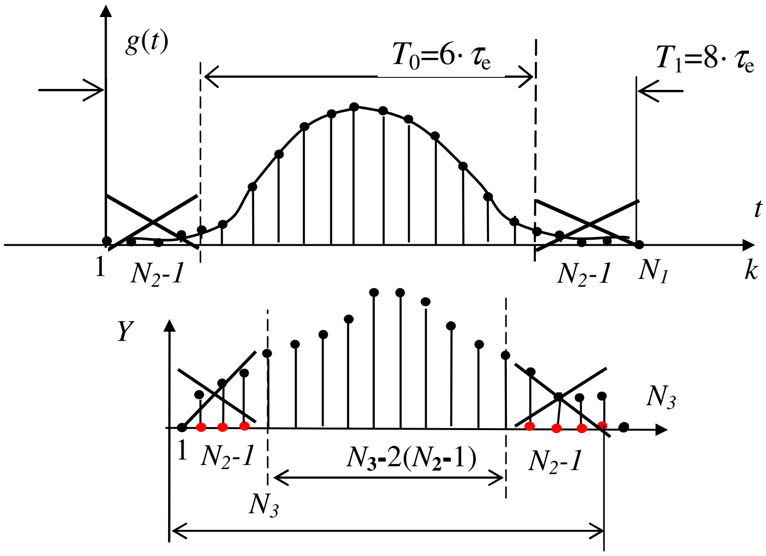


Figure 2.8 – Output signal for an aperiodic infinite input signal

$$N_4 = N_3 - 2\cdot(N_2 - 1) = 8\cdot LB - N_2 + 1, \tag{2.18}$$

If τ_e of the input signal does not meet the condition (2.15) and $\tau_e < T_2$, a symmetric algorithm of discrete convolution has to be used::

$$y(m) = \sum_{i=1}^{N_2} h(i)\cdot g(m-i), \tag{2.19}$$

Comparison of algorithms (2.2) and (2.19) results in the following rule: the input signal parameters are to be calculated according to the formulas for calculating the impulse response function, and the impulse response function parameters are to be calculated according to the formulas of the input signal.

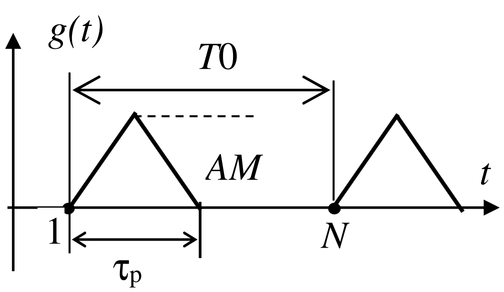
2.7 Example of exercise

Example of task

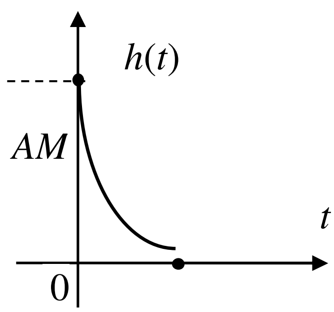
```
%Digital signal processing in optoelectronics
%Laboratory Task 2
%Fundamental  properties  and  the  main  equation  for  %
calculating the parameters of Discrete Convolution
%
%                     General Instructions
%
%For this laboratory exercise you have to calculate
```

```
% the parameters of Discrete convolution.
%
%   The   transformation   version   is   SPATIAL-DOMAIN
%CONVOLUTION
%
%Then generate the program code (in MatLab technology) and
execute this procedure.
%You will get the following graphics.
%1. Sampling the Impulse Responce Function of the
% Linear Space-Invariant System in the Spatial Domain
%2. Sampling the Impulse Responce Function after
% some transformation (optional) in the Spatial Domain
%3. Sampling the Input Signal in the Spatial Domain
%4. Sampling the Input Signal in the Spatial Domain after
%additional  Transformation  (optional)  in  the  Spatial
%Domain
%5. Sampling the Output Signal in the Spatial Domain
%6. Sampling the Output Signal in the Spatial Domain after
% the required Transformation (optional) in the Spatial
Domain
```

Input signal is the periodic triangle impulses



Impulse response function is the exponent




```
%       Input signal

%   Parameters:  AM = 2, E = 0

%  Impulse duration τp, sec               0.05
%  Period T0, sec                        0.1
% Coefficient aliasing Kal                2.5e-4
%
%
```

```
% Impulse response function

% Parameters AM = ?

%Effective duration τε, sec.                  0.029
%
%Coefficient aliasing Kal                     0.01
%
%   Exercises
%   Fill the following gaps
%1. Spectrum rank for the Input Signal         n01 = [2]
%2. Cutting frequency for the Input Signal, Hz
%                                          fc1 = [3.183]
%3. Nyquist frequency of the Discrete Fourier spectrum for
the Input Signal                   fg1 = [247.03]
%4. Space sampling for the Input Signal (discrete time
quant), sec                            dt1 = [2,0241e-3]
%5. Spectrum rank for the Impulse Response Function
%                                             n02 = [1]
%6.Cutting frequency for the Impulse Response Function, Hz
%                                            fc2 = [11.8]
%7. Nyquist frequency of the Discrete Fourier spectrum for
the impulse Response Function, Hz
%                                        fg2 = [1,727E3]
%8. Space sampling for the Impulse Response Function
% (discrete time quanta), sec        dt2 = [2,896e-4]
%9. Chosen optimal Space sampling for the Input Signal %
and the Impulse Response Function, sec    dt = [2,896e-4]
%10. Sampling Period of the Impulse Response Function, sec
                                        T2 = [0,087
%11. The number of the samples in the array of the Impulse
Response Function                            N2 = [301]
%12. Preliminary Sampling Period of the Input Signal,
 sec                                       T1% = [0,174]
%13. Updated Sampling Period of the Input Signal,
% sec                                          T1 = [0.3]
%14. Number of samples in the interval T1 of the Input
Signal                                   N1 = [1035]
%15. Number of samples for one period T0 of the Input
signal                                   NI = [345]
%16. Number of samples for one impulse by period
%                                            LI = [173]
```

```
%17. Number of samples for the Output Signal
%                                        N3 = [1336]
%18. Number of the correct samples for Output Signal
%                                        N4 =   [736]
```

Calculation of the parameters of Convolution in the Spatial Domain (Example)

1. Calculation the optimal space sampling dt (discrete time quant dt)

According to DFT approach the discrete space sampling (time quant is calculated (chapter 1.2).

1.1 Discrete time quant for an impulse by one period.

Impulse duration τ_p=0.05 sec, space sampling is $\Delta t_1 = 2{,}0241\cdot 10^{-3}$sec.

1.2 Discrete time quant for the impulse response function.

Effective duration τ_e=0,0135 sec, space sampling is $\Delta t_2 = 2{,}896\cdot 10^{-4}$sec.

1.3 Optimal space sampling *dt* of convolution in accordance with the formula (2.3):

$\Delta t = \min\{\Delta t_1, \Delta t_2\} = 2{,}896\cdot 10^{-4}$sec

2. . Calculation of the Sampling Period T_2 and the number of samplings N_2 for the one-side infinite aperiodic impulse response function according to the formulae ((1.20),(1.21) and (2.4), (1.34):

$T_2 = 3\cdot\tau_e = 3\cdot 0{,}029 = 0{,}087$sec;

$N_2 = 0.087/2{,}896\cdot 10^{-4} + 1 = 301$

$LB = \tau_e/\Delta t = 0{,}029/\ 2{,}896\cdot 10^{-4} = 100$

3. Calculation the Sampling Period T_1 of periodic input signal $g(t)$ according to formulae (2.9),(2.10),(2.11).

$T_1\% = 2\cdot T_2 + T_0 = 2\cdot 0.087+0{,}1 = 0{,}174$sec

The choice is $p = 3$

$T_1 = 3\cdot T_0 = 3\cdot 0.1 = 0{,}3$ sec

4. Calculation of the number N_1 of samples of the periodic input signal $g(t)$ according to formula (1.30):

$N_1 = T_1/\Delta t +1 = 0{,}3/2{,}896\cdot 10^{-4} +1 = 1036$

5. Calculation of the number *NI* of samples by 1 period and the number *LI* of the impulse samples by the period according to formulae (1.36),(1.37),(1.38).

$NI = T_0/\Delta t +1 = 0.1/2{,}896\cdot 10^{-4} +1 = 346$

$SKV = T_0/\tau_p = 0{,}1/0{,}05 = 2$

$LI = NI/SKV = 346/2 = 173$

6. Calculation of the number N_3 of the output signal samples according to the formulae (2.8),(2.12).
$N_3 = N_1+N_2-1=1036+301-1 = 1336$

7. Calculation of the number N_4 of correct samples in the output signal (formula (2.14)).
$N_4 = N_3-2\cdot(N_2-1) = 1336-2\cdot(301-1) = 736$

Example of Matlab technology program code

1. Creating the array of the Impulse Response Function. Function EP.

Firstly, the amplitude AM of impulse response function is found. This calculation is made several times

```
AM =1.0; LB = 100
H = EP(300,AM,LB)
sum(H)
```

The value AM is changed every time. The objective of this action is to find the required value of the amplitude AM meeting the condition (2.6).
The resulting value AM = 0,0105 has been found.

Secondly, in the beginning of program code,the array of the Impulse Response Function is generated (Figure 2.9). Function EP is used (see “Appendix”).

```
AM =0.0105; LB = 100
H = EP(301,AM,LB)
plot(H)
pause
```

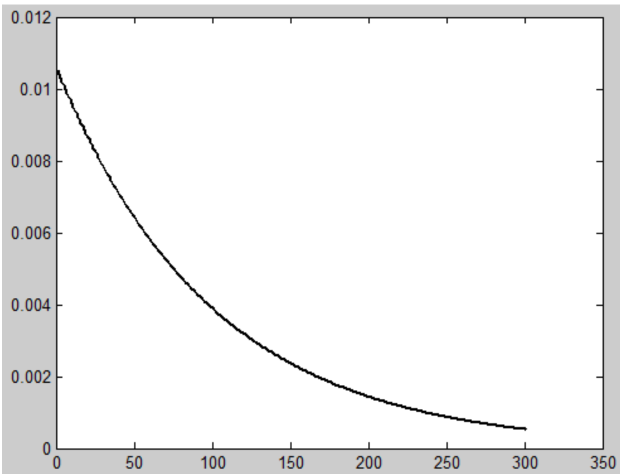


Figure 2.9 – Impulse response function array

2. Creating the array of Impulse by one Period of the Input Signal.

```
A = SIG(173,72,72,2.0,0.0)
```

```
plot(A)
pause
```

3. Creating the Array of the 3 periods of the Input Signal (Figure 2.10).
Function SIGM.

```
NI = 173, N1 = 1036, SKV = 2, M =3
G = SIGM(A,173,1036,2.0,3)
plot(G)
pause
```

4. Spatial-Domain Convolution. Creation the array YR of the Output Signal (Figure 2.11)

```
YR=conv(G,H)
plot(YR)
```

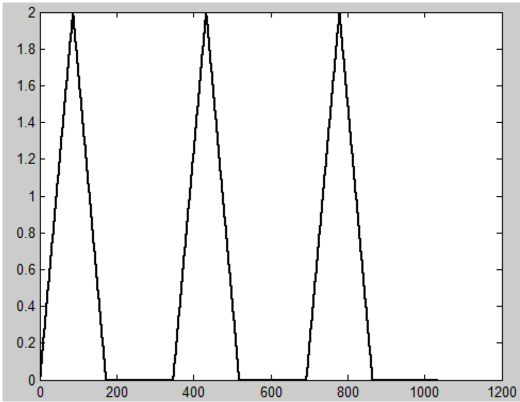


Figure 2.10 – Input signal array

5. The part of the array cut out of the region contains correct samples of the Output Signal. Creating the correct Output Signal (Figure 2.12):

```
Y = SECTION(YR,1036,736,300)
plot(Y)
```

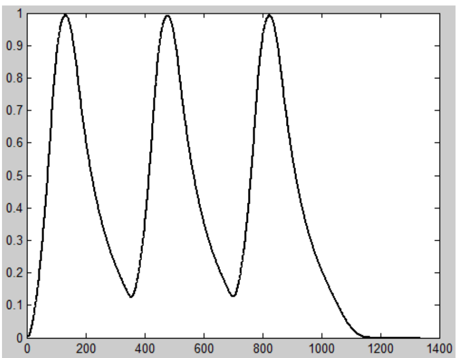


Figure 2.11 – Output signal array

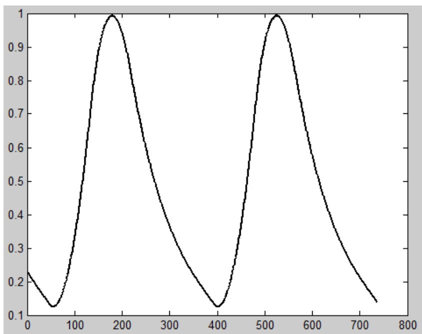


Figure 2.12 – Correct samples of output signal

Test questions

1. What is the difference between standard (analog) and discrete convolution?

2. Explain how to calculate the space Δt (time quant) in convolution algorithm?

3. Explain the reason of the errors in the Matlab convolution algorithm?

4. What approach can be used to find the correct part of the output signal if the input signal is a periodic object?

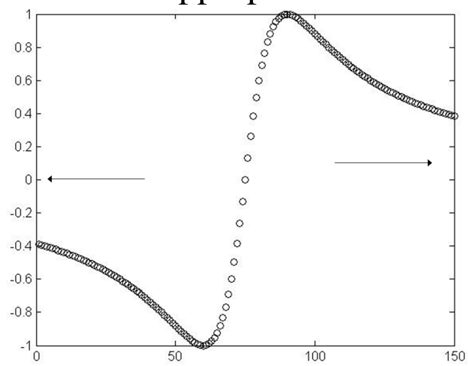
5. What is the main rule of Fast Fourier transform algorithm?

6. What convolution parameters are calculated using the DFT approach?

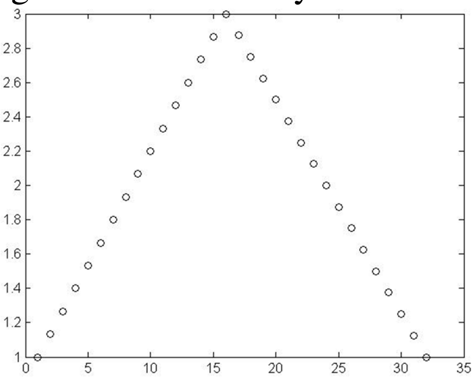
7. What kind of system is simulated by the convolution approach?

8. Explain the fundamental properties of the impulse response function.

9. Tick the appropriate boxes for the signal transformed by LSI.



Input signal



Impulse response function

Action	Check-box
“Window” approach is implemented for the input signal	
The sampling space Δt (time quant) of the input signal is equal to the same space of the impulse response function	
The number of samples in the input signal array is made equal to the number of samples in the impulse response function array	
Avoiding “mirror”, two halves of the output signal array are re arranged	
The array of the input signal includes several periods	
Zero samples are added to the input signal array and to the impulse response function array (avoiding the circular convolution)	
The correct part of the output signal array is cut out	
The number of sampling in the arrays of the input signal and the impulse response function meet the condition “2 power M”, M is integer.	
“Window” approach is implemented for the impulse response function or the input signal	

3 Discrete Convolution in the Frequency Domain as a procedure of the Signal Transform by Linear Space-Invariant System

Digital Discrete Convolution in the Frequency Domain is an alternative approach to computer design, research, and simulation of Linear Space-Invariant (LSI) systems and non-recursive filters [1,2].

The difference between the convolution in the spatial domain and the convolution in the frequency domain is the computational complexity of the procedure. The speed of calculations using convolution in the frequency domain is definitely greater than that of the calculations using convolution in the spatial domain. These are the reasons for preferring convolution in the frequency domain for signal processing in many applications.

On the other hand, signal processing errors when using convolution in the frequency domain are greater than when calculating the output signal by convolution in the spatial domain. Therefore, convolution in the spatial domain is used if the high accuracy of digital processing is required [1,2].

3.1 Convolution in the Frequency Domain Algorithm

Signal processing by using the convolution algorithm in the frequency domain is based on the fundamental property of Fourier transform. In accordance to this property, the procedure of convolution of the two operands $g(t)$ and $h(t)$ in the spatial domain is equivalent to the product of the spectra $S(f)$,$H(f)$ of these operands in the frequency domain:

$$\int_{-\infty}^{\infty} g(\tau)\cdot h(t-\tau)\cdot d\tau \Rightarrow S(f)\cdot H(f) \ , \qquad (3.1)$$

where $S(f)$,is the spectrum of the input signal and $H(f)$ is the spectrum of the impulse response function.

Discrete convolution in the frequency domain is the sampling view of the algorithm (3.1):

$$Y(n) = S(n)\cdot H(n) \ , \qquad (3.2)$$

where $S(n)$ is the array of the sampled spectrum input signal $g(t)$, and $H(n)$ is the array of the sampled spectrum of the impulse response function, $Y(n)$ is the array of the spectrum output signal.

So a calculation of the transformed signal as the output of LSI system includes 4 stages as follows (Figure 3.1):

1. Calculation of the array $S(n)$, $n = 1,\ldots,N$ of the spectrum input signal resulting from DFT.
2. Calculation of the array $H(n)$, $n = 1,\ldots,N$ of the spectrum impulse response function resulting from DFT.
3. Calculation of the array $Y(n)$, $n = 1,\ldots,N$ of the spectrum output signal resulting from multiplication of the arrays $S(n)$ and $H(n)$.

4. Calculation of the array $y(m)$, $m = 1,\ldots,N$ of the output signal resulting from the Inverse Fourier Transform (IDFT).

Convolution algorithm in the frequency domain has two main features:

- the arrays of the input signal $g(i)$, the impulse response function $h(i)$ and the output signal $y(i)$ have the same number N of sampling;
- the number N of samples in the arrays $g(i)$, $h(i)$ and $y(i)$ has to meet the rules (1.30),(1.31),(1.32) of the algorithm of Fast Discrete Fourier Transform (FDFT): $N = 2$ power M, M is integer.

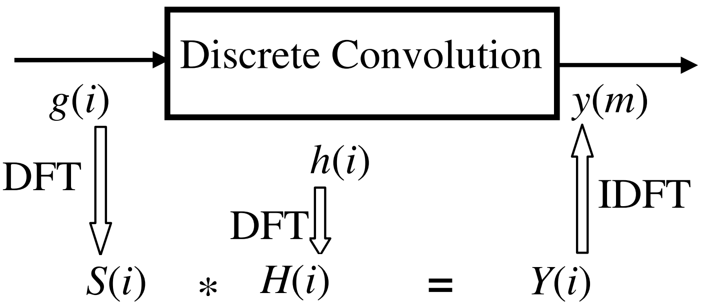


Figure 3.1 – Discrete convolution in the frequency domain

3.2 Convolution Periodic Input Signal in the Frequency Domain

1. The preliminary space $\Delta t\%$ of the convolution procedure is calculated according to the rule (2.3).

2. Sampling period T_2 for the impulse response function $h(t)$ is calculated according to the rules (1.27) if $h(t)$ is a single impulse and (1.20),(1.21) or (1.20),(1.22) if $h(t)$ is an aperiodic infinite impulse, respectively.

3. The sampling period T_1 for the input periodic signal is defined by the conditions:

$$T_1 = p \cdot T_0, \qquad (3.3)$$

$$T_1 \geq T_2, \qquad (3.4)$$

where p is integer, T_2 is the sampled period of the impulse responds function, T_0 is the period of the input signal.

An example for $p = 2$ is shown in Figure 3.2.

4. Calculation the preliminary number $N\%$ of samples in the arrays of the input signal, the impulse response function and the output signal:

$$N\% = \frac{T}{\Delta t} + 1. \qquad (3.5)$$

5. The updated number N of sampling is calculated according to the 3 rules FDFT:

$$N \geq N\%, \qquad (3.6)$$

$$N = 2^M , \tag{3.7}$$
$$N > 2^{M-1} , \tag{3.8}$$

6. Updated space Δt of the convolution procedure is calculated as:

$$\Delta t = \frac{T}{N-1} . \tag{3.9}$$

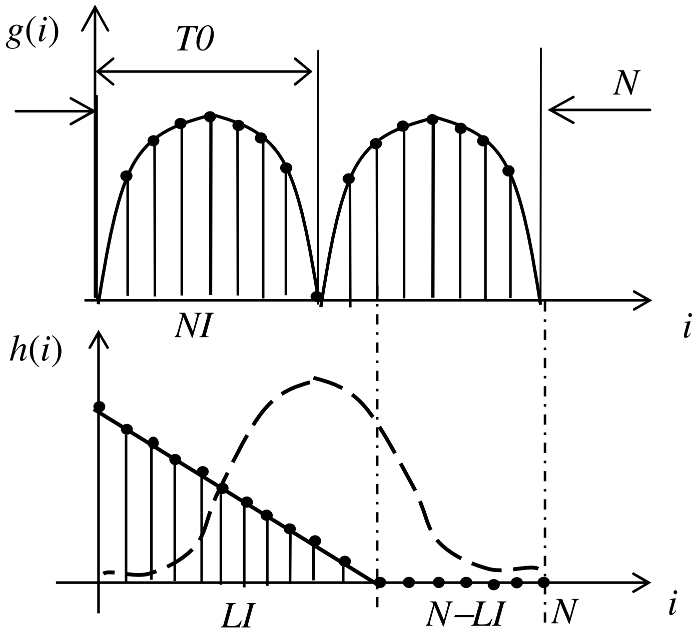


Figure 3.2 – Periodic input signal convolution

7. The parameters of the impulse by one period of the input signal are calculated using formulas (1.36), (1.37), (1.38).

8. The number of samples *LB* corresponding to effective duration τ_e of the aperiodic infinite impulse response function (Figure (3.2), dash line) is defined by the formula:

$$LB = \frac{\tau_e}{\Delta t} , \tag{3.10}$$

The array $h(i)$, $i = 1,...,N$ is completely filled with its samples if the impulse response function is an aperiodic infinite function.

If the impulse response function is a single impulse, only the part *LI* of the array $h(i)$ is filled with its samples:

$$LI = \frac{\tau_p}{\Delta t} , \tag{3.11}$$

where τ_p is the duration of the impulse.

The other part of the array is filled with zero samples (Figure (3.2), solid line).

3.3 Convolution Aperiodic Infinite Input Signal in the Frequency Domain

Points 1 and 2 in the calculations are the same as those to the period signal convolution (chapter 3.2).

3. The sampling period T_1 for the input aperiodic infinite signal is defined by the conditions:

$$T_1 = p \cdot \tau_e, \tag{3.12}$$

$$p \geq 6, \tag{3.13}$$

$$T_1 \geq T_2, \tag{3.14}$$

where p is integer, τ_e is the effective duration of the infinite impulse responds function.

An example for $p = 6$ is shown in Figure 3.3.

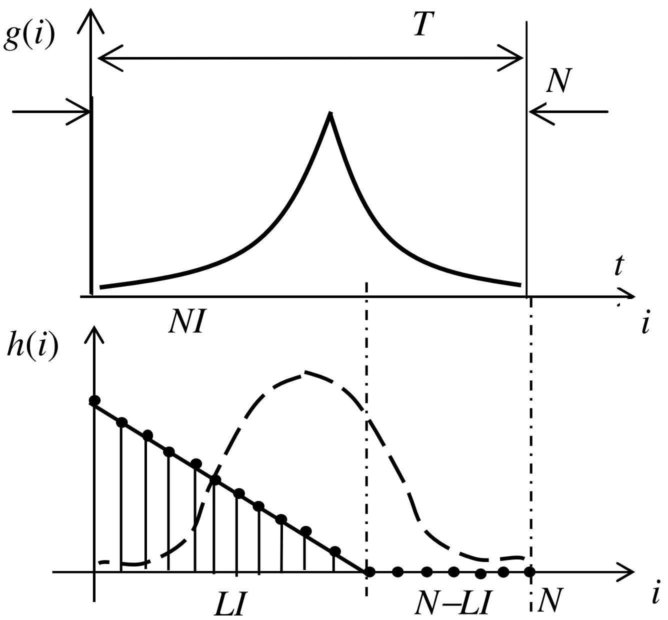


Figure 3.3 – Aperiodic infinite input signal convolution

4. Calculation the preliminary number $N\%$ of samples in the arrays of the input signal, the impulse response function and the output signal:

$$N\% = \frac{T}{\Delta t} + 1. \tag{3.15}$$

5. The updated number N of sampling is calculated according to the 3 rules of FDFT:

$$N \geq N\%, \tag{3.16}$$

$$N = 2^M, \tag{3.17}$$

$$N > 2^{M-1}, \tag{3.18}$$

6. The updated space Δt of the convolution procedure is calculated as:

$$\Delta t = \frac{T}{N-1}. \tag{3.19}$$

7. The number of samples *LB* corresponding to the effective duration τ_e of the aperiodic infinite input signal (Figure 3.3, top part) is defined by the formula:

$$LB = \frac{\tau_e}{\Delta t}, \tag{3.20}$$

where τ_e is the effective duration of the aperiodic infinite input signal.

8. Point 8 corresponding to calculation of the parameters *LB* or *LI* of the impulse response function is the same as that for the period signal convolution (chapter 3.2, Figure 3.3, bottom part).

3.4 Convolution of a Single Impulse Input Signal in the Frequency Domain

1. Space Δt (time quant) of the convolution procedure is calculated according to the rule (2.3).

2. The number of samples *LI* in array $g(t)$ of the impulse input signal is calculated using the formula (1.27) – see Figure 3.4, top part.

3. The sampling period T_2 for the impulse response function $h(t)$ is calculated according to the rules (1.27) if $h(t)$ is a single impulse and to (1.20),(1.21) or (1.20),(1.22) if $h(t)$ is an aperiodic infinite impulse, respectively.

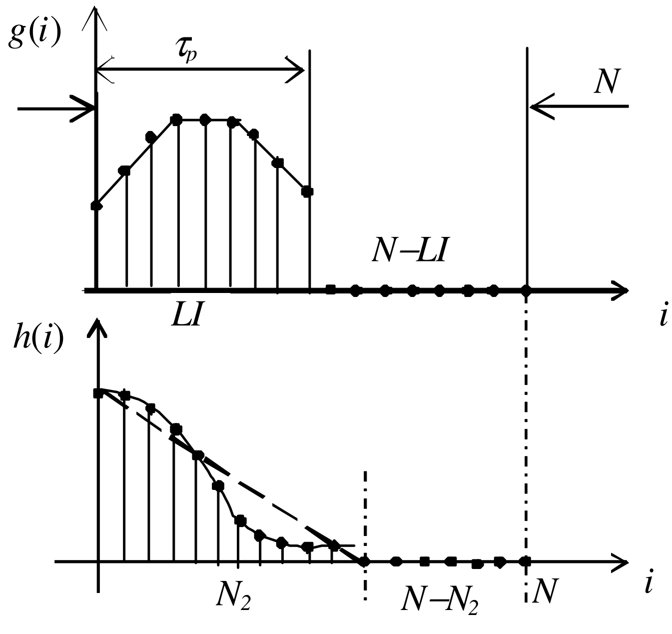


Figure 3.4 – Alone impulse input signal convolution

4. The number of samples N_2 in the array $h(t)$ of the impulse response function is calculated using the formula (2.4) – see Figure 3.4, bottom part, the solid line refers to the infinite impulse response function, the dashed line refers to the limited impulse

response function.

5. The preliminary number N_3 of samples of the output signal is calculated according to rule (2.8):

$$N_3 = LI + N_2 - 1, \quad (3.21)$$

where *LI* is the number of samples in the array $g(t)$ of the impulse input signal.

6. The number N_3 of sampling updated to the number N according to the 3 rules of FDFT:

$$N \geq N_3, \quad (3.22)$$

$$N = 2^M, \quad (3.23)$$

$$N > 2^{M-1}, \quad (3.24)$$

where M is integer.

7. Part *LI* of array $g(i)$ is filled with the samples of the impulse input signal. The other part of the array is filled with *N-LI* zero samples (Figure 3.4, top part).

8. Part N_2 of array $h(i)$ is filled with the samples of the impulse response function as a limited impulse or an infinite impulse. The other part of the array is filled with $N - N_2$ zero samples (Figure 3.4, bottom part).

Ignoring the points 5,6,7,8 would result in *inter-period interference error* or *circular convolution* phenomena [11] and errors in output signal calculations.

3.5 Example of exercise

The example of the task is the same as in paragraph 2.7 regarding Convolution in the Spatial Domain.

1. Calculation of the parameters Δt_1, Δt_2 of Convolution in the Spatial Domain is the same as in paragraph 2.7 regarding Convolution in the Spatial Domain.

$\Delta t\% = \min\{\Delta t_1, \Delta t_2\} = 2{,}896 \cdot 10^{-4}$ sec

2. The sampling period T_2 of the impulse response function is calculated in paragraph 2.7 regarding Convolution in the Spatial Domain.

$T_2 = 3 \cdot \tau_e = 3 \cdot 0{,}029 = 0{,}087$ sec

3. The sampling T_1 for the input periodic signal is defined by the conditions (3.3):

$T_1 = p \cdot T_0$ and $T_1 \geq T_2$

If $T_0 = 0{,}1$ sec, $T_2 = 0{,}041$ sec, p = 1

As result, $T_1 = 0{,}1$ sec

4. The preliminary number *N%* of samples in the arrays of the input signal is calculated by the formula (3.8):

$N\% = T/\Delta t\% + 1 = 0{,}1/2{,}896 \cdot 10^{-4} + 1 = 345$

5. The updated number N of sampling is calculated according to the 3 rules of FDFT (3.6),(3.7),(3.8):

$N=2^9 = 512$

6. The space Δt of the convolution procedure is updated using the formula (3.9):
$\Delta t = T/(N\text{-}1) = 0{,}1/511=1{,}957\cdot 10^{-4}$ sec.

7. Calculation the parameters of impulse by one period of the input signal, formulae (1.36), (1.37), (1.38):
$SKV = T_0/\tau_p=0{,}1/0{,}05 = 2$
$NI = N/\text{p} = \text{N}=512$
$LI=NI/SKV = 512/2=256$

8. The number of samples LB corresponding to the effective duration τ_e of the aperiodic infinite impulse response function is calculated by the formula (3.10):

$LB=\tau_e/\Delta t = 0{,}029/1{,}957\cdot 10^{-4} = 148$

Example of the program code of Matlab technology

1. Firstly, the amplitude AM of impulse response function is found.
This calculation is made several times

```
AM =1.0; LB = 148
H = EP(512,AM,LB)
sum(H)
```

The value AM is changed every time. The objective of this action is to find the required value of the amplitude AM meeting the condition (2.6).
The resulting value AM = 0.007 has been found.

Secondly, in the beginning of the program code the array of the Impulse Response Function is generated (Figure 3.5). Function EP is used (see “Appendix”).

```
AM =0.007; LB = 148
H0 = EP(512,AM,LB)
plot(H0)
pause
```

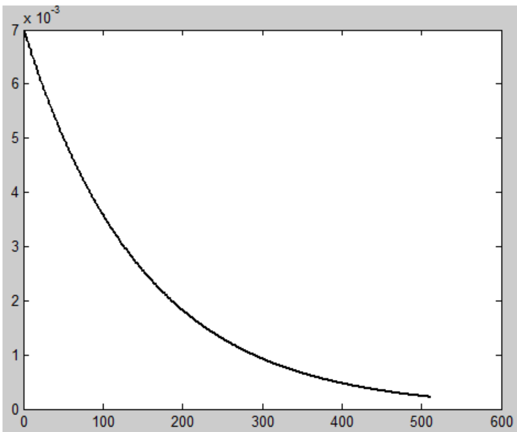


Figure 3.5 – Impulse response function array

2. “Window” approach implemented

```
W=TUKEYS(512)
H =H0.*W
```

3. Creating the array of Impulse by one Period of Input Signal.

```
A = SIG(256,128,128,2.0,0.0)
plot(P1,A)
pause
```

4. Creating the Array of the 1 period of the Input Signal (Figure 3.6).
SIGM function.

```
NI = 256, N1 = 512, SKV = 2, M =1
G = SIGM(A,256,512,2.0,1)
plot(PG)
pause
```

5. Frequency-Domain Convolution. Calculation of the preliminary array YR of the Output Signal (see chapter 3.1)

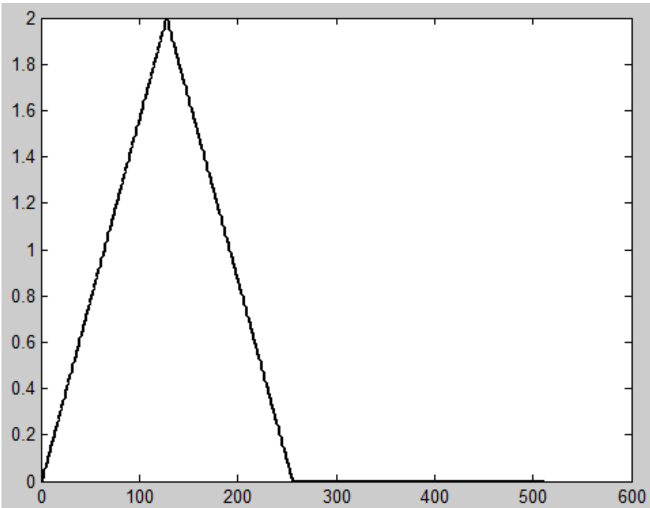


Figure 3.6 –One period input signal array

```
SG = fft(G)
SH = fft(H)
SY = SG.*SH
YR = ifft(SY)
```

6. Reducing the errors of the Convolution transform. Creating an array of the Output signal without errors (Figure

```
Y = real(YR)
plot(Y)
```

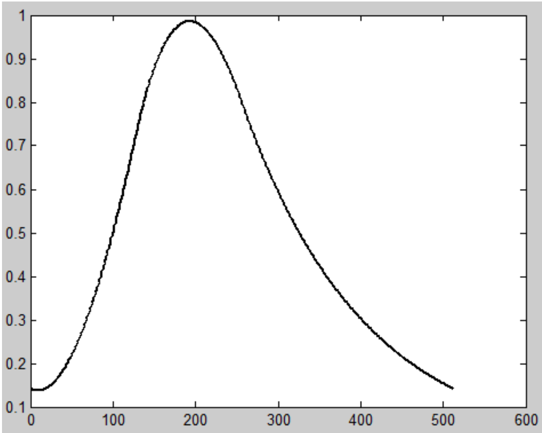
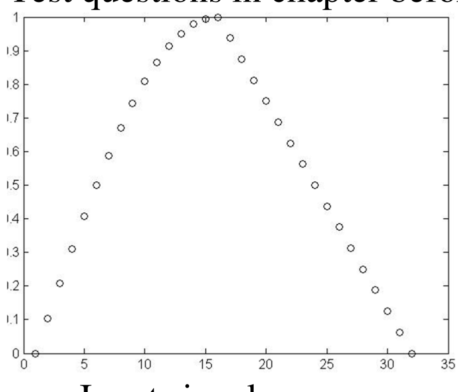


Figure 3.7 –Output signal array

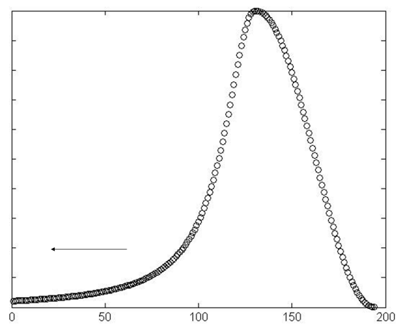
Comparing the two options of the output signal in Figure 3.7 and Figure 2.12 shows that the convolution in the spatial domain creates some redundancy, due to the two periods of output signal formed. The output signal resulting from the convolution in the frequency domain includes only one period.

Test questions

1. What is the difference between convolution in the spatial domain and convolution in frequency domain?
2. Explain the features of the convolution algorithm in the frequency domain.
3. Explain the reason for the errors in Matlab convolution algorithm.
4. Why it is necessary to upgrade the value of the sampling space Δt (time quant)?
5. What kind of system is simulated by the convolution approach?
6. What are the advantages and disadvantages of convolution in the frequency domain?
7. Why is no correction of the mirror phenomenon required in the frequency domain convolution?
8. How to avoid the circular phenomena errors in the output signal array?
9. Tick the appropriate boxes in table by question 9 for the signal transformed by LSI (see Test questions in chapter before).



Input signal



Impulse response function

Conclusion

The methods for performing two fundamental procedures of digital signal processing are considered: harmonic Fourier analysis and synthesis of linear invariant systems. Discrete Fourier transform and discrete convolution are the bases for designing and modeling electronic and optical linear elements.

The features of the transition from analog procedures to discrete algorithms are analyzed. Computational and algorithmic methods are recommended to reduce the errors of discrete procedures. Digital processing using the discrete Fourier transform is found to be a direct change in the spectrum of the processed signal. The signal transformation by a linear system is shown to be a discrete convolution (non-recursive filtering) of a signal with a pulse characteristic of the filter. The described examples of calculating the parameters and program code of discrete procedures in Matlab technology will help in the practical application of the digital processing procedures studied.

References

1 Khan Iftekhauddin, Abdul A. Awwal. Field Guide to Image Processing. SPIE Field Guides, vol. FG25 – SPIE Press, USA, 2012. – 86 p.

2 Antoniou Andreas. Digital signal processing. Signals, Systems and Filters. – London, San Juan, Toromti: McGrall-Hill Companies, 2006 – 937 p

3 Claudio Durastanti. Aliasing effects for random fields over spheres of arbitrary dimension//Electronic Journal of Statistics, imsart-ejs ver. 2014/10/16 file: aliasing_d__ultimate.tex, November 27, 2019. P. 39

4 Isaac Amidror, Roger D. Hersch. Jaggies as aliasing or reconstruction phenomena: a tutorial // Journal of Electronic Imaging 23(1), 010801, Jan.–Mar., 2014. P. 010801-1 - 010801-11.

5 Yan Feng Li, Kui Fu Chen. Eliminating the picket fence effect of the fast Fourier transform// Computer Physics Communications. V. 178, issue 5, 2008. P486-491

6 Bharti Thakur, Rajesh Mehra. Discrete Fourier Transform Analysis with Different Window Techniques Algorithm // International Journal of Computer Applications (0975 – 8887), V. 135,No.12, 2016. P.17-20

7 Fredrik Gustafsson, Lennart Ljung, Mille Millnert. Signal Processing. , Lund, Sweden:Studentlitteratur, 2010, p220.

8 Frigo, Matteo; Johnson, Steven G. (January 2007) [2006-12-19]. A Modified Split-Radix FFT With Fewer Arithmetic Operations// IEEE Transactions on Signal Processing. 55 (1),2007. P.: 111–119

9 Holly Moore MATLAB for Engineers, Global Edition. Pearson (Anspear), 2018. 312 p.

10 Aratyn, Henrik; Rasinariu, Constantin (2006), A short course in mathematical methods with Maple: World Scientific, 2006. 215 p.

11 Jeruchim, Michel C.; Balaban, Philip; Shanmugan, K. Sam. Simulation of Communication Systems: Modeling, Methodology and Techniques. New York: Kluwer Academic Publishers, 176 p.

Appendix

Basic operations and functions

Basic operations and functions

The following functions generate an array of the impulse signal

1. The array of the aperiodic infinite impulse

A function code is defined as follows:

```
A = NAME(L, AM, E),
```

where NAME is the name of the function;
A is the name of the generating array;
L is the number of samplings in array A;
AM is the amplitude or the value of the varied part of the signal;
E is the average level, or the value of the unchangeable part of the signal

1.1. Cosine to power 2 Impulse ($\cos^2(x)$ or cos(x)*cos(x)). Figure A 1.1.
NAME of the function is CS2

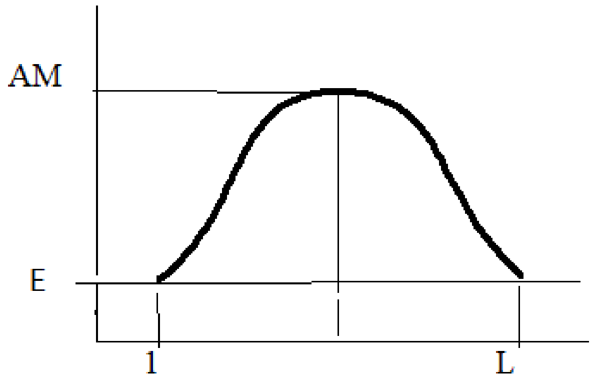


Figure. A1.1 Cosine to power 2 Impulse

1.1. Parabolic impulse. Figure A1.2.
NAME of the function is PARABOL

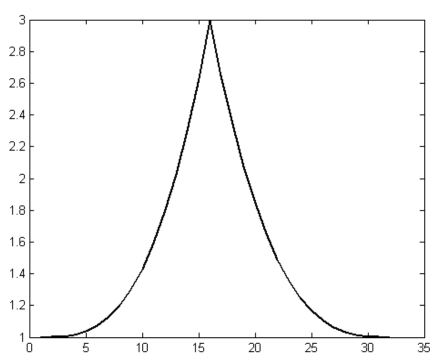


Figure A1.2 Parabolic impulse

1.3 Cosine impulse. Figure A1.3.

NAME of the function is CS

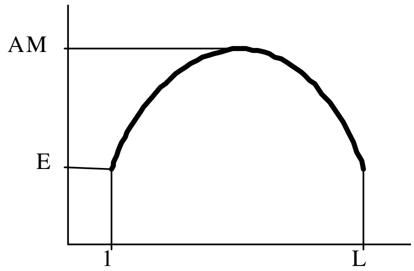


Figure A1.3 Cosine impulse

1.4 Trapezoid and Triangle impulses

NAME of the function is SIG.

The function code is defined as follows:

```
A = SIG(L,L1,L2,AM,E),
```

where L1, L2 are the numbers of the positions of the reference points for trapezoid – see Figure A1.4.

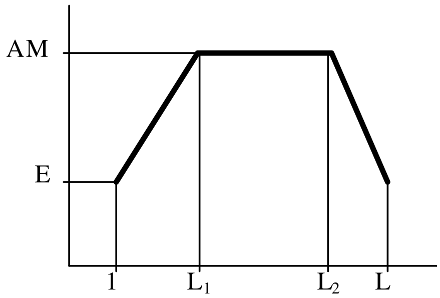


Figure A1.4 Trapezoid and Triangle impulses

The parameter AM determines the "roof" size of the trapezoid, and the parameter E determines the "basement" of the trapezoid.

Figure A1.4. corresponds to positive AM. For a negative AM the trapezoid is shaped as « pressed through the roof ».

A triangle impulse would require L1 = L2 = L/2 - see Figure A1.5

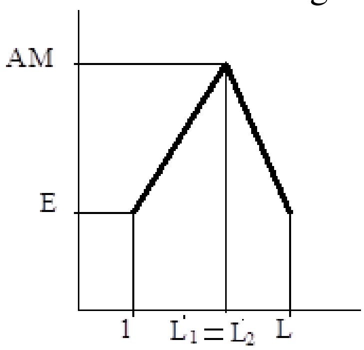


Figure A1.5 Triangle impulse (option 1)

For a triangle impulse L1 = L2 = L – see Figure A1.6

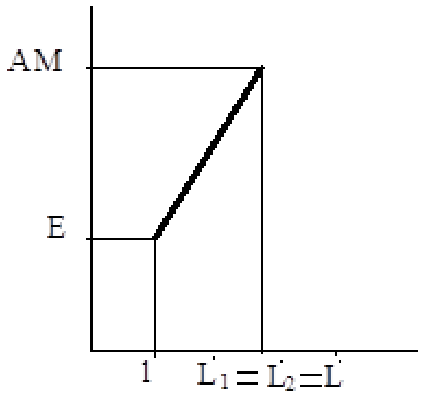


Figure A1.6 Triangle impulse (option 2)

For a triangle impulse L1 = L2 = 1 – see Figure A1.7

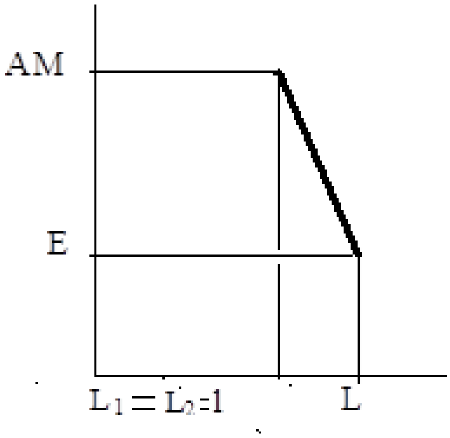


Figure A1.7 Triangle impulse (option 3)

2. Infinite aperiodic signals

The function code is defined as follows:

```
A = NAME(L,AM,LB)
```

where NAME is the name of the function;
A is the name of the generated array;
L is the number of samplings in the signal array;
AM is the value of the amplitude, or the value of the varied part in the signal;
LB is the number referring to the effective duration of the signal;

$$LB = \frac{\tau_e}{\Delta t}$$

here τ_e is the effective duration of the signal;
Δt is the discrete time quanta (the space of the sampling)

2.1 Exponential impulse. Figure A2.1.
NAME of the function is EPN

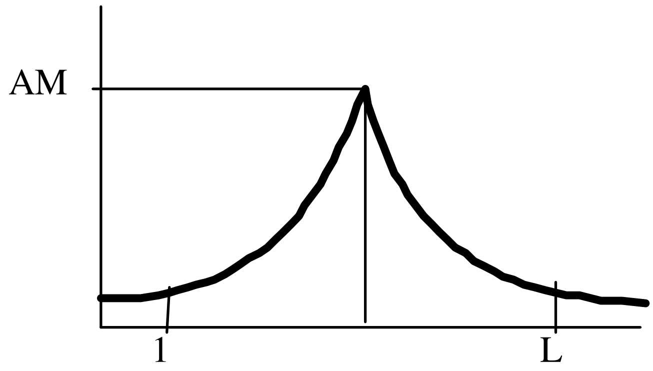


Figure A2.1 Exponential impulse

2.2. Inverse exponential impulse - see Figure A2.2
NAME of the function is EPNI

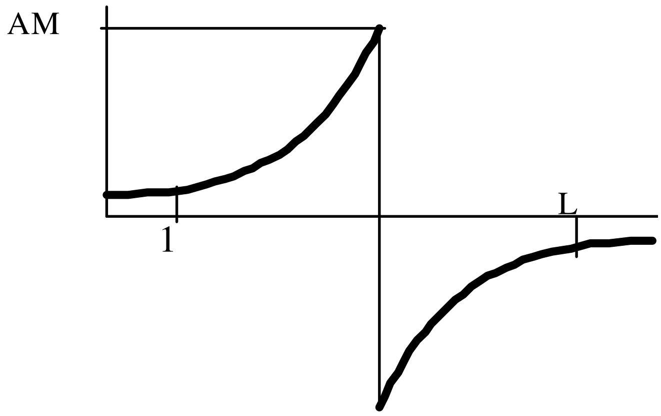


Figure A2.2 Inverse exponential impulse

2.3 Impulse “Ringlet of Mary Agnese”. Figure A2.10
NAME of the function is ANEZI

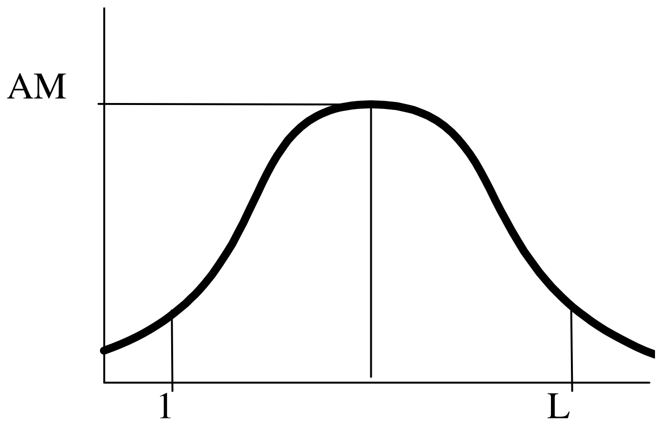


Figure A2.3. Impulse “Ringlet of Mary Agnese”

2.4. “Ringlet of Mary Agnese” + Part of the Inverse exponent. Figure A2.4.
NAME of the function is ANEPM

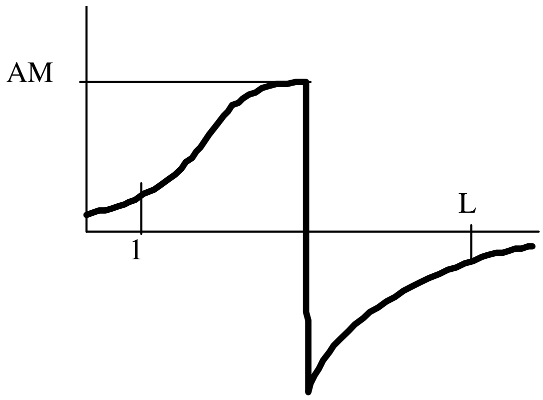


Figure A2.4. Impulse “Ringlet of Mary of Agnese” + Part Inverse exponent

2.5 . Special exponential impulse. Figure A2.5
NAME of the function is EPSC

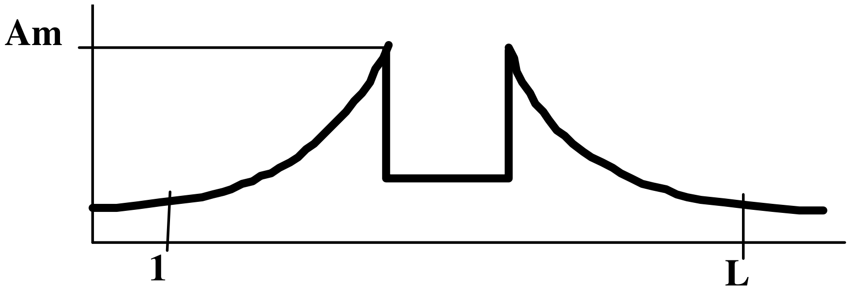


Figure A2.5. Special exponential impulse

2.6. Exponential impulse. Figure A2.6.
NAME of the function is EP

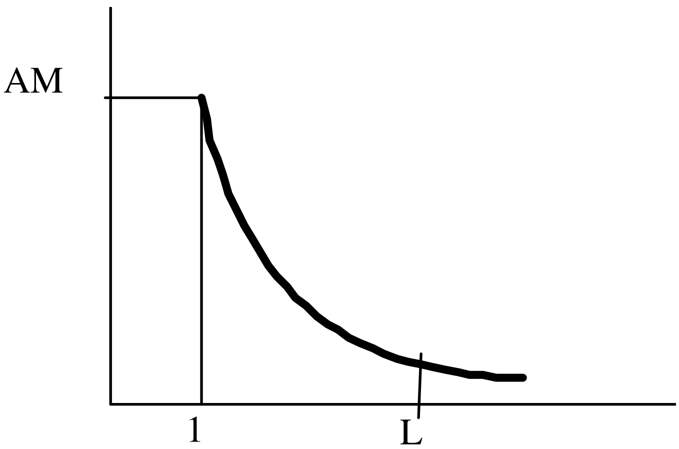


Figure A2.6 Exponential impulse

2.7. (1/2) part of the Gauss impulse - see Figure A2.7
Name of the function is GSS

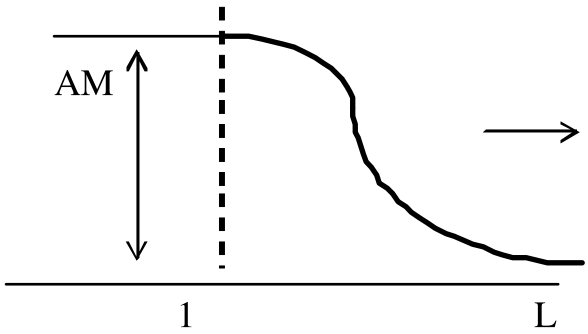


Figure A2.7. (1/2) part impulse of Gauss

2.8. Harmonic signals. - Figure A2.8
Name of the function is GARM.
The function code is defined as follows:

```
A = GARM (L, TM, AM, E, FI)
```

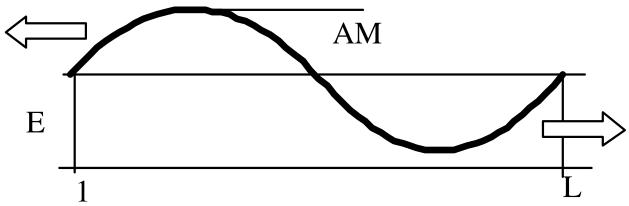


Figure A2.8 Harmonic signal

where TM is the number of periods in an interval L;
FI is the phase angle (in radians). The parameters A, L, AM, E are similar to those considered for other signals.

2.9 Impulse $f(t) = (AM \cdot t)/(t \cdot \tau + a)$ - Figure A2.9
NAME of the function is DRF

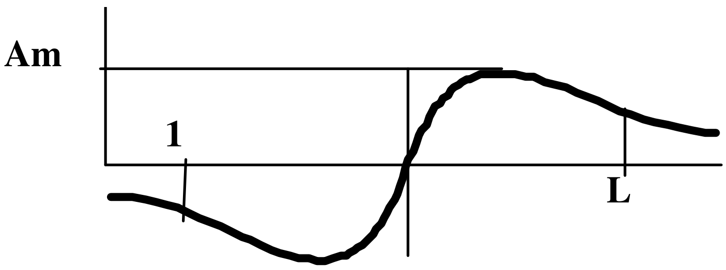


Figure A2.9 Impulse $f(t) = (AM \cdot t)/(t \cdot \tau + a)$

3. Creating the array of the periodic signals
NAME of the function is SIGM

```
B = SIGM (A, L, N, SKV, M1)
```

where A is the name of an array containing 1 impulse. A is the input array;
L is the number of samplings in the array A;
B is the name of the periodic signal array (output array);
N is the number of samplings in the output array B;

$$SKV = \frac{T_0}{\tau_p}$$

here T0 is the period (sec);
τ_p is the duration the impulse per one period (sec);
M1 is the number of the periods in the output array B.
The number N has to meet the condition:

$$N \geq (SKV \cdot L) \cdot M1$$

The array of the original signal A is formed by the functions 1.1…2.7.

For example**,** let the original signal be an infinite periodic sequence of the "cosine" impulse - see Figure A3.1.

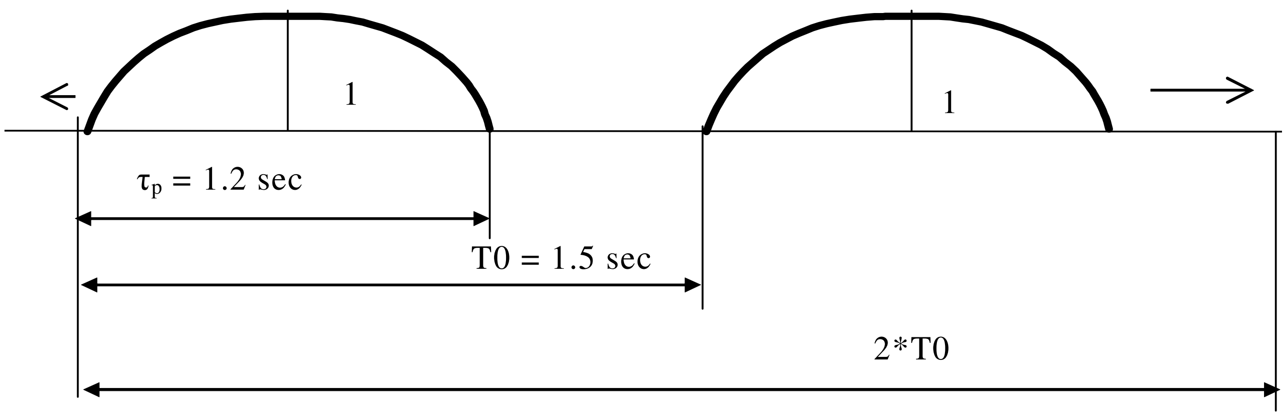


Figure A3.1. Original infinite periodic sequence of the "cosine" impulse

It is necessary to generate an array including 2 periods of this signal. The parameters of the signal are as follows.
The discrete time quanta is $\Delta t = 0.0469$ sec

1. Generation of a single "cosine" signal array (see paragraph 1.3 in the Appendix):
   $L = \tau_p/\Delta t = 1.2/0.0469 = 25.6 = 26$

```
A = CS(26,1,0);
plot(A)
pause
```

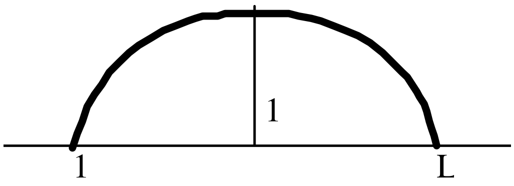


Figure. A3.1 Array of a single “cosine’ impulse

2. Generation of the two periods of the “cosine” signal array:
SKV = 1.5/1.2 = 1.25
N = 2*T0/ Δt = 2*1.5/0.0469 = 64

```
B = SIGM (A,26,64,1.25,2)
plot(B)
```

Result of the procedure is shown in Figure A3.2.

Figure. A3.2 Periodic signal array

4. Function for create the array with “Zero padding” approach.

Name of the function is ZEROF.

```
B = ZEROF (A, L, N, LH)
```

where A is the name of the original signal array (input array) – see Figure A4.1;
L is the number of samplings in the array A;
B is the “zero padding” array (output array) – see Figure A4.2;
N is the number of samplings in the array B;
LH is the number of zero samples in the beginning of the array B. LH = 1 is recommended.

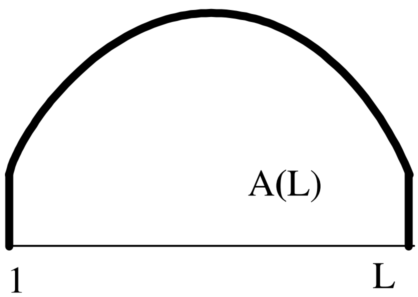


Figure. A4.1 Array of the original signal

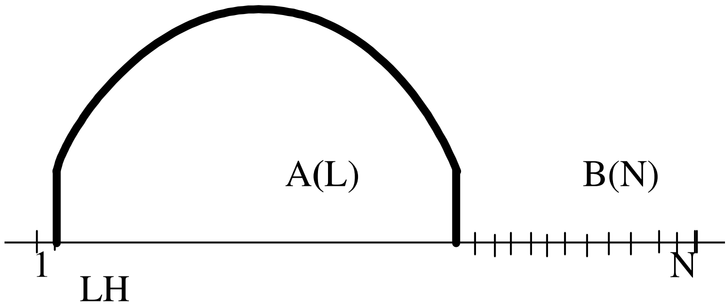


Figure. A4.1 “Zero padding” array

5. Function for creating WINDOW

5.1 Tukey “window”.
The name of the function is TUKEY. Figure A5.1

```
W =TUKEY(L)
```

where W is the array of the "window";
L is the number of samplings in the array W.

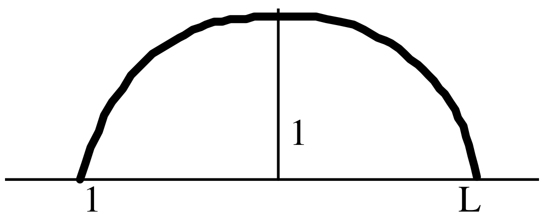


Figure. A5.1 Window Tukey

5.2 Hunn “window”.
Name of the function is HUNN. Figure A5.2

W = HUNN(L)

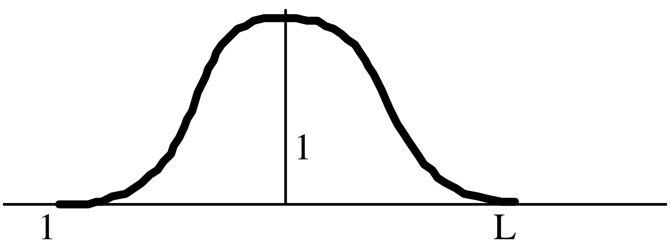


Figure. A5.2 Array of the HUNN window

where W is the array of the "window";
L is the number of samplings in the array W

5.3 One-side Tukey “window”.
The name of the function is TUKEYS. Figure A5.3

W =TUKEYS(L)

W is the array of "1/2 window";
L is the number of samplings in the array W.

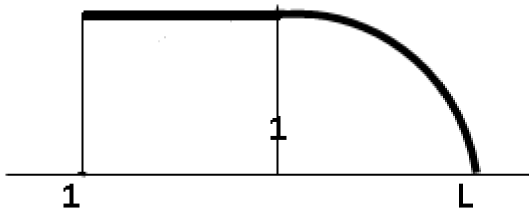


Figure. A5.3 Array of the TUKEYS window

5.4 One-side Hunn "window".
The name of the function is HUNNS. Figure A5.4

W =HUNNS(L)

where W is the array of the "window";
L is the number of samplings in the array W.

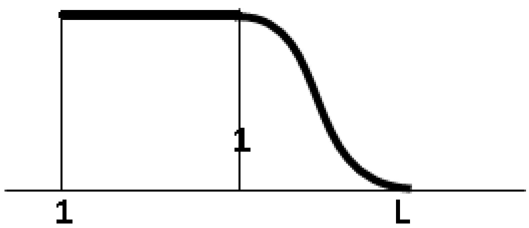


Figure. A5.4 Array of the HUNNS window

5. Cutting a part of array

Name of the function is SECTION.

B = SECTION (A, L, N, LH)

where A is the name of an array of the original signal (input);
L is the number of samplings in the array A;
B is the array as part of the array A (output) – see Figure A6.1
N is the number of samplings in the array B;
LH is the number of the point in the array A after which comes the piece cut out from the array A.

Array B contains N points cut out starting from LH + 1 up to LH + N number of points (Figure A6.2)

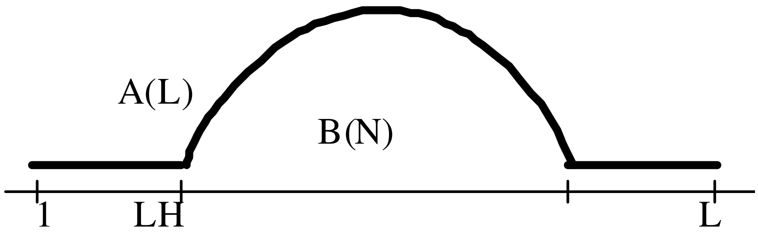


Figure A6.1 Input array A(L)

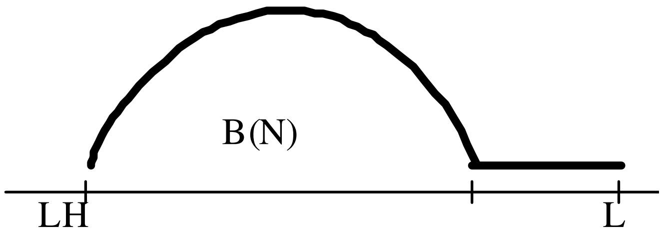


Figure A6.2 Array B as the part cut out of the array A

Коняхин Игорь Алексеевич

Digital signal processing. Basic procedures

Учебно-методическое пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49, литер А