Министерство науки и высшего образования Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт электроники и телекоммуникаций
Высшая школа электроники и микросистемной техники

*М. С. Енученко Н. В. Иванов*

# ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Конспект лекций

ПОЛИТЕХ-ПРЕСС
Санкт-Петербургский
политехнический университет
Петра Великого

Санкт-Петербург
2024

УДК 621.396.1
ББК 32.811.3
Е64

Р е ц е н з е н т ы:
Кандидат технических наук, доцент,
доцент Высшей школы прикладной физики и космических технологий
Санкт-Петербургского политехнического университета Петра Великого
*А. В. Рашич*
Инженер научной лаборатории
«Специальный технологический центр-Политех»
Санкт-Петербургского политехнического университета Петра Великого
*Е. В. Леонтьев*

*Енученко М. С.* **Основы цифровой обработки сигналов** : конспект лекций / М. С. Енученко, Н. В. Иванов. – СПб. : ПОЛИТЕХ-ПРЕСС, 2024. – 101 с.

Конспект лекций соответствует содержанию профильной дисциплины «Основы цифровой обработки сигналов» федерального государственного образовательного стандарта высшего образования по направлениям подготовки 11.04.02 «Инфокоммуникационные технологии и системы связи» и 11.04.04 «Электроника и наноэлектроника». В конспекте отражён базовый материал, необходимый для освоения данного курса и подготовки к практическим работам, семинарам, зачётам и экзаменам. Способствует получению знаний по работе с цифровыми сигналами.

Предназначен для студентов Института электроники и телекоммуникаций СПбПУ, обучающихся по курсам «Основы цифровой обработки сигналов» и направлениям подготовки 11.04.02 «Инфокоммуникационные технологии и системы связи» и 11.04.04 «Электроника и наноэлектроника».

Ил. 110. Библиогр.: 6 назв.

Печатается по решению
Совета по издательской деятельности Ученого совета
Санкт-Петербургского политехнического университета Петра Великого.

**ISBN 978-5-7422-8718-6**
doi:10.18720/SPBPU/2/id24-220

The Ministry of Science and Higher Education of the Russian Federation

PETER THE GREAT
ST. PETERSBURG POLYTECHNIC UNIVERSITY

Institute of Electronics and Telecommunications
Higher School of Electronics and Microsystems Engineering

*M. S. Yenuchenko N. V. Ivanov*

# BASICS OF DIGITAL SIGNAL PROCESSING

Work-book

POLYTECH PRESS
Peter the Great
St.Petersburg Polytechnic
University

Saint Petersburg
2024

R e v i e w e r s:
PhD, Associate Professor, Associate Professor
of the Higher School of Applied Physics and Space Technologies
Peter the Great St. Petersburg Polytechnic University
*A. V. Rashich*
Engineer of the scientific laboratory «Special Technological Center-Polytech»
Peter the Great St. Petersburg Polytechnic University
*E. V. Leontiev*

*Yenuchenko M. S.* **Basics of digital signal processing** : work-book / M. S. Yenuchenko, N. V. Ivanov. – St. Petersburg : POLYTECH-PRESS, 2024. – 101 p.

The lecture notes correspond to the content of the specialized disciplines “Fundamentals of Digital Signal Processing” of the federal state educational standard of higher education in the areas of training 11.04.02 “Infocommunication technologies and communication systems” and 11.04.04 “Electronics and nanoelectronics”. The notes reflect the basic material necessary for mastering this course and for preparing for practical work, seminars, tests and exams. Promotes the acquisition of knowledge on working with digital signals.

Intended for students of the Institute of Electronics and Telecommunications of SPbPU, studying in the courses “Fundamentals of Digital Signal Processing”, in the areas of training 11.04.02 “Infocommunication technologies and communication systems” and 11.04.04 “Electronics and nanoelectronics”.

Figures 110. References: 6 titles.

Printed by the Publishing Council
of Peter the Great St. Petersburg Polytechnic University Academic Council.

**ISBN 978-5-7422-8718-6**
doi:10.18720/SPBPU/2/id24-220

# Content

# Introduction

Modern telecommunications systems tend to increase application of digital signal processing systems due to their flexibility and scalability. As a result, digital signal processing (DSP) is an important area of research. Digital signal processing encompasses a variety of techniques for transmitting, receiving, analyzing, transforming, and synthesizing signals with digital devices. This work-book aims to become an entry point to the fundamental terms and concepts of DSP, with a focus on both theoretical foundations and implementations issues.

Digital signal processing systems have several features that should be carefully taken into account for correct processing of the signals. One of these features is discrete time. The discrete time changes an interpretation of signals in time-domain and their properties, particularly, spectrum representation. Besides, discrete time digital systems have another limitation – finite resolution, which results in inaccurate representation of signal values and errors in operations with them. Also, digital systems have very limited set of feasible operations and, consequently, digital processing requires special algorithms.

This work-book covers the following topics: introduction to discrete sequences and systems, feasible operations, mathematical instruments for processing and analyzing signals. The discrete Fourier transform (DFT) is discussed in details. The DFT is the essential instrument for digital signal processing, which features assumptions, leakage, axes conversion, symmetry etc. should be deeply understood before you start to use it. A fast implementation of DFT – fast Fourier transform – is described. Its different realizations are presented.

In terms of a device implementation, the work-book focuses on the basis of finite and infinite impulse response filters. Their different structures are discussed in details, particularly in terms of hardware costs, critical path and errors susceptibility. Approach for analysis of frequency and phase responses is discussed. It is considered that a FIR phase response demonstrates their properties to have linear phase response and constant group delay in the pass-band. Stability of IIR filter is discussed and a cascaded design is presented as its possible solution.

Then, the mentioned filters are considered in the typical processing tasks like averaging and sample rate conversion. Finally, the work-book briefly reminds the basis of analytic signals and applies considered material for their processing. Especially, Hilbert transform and its implementation are in focus.

# Chapter 1 Basic knowledge

## §1.1 Geometric progression and series

A geometric progression – a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed non-zero number called the common ratio, i.e.

$$b_n = b_{n-1}q,$$

where *n* – a term number, *q* – the common ratio. Each term of the geometric progression is given by

$$b_n = b_0 q^n,$$

where $b_0$ – the first term. For example

$$b_0 = 1; q = 2; b_n = 1 \cdot 2^n$$
$$b_n = 1, 2, 4, \dots$$

A geometric series is a sum of numbers in a geometric progression. That is

$$\sum_{n=0}^{N-1} b_n = \sum_{n=0}^{N-1} b_0 q^n$$

where *N* – a number of terms. To calculate a geometric series sum, let's remember the following equation

$$1 - q^N = (1-q)(1 + q + q^2 + \dots + q^{N-1}) \Leftrightarrow 1 + q + q^2 + \dots + q^{N-1} = \frac{1-q^N}{1-q}$$

Now we can derivate the formula of the sum

$$b_0 \sum_{n=0}^{N-1} q^n = b_0(1 + q + q^2 + \dots + q^{N-1}) = b_0 \frac{1-q^N}{1-q}$$

For our example, sum of *N* = 5 items will be

$$\sum_{n=0}^{5-1} 1 \cdot 2^n = 1 + 2 + 4 + 8 + 16 = 1 \cdot \frac{1-2^5}{1-2} = \frac{-31}{-1} = 31.$$

## §1.2 Complex numbers

Let's remember some points about complex numbers. A complex number is a pair of real numbers. And complex number *z* can be expressed as *a* + *bi*. In this case, *a* is a real part of *z* and *b* is an imaginary part of *z*. A mathematical form for this statement

$$\operatorname{Re} z = a; \operatorname{Im} z = b.$$

All complex numbers, except 0, have a polar form. We can write them like:

$$z = r(\cos\varphi + j\sin\varphi),$$

where *r* is an absolute value of complex number *z*, φ is an argument of complex number *z*. They are calculated by

$$r = |z| = \sqrt{(\operatorname{Re} z)^2 + (\operatorname{Im} z)^2} = \sqrt{a^2 + b^2}$$
$$\varphi = \arg z = \operatorname{arctg} \frac{b}{a}$$

If we remember Euler's formula

$$e^{jx} = \cos x + j\sin x,$$

the polar form can be rewritten as:

$$z = r \cdot e^{j\varphi}.$$

## §1.3 Trigonometric expressions

### *1.3.1 Basic formulas*

Let's start with the basic set of formulas:

$$\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta); \quad \sin(\alpha+\beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta);$$
$$\cos(\alpha-\beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta); \quad \sin(\alpha-\beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta);$$

Check some of them. Let:

$$\alpha = \frac{\pi}{6};\ \beta = \frac{\pi}{3}.$$

Then we can write:

$$\cos\left(\frac{\pi}{2}\right) = \cos\left(\frac{\pi}{6}+\frac{\pi}{3}\right) = \cos\left(\frac{\pi}{6}\right)\cdot\cos\left(\frac{\pi}{3}\right) - \sin\left(\frac{\pi}{6}\right)\cdot\sin\left(\frac{\pi}{3}\right) = \frac{\sqrt{3}}{2}\cdot\frac{1}{2} - \frac{1}{2}\cdot\frac{\sqrt{3}}{2} = 0,$$

$$\sin\left(\frac{\pi}{2}\right) = \sin\left(\frac{\pi}{6}+\frac{\pi}{3}\right) = \sin\left(\frac{\pi}{6}\right)\cdot\cos\left(\frac{\pi}{3}\right) + \cos\left(\frac{\pi}{6}\right)\cdot\sin\left(\frac{\pi}{3}\right) = \frac{1}{2}\cdot\frac{1}{2} + \frac{\sqrt{3}}{2}\cdot\frac{\sqrt{3}}{2} = \frac{1}{4}+\frac{3}{4} = 1.$$

Other set of formulas can be derived from the basic ones:

$$\cos(2\alpha) = \cos^2(\alpha) - \sin^2(\alpha);\qquad \cos(\alpha)\cos(\beta) = \frac{\cos(\alpha-\beta)+\cos(\alpha+\beta)}{2};$$

$$\sin(2\alpha) = 2\sin(\alpha)\cos(\alpha);\qquad \sin(\alpha)\sin(\beta) = \frac{\cos(\alpha-\beta)-\cos(\alpha+\beta)}{2};$$

$$\sin^2(\alpha) = \frac{1-\cos(2\alpha)}{2};\qquad \cos^2(\alpha) = \frac{1+\cos(2\alpha)}{2};$$

$$\cos(\alpha)+\cos(\beta) = \cos(x+y)+\cos(x-y) = 2\cos(x)\cos(y) = 2\cos\left(\frac{\alpha+\beta}{2}\right)\cdot\cos\left(\frac{\alpha-\beta}{2}\right)$$

$$\cos(\alpha)-\cos(\beta) = \cos(x+y)-\cos(x-y) = -2\sin(x)\sin(y) = -2\sin\left(\frac{\alpha+\beta}{2}\right)\cdot\sin\left(\frac{\alpha-\beta}{2}\right)$$

$$\sin(\alpha)+\sin(\beta) = \sin(x+y)+\sin(x-y) = 2\sin(x)\cos(y) = 2\sin\left(\frac{\alpha+\beta}{2}\right)\cdot\cos\left(\frac{\alpha-\beta}{2}\right)$$

$$\sin(\alpha)-\sin(\beta) = \sin(x+y)-\sin(x-y) = 2\cos(x)\sin(y) = 2\cos\left(\frac{\alpha+\beta}{2}\right)\cdot\sin\left(\frac{\alpha-\beta}{2}\right)$$

$$(\sin(\alpha))' = \cos(\alpha)\qquad (\cos(\alpha))' = -\sin(\alpha)$$

$$\int \cos(\alpha)\,dt = \sin(\alpha)+c\qquad \int \sin(\alpha)\,dt = -\cos(\alpha)+c$$

### 1.3.2 Integrals

Calculate some auxiliary integrals (assume that α∈ℤ and *T* – a period):

$$\int_0^T \cos\left(\frac{2\pi\alpha}{T}t\right)dt = \begin{cases} \int_0^T \underbrace{\cos(0)}_{1}\,dt, \text{if } \alpha = 0 \\ \left.\frac{\sin\left(\frac{2\pi\alpha}{T}t\right)}{\frac{2\pi\alpha}{T}}\right|_0^T, \text{if } \alpha \neq 0 \end{cases} = \begin{cases} t|_0^T, \text{if } \alpha = 0 \\ \frac{\sin(2\pi\alpha)-\sin(0)}{\frac{2\pi\alpha}{T}}, \text{if } \alpha \neq 0 \end{cases} = \begin{cases} T, \text{if } \alpha = 0 \\ 0, \text{if } \alpha \neq 0 \end{cases}$$

$$\int_0^T \sin\left(\frac{2\pi\alpha}{T}t\right)dt = \begin{cases} \int_0^T \underbrace{\sin(0)}_{0}\,dt, \text{if } \alpha = 0 \\ \left.\frac{-\cos\left(\frac{2\pi\alpha}{T}t\right)}{\frac{2\pi\alpha}{T}}\right|_0^T, \text{if } \alpha \neq 0 \end{cases} = \begin{cases} 0, \text{if } \alpha = 0 \\ \frac{-\cos(2\pi\alpha)+\cos(0)}{\frac{2\pi\alpha}{T}}, \text{if } \alpha \neq 0 \end{cases} = 0$$

$$\boxed{\int_0^T \cos\left(\frac{2\pi\alpha}{T}t\right)dt = \begin{cases} T, \text{if } \alpha = 0 \\ 0, \text{if } \alpha \neq 0 \end{cases};\ \int_0^T \sin\left(\frac{2\pi\alpha}{T}t\right)dt = 0}$$

### 1.3.3 Orthogonality

Now, let's have a look at the following integrals

$$\int_0^T \sin\left(\frac{2\pi k}{T}t\right)\sin\left(\frac{2\pi l}{T}t\right)dt;\quad \int_0^T \cos\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt;\quad \int_0^T \sin\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt;$$

$$k, l \in \mathbb{Z}\backslash\{0\}.$$

The first one.

$$\int_0^T \sin\left(\frac{2\pi k}{T}t\right)\sin\left(\frac{2\pi l}{T}t\right)dt = \int_0^T \frac{\cos\left(\frac{2\pi(k-l)}{T}t\right) - \cos\left(\frac{2\pi(k+l)}{T}t\right)}{2}dt =$$

$$= \begin{cases} \underbrace{\frac{1}{2}\int_0^T \cos(0\cdot t)\,dt}_{T} - \underbrace{\frac{1}{2}\int_0^T \cos\left(\frac{2\pi\cdot 2l}{T}t\right)dt}_{0}, \text{if } k = l \\ \underbrace{\frac{1}{2}\int_0^T \cos\left(-\frac{2\pi\cdot 2l}{T}t\right)dt}_{0} - \underbrace{\frac{1}{2}\int_0^T \cos(0\cdot t)\,dt}_{T}, \text{if } k = -l \\ 0, \text{if } k \neq l \end{cases} = \begin{cases} \pm\frac{T}{2}, \text{if } k = \pm l \\ 0, \text{if } k \neq \pm l \end{cases}$$

The second one is similar.

$$\int_0^T \cos\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt = \int_0^T \frac{\cos\left(\frac{2\pi(k-l)}{T}t\right) + \cos\left(\frac{2\pi(k+l)}{T}t\right)}{2}dt =$$

$$= \begin{cases} \underbrace{\frac{1}{2}\int_0^T \cos(0\cdot t)\,dt}_{T} + \underbrace{\frac{1}{2}\int_0^T \cos\left(\frac{2\pi\cdot 2l}{T}t\right)dt}_{0}, \text{if } k = l \\ \underbrace{\frac{1}{2}\int_0^T \cos\left(-\frac{2\pi\cdot 2l}{T}t\right)dt}_{0} + \underbrace{\frac{1}{2}\int_0^T \cos(0\cdot t)\,dt}_{T}, \text{if } k = -l \\ 0, \text{if } k \neq l \end{cases} = \begin{cases} \frac{T}{2}, \text{if } k = \pm l \\ 0, \text{if } k \neq \pm l \end{cases}$$

The third one looks different.

$$\int_0^T \sin\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt = \int_0^T \frac{\sin\left(\frac{2\pi(k-l)}{T}t\right) + \sin\left(\frac{2\pi(k+l)}{T}t\right)}{2}dt = 0$$

As a result, we have

$$\int_0^T \sin\left(\frac{2\pi k}{T}t\right)\sin\left(\frac{2\pi l}{T}t\right)dt = \begin{cases} 0, \text{if } k \neq \pm l \\ \pm\frac{T}{2}, \text{if } k = \pm l \end{cases};$$

$$\int_0^T \cos\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt = \begin{cases} 0, \text{if } k \neq \pm l \\ \frac{T}{2}, \text{if } k = \pm l \end{cases}$$

$$\int_0^T \sin\left(\frac{2\pi k}{T}t\right)\cos\left(\frac{2\pi l}{T}t\right)dt = 0$$

This all means that functions like:

$$\sin\left(\frac{2\pi k}{T}t\right) \text{ and } \cos\left(\frac{2\pi l}{T}t\right), \text{if } k \neq l \text{ and } k, l \in \mathbb{Z}\backslash\{0\}$$

are orthogonal.

## §1.4 Linear operators

An operator – a transformation of one set into another. A linear operator – an operator that fulfills the next statement:

$$\text{for } \forall\, \bar{x}, \bar{y}, \lambda, \mu$$

$$A(\lambda\bar{x} + \mu\bar{y}) = \lambda A(\bar{x}) + \mu A(\bar{y})$$

where *A* – an operator. Remind some basic information about linear operators.

Operation definitions:

1. $(A + B)\bar{x} = A\bar{x} + B\bar{x}$
2. $(\lambda A)\bar{x} = \lambda \cdot A\bar{x}$
3. $(AB)\bar{x} = A(B\bar{x})$

Properties with constants:

1. $(\alpha\beta)A = \alpha(\beta A)$;
2. $\lambda(AB) = (\lambda A)B = A(\lambda B)$ (associative multiplication).
3. $(\alpha + \beta)A = \alpha A + \beta A$ (distributive property);
4. $\alpha(A + B) = \alpha A + \alpha B$ (distributive property);

Properties with operators:

1. $A + B = B + A$ (commutative addition);
2. $(AB)C = A(BC)$ (associative multiplication);
3. $(A + B)C = AC + BC$ (distributive property);
4. $A(B + C) = AB + AC$ (distributive property);

In general case, commutativity of multiplication is not performed, i.e. *AB* ≠ *BA*. However, for symmetric linear operators (i.e. that have symmetric matrices) is that – *AB* = *BA*. We can show it. Symmetric operator is a bilinear operator that satisfy definition that for any vectors *x* and *y*

$$A(\bar{x};\bar{y}) = A(\bar{y};\bar{x}) \Leftrightarrow (A\bar{x};\bar{y}) = (\bar{y};A\bar{x})$$

Then

$$(AB\bar{x};\bar{y}) = (A(B\bar{x});\bar{y}) = (B\bar{x};A\bar{y}) = \big(\bar{x};B(A\bar{y})\big) = (\bar{x};BA\bar{y}) \Leftrightarrow AB = BA$$

In our course, discussed systems will have this symmetry property and, as consequence, the commutativity of multiplication (*AB* = *BA*).

## §1.5 Convolution

### *1.5.1 Linear convolution*

A convolution is defined by means of formula

$$y(t) = (h * x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau$$

That is, a convolution is a mathematical operation over two functions *x*(*t*) and *h*(*t*) producing the third function *y*(*t*). The convolution is designated by asterisk ("*"). It is seen from the definition that the convolution is a linear operator because the linearity is performed

$$\big(h * (\alpha x + \beta y)\big)(t) = \alpha(h * x)(t) + \beta(h * y)(t)$$

due to the linearity of integral. Also it can be noticed that the convolution is symmetric, that is

$$(h * x)(t) = (x * h)(t)$$

Let's look at this

$$(h * x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau = |t-\tau \to p; \tau \to t-p| = \int_{+\infty}^{-\infty} x(p)h(t-p)d(t-p) = \int_{-\infty}^{+\infty} x(p)h(t-p)dp$$
$$= (x * h)(t)$$

The classic example is a convolution of two rectangular functions

$$h(t) = x(t) = \begin{cases} 1 \text{ for } |t| \le a \\ 0 \text{ for } |t| > a \end{cases}$$

Function *h*(*t*) and its convolution illustration are depicted in Figure 1.1 and 1.2 respectively.

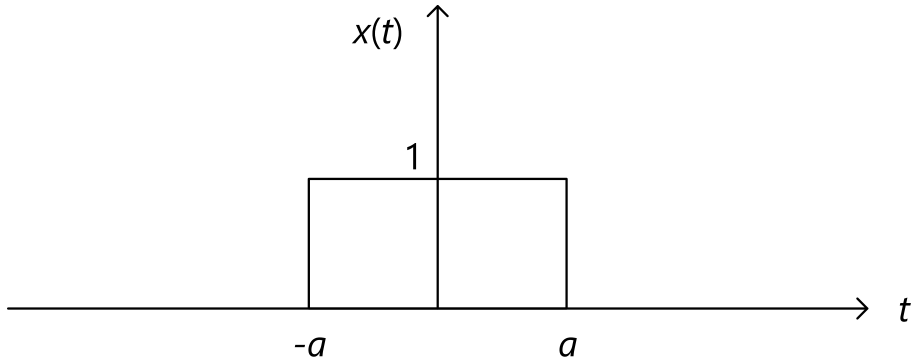



Figure 1.1 – A function *x*(*t*)

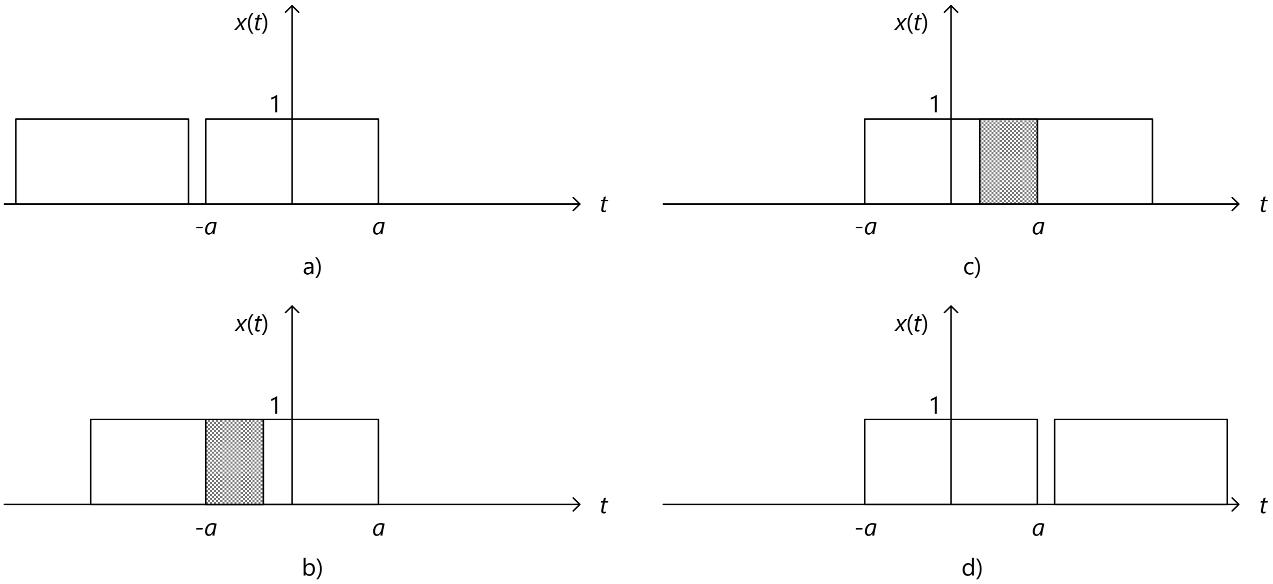



Figure 1.2 – The convolution illustration (a) $t<-a$ (b) $-a<t<0$ (c) $0<t<a$ (d) $t>a$

$$y(t) = (h * x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau = \int_{-a}^{a} x(t-\tau)d\tau = \int_{t-a}^{t+a} x(\tau)d\tau = \begin{cases} 0 \text{ for } t < -2a \\ \int_{-a}^{t+a} d\tau \text{ for } -2a \le t < 0 \\ \int_{t-a}^{a} d\tau \text{ for } 0 \le t \le 2a \\ 0 \text{ for } t > 2a \end{cases} =$$

$$= \begin{cases} 0 \text{ for } |t| > 2a \\ t + a + a \text{ for } -2a \le t < 0 \\ a - t + a \text{ for } 0 \le t \le 2a \end{cases} = \begin{cases} 0 \text{ for } |t| > 2a \\ 2a + t \text{ for } -2a \le t < 0 \\ 2a - t \text{ for } 0 \le t \le 2a \end{cases} = \begin{cases} 0 \text{ for } |t| > 2a \\ 2a - |t| \text{ for } |t| \le 2a \end{cases}$$

The result of the calculation can be presented as in Figure 1.3.

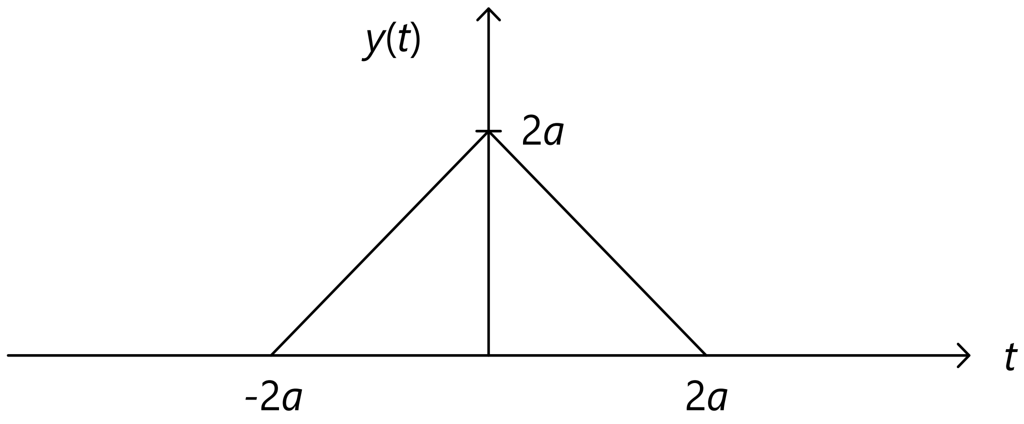



Figure 1.3 – Convolution result

*1.5.2 Cyclic convolution*

If $x_T(t)$ and $h_T(t)$ are periodic functions with period $T$, i.e.

$$x_T(t) = \sum_{k=-\infty}^{+\infty} x(t-kT) = \sum_{k=-\infty}^{+\infty} x(t+kT)\,;\; h_T(t) = \sum_{k=-\infty}^{+\infty} h(t-kT) = \sum_{k=-\infty}^{+\infty} h(t+kT),$$

where $x(t)$ and $h(t)$ are aperiodic functions, then

*Circular convolution*

$$(h * x_T)(t) = \int_{-\infty}^{+\infty} h(\tau)x_T(t-\tau)d\tau$$

*Periodic convolution*

$$(h_T * x_T)(t) = \int_{t_0}^{t_0+T} h_T(\tau)x_T(t-\tau)d\tau$$

*Equivalence*

$$(h * x_T)(t) = (h_T * x_T)(t)$$

Prove their equivalence.

$$(h_T * x_T)(t) = \int_{t_0}^{t_0+T} h_T(\tau)x_T(t-\tau)d\tau = \int_{t_0}^{t_0+T} \sum_{k=-\infty}^{+\infty} h(\tau+kT)\, x_T(t-\tau)d\tau = \sum_{k=-\infty}^{+\infty} \int_{t_0}^{t_0+T} h(\tau+kT)x_T(t-\tau)d\tau$$

$$= \begin{vmatrix} \tau + kT \to p \\ \tau \to p - kT \\ d\tau \to dp \end{vmatrix} = \sum_{k=-\infty}^{+\infty} \int_{t_0+kT}^{t_0+kT+T} h(p)\,\underbrace{x_T(t-p+kT)}_{x_T(t-p)}\,dp = \int_{-\infty}^{+\infty} h(p)x_T(t-p)dp = (h * x_T)(t)$$

In other words, a linear convolution of periodic functions is equal to a periodic convolution for their common period.

## §1.6 Fourier series

If a function is periodic, we can expand it into a series of harmonic functions (sine and cosine). Such a series is called a Fourier Series (FS). For periodic function $x_T(t)$ with a period $T$ the Fourier series is described with following formulas

*Fourier Series*

$$x_T(t) = \sum_{k=-\infty}^{+\infty} c_k e^{j\omega_k t}\,;\, \omega_k = \frac{2\pi k}{T}\,;\, c_k = \frac{1}{T}\int_0^T x_T(t)\, e^{-j\omega_k t} dt$$

Fourier Series coefficients represent magnitude and phase of corresponding frequencies. Let's have a look at an example. Let

$$x_T(t) = \sin\left(\frac{2\pi m}{T} t\right) = \sin(\omega_m t)\,;\, \omega_m = \frac{2\pi m}{T}$$

Calculate coefficients of the Fourier Series (remember results from §1.3.3).

$$c_k = \frac{1}{T}\int_0^T \sin(\omega_m t)\, e^{-j\omega_k t} dt = \frac{1}{T}\underbrace{\int_0^T \sin(\omega_m t)\cos(\omega_k t)\, dt}_{0} - j\frac{1}{T}\underbrace{\int_0^T \sin(\omega_m t)\sin(\omega_k t)\, dt}_{\pm\frac{T}{2}} = \begin{cases} \mp j\frac{1}{T}\cdot\frac{T}{2}, \text{if } k = \pm m \\ 0, \text{if } k \neq \pm m \end{cases} =$$

$$= \begin{cases} \mp j\frac{1}{2}, \text{if } k = \pm m \\ 0, \text{if } k \neq \pm m \end{cases}$$

For the cosine function, it will be

$$c_k = \frac{1}{T}\int_0^T \cos(\omega_m t)\, e^{-j\omega_k t} dt = \frac{1}{T}\underbrace{\int_0^T \cos(\omega_m t)\cos(\omega_k t)\, dt}_{\frac{T}{2}} - j\frac{1}{T}\underbrace{\int_0^T \cos(\omega_m t)\sin(\omega_k t)\, dt}_{0} = \begin{cases} \frac{1}{T}\cdot\frac{T}{2}, \text{if } k = \pm m \\ 0, \text{if } k \neq \pm m \end{cases}$$

$$= \begin{cases} \frac{1}{2}, \text{if } k = \pm m \\ 0, \text{if } k \neq \pm m \end{cases}$$

## §1.7 Integral Fourier Transform

### *1.7.1 Definition*

The Integral Fourier Transform (IFT) helps us to know a magnitude and a phase of frequencies in our signal. It performs a transformation from a time domain into a frequency domain and back.

*Direct Integral Fourier Transform (Direct IFT)*

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt.$$

*Inverse Integral Fourier Transform (Inverse IFT)*

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega) e^{j\omega t} d\omega.$$

The Direct IFT provides us a complex function of a frequency. The Inverse IFT provides a complex (in general case) function of time.

### *1.7.2 Spectrum of signal*

A spectrum is a signal representation in the frequency domain. To get this representation, Fourier Transform is needed. To put it simply, *F*(ω) is a spectrum density function or just a spectrum. As *F*(ω) is a complex function, the spectrum is described by means of two characteristics: magnitude and phase.

The simplest example for a signal spectrum is a spectrum of a harmonic function x(*t*). The harmonic is shown in figure 1.4 and has amplitude 1 and frequency $f_0$. Its spectrum *X*(*f*) (magnitude and phase) is presented in figure 1.5; there is a tone at frequency $f_0$ with magnitude value 0.5 and phase value -π/2. Calculation is the following

$$\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt = \int_{-\infty}^{+\infty} \sin(\omega_0 t)e^{-j\omega t}dt = \int_{-\infty}^{+\infty} \frac{e^{j\omega_0 t} - e^{-j\omega_0 t}}{2j} e^{-j\omega t}dt$$

$$= \frac{1}{2j}\int_{-\infty}^{+\infty} e^{j(\omega_0-\omega)t}dt - \frac{1}{2j}\int_{-\infty}^{+\infty} e^{-j(\omega_0+\omega)t}dt = \frac{\delta(\omega_0-\omega) - \delta(\omega_0+\omega)}{2j}$$

$$= -\frac{j}{2}\big(\delta(\omega_0-\omega) - \delta(\omega_0+\omega)\big)$$

(We discuss Dirac delta function more detailed in §1.10.)

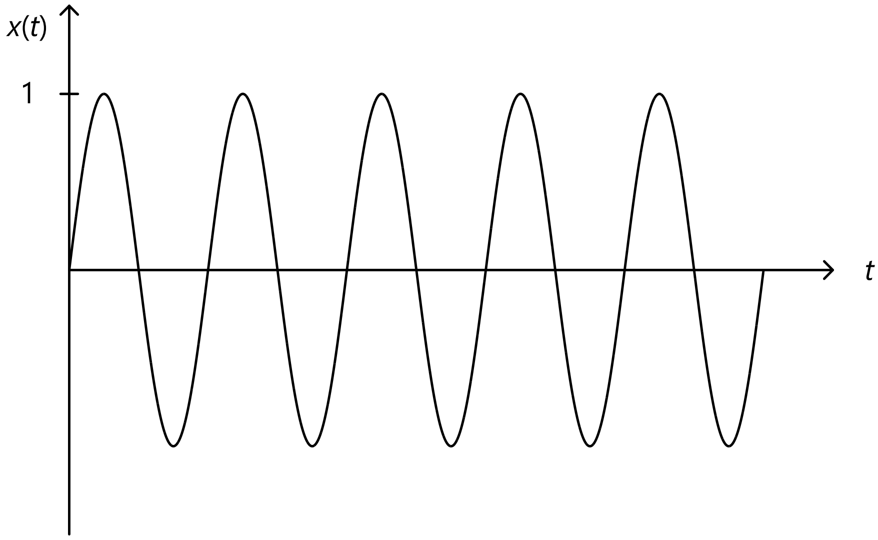



Figure 1.4 – A sine function $x(t)$

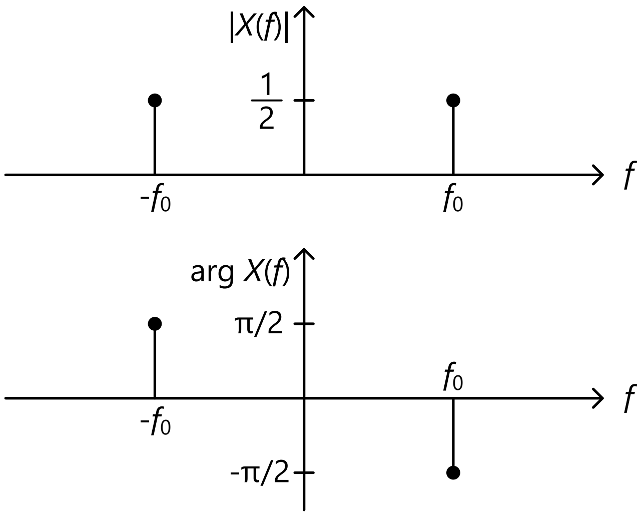



Figure 1.5 – A magnitude and a phase of $x(t)$

Best practice for the FT calculation is to evaluate a constant level (by averaging all samples) and subtract it from the signal. In most applications, the constant level doesn't carry information, so there is no need to represent it in the spectrum. Also note that a real signal spectrum is symmetric: the magnitude is symmetric, the phase is anti-symmetric.

*1.7.3 Properties*

1. *Linearity*

It can be seen from Fourier Transform definition that the next statement is true:

$$\mathcal{F}\{\alpha x(t) + \beta y(t)\} = \alpha\mathcal{F}\{x(t)\} + \beta\mathcal{F}\{y(t)\}$$

for all $x(t)$, $y(t)$, α, β. It means that Fourier Transform is a linear operator and has all its properties.

2. *Invertibility*

Declaration

The Fourier Transform is also an invertible operator. That is:

$$\mathcal{F}^{-1}\{\mathcal{F}\{x(t)\}\} = x(t) \text{ and } \mathcal{F}\{\mathcal{F}^{-1}\{X(\omega)\}\} = X(\omega).$$

Proof

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt\,; f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega$$

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega x}d\omega = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt\right)e^{j\omega x}d\omega = \frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)\underbrace{\int_{-\infty}^{+\infty} e^{j\omega(x-t)}d\omega}_{2\pi\cdot\delta(x-t)}\,dt$$

$$= \frac{2\pi}{2\pi}\int_{-\infty}^{+\infty} f(t)\cdot\delta(x-t)dt = f(x)$$

*1.7.4 Sine and cosine transforms*

Fourier Transform has sine and cosine forms. Let's look at them

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt = \int_{-\infty}^{+\infty} f(t)(\cos\omega t - j\sin\omega t)dt =$$

$$= \int_{-\infty}^{+\infty} f(t)\cos\omega t\,dt - j\int_{-\infty}^{+\infty} f(t)\sin\omega t\,dt = F_c(\omega) - jF_s(\omega)$$

where $F_c(\omega)$ – the Fourier cosine transform, $F_s(\omega)$ – the Fourier sine transform, and their formulas:

$$F_c(\omega) = \int_{-\infty}^{+\infty} f(t)\cos\omega t\,dt\,;\; F_s(\omega) = \int_{-\infty}^{+\infty} f(t)\sin\omega t\,dt$$

For an even function *f*(*t*) the Fourier cosine transform can be simplified:

$$F_c(\omega) = \int_{-\infty}^{+\infty} f(t)\cos\omega t\,dt = \int_{-\infty}^{0} f(t)\cos\omega t\,dt + \int_{0}^{+\infty} f(t)\cos\omega t\,dt =$$

$$= \int_{+\infty}^{0} f(-t)\cos\omega(-t)\,d(-t) + \int_{0}^{+\infty} f(t)\cos\omega t\,dt = \int_{0}^{+\infty} f(t)\cos\omega t\,dt +$$

$$+\int_{0}^{+\infty} f(t)\cos\omega t\,dt = 2\int_{0}^{+\infty} f(t)\cos\omega t\,dt$$

That is:

$$F_c(\omega) = 2\int_{0}^{+\infty} f(t)\cos\omega t\,dt$$

The same story with an odd function *f*(*t*) and the sine transform. The Fourier sine transform can also be simplified:

$$F_s(\omega) = \int_{-\infty}^{+\infty} f(t)\sin\omega t\,dt = \int_{-\infty}^{0} f(t)\sin\omega t\,dt + \int_{0}^{+\infty} f(t)\sin\omega t\,dt =$$

$$= \int_{+\infty}^{0} f(-t)\sin\omega(-t)\,d(-t) + \int_{0}^{+\infty} f(t)\sin\omega t\,dt = \int_{0}^{+\infty} f(t)\sin\omega t\,dt +$$

$$+\int_{0}^{+\infty} f(t)\sin\omega t\,dt = 2\int_{0}^{+\infty} f(t)\sin\omega t\,dt$$

That is:

$$F_s(\omega) = 2\int_{0}^{+\infty} f(t)\sin\omega t\, dt$$

*1.7.5 Shifting theorem*

Declaration

Now we discuss what happens with a spectrum of a time shifted signal. Shifting theorem states that

$$y(t) = x(t-\tau) \leftrightarrow Y(\omega) = e^{-j\omega\tau}\cdot X(\omega)$$
$$Y(\omega) = X(\omega-\tau) \leftrightarrow y(t) = e^{j\omega\tau}\cdot x(t).$$

Proof

Assume that our original signal *x*(*t*) is shifted in time by τ. It can be written as:

$$y(t) = x(t-\tau);$$

And we know that:

$$\mathcal{F}\{y(t)\} = Y(\omega);\ \mathcal{F}\{x(t)\} = X(\omega)$$

Do FT from both sides:

$$\mathcal{F}\{y(t)\} = \mathcal{F}\{x(t-\tau)\}$$

$$Y(\omega) = \int_{-\infty}^{+\infty} x(t-\tau)e^{-j\omega t}dt = |t-\tau\to p; t\to p+\tau| = \int_{-\infty}^{+\infty} x(p)e^{-j\omega(p+\tau)}d(p+\tau) = e^{-j\omega\tau}\int_{-\infty}^{+\infty} x(p)e^{-j\omega p}dp$$
$$= e^{-j\omega\tau}\cdot\mathcal{F}\{x(t)\} = e^{-j\omega\tau}\cdot X(\omega)$$

That is:

$$Y(\omega) = e^{-j\omega\tau}\cdot X(\omega)$$

For inverse FT it will be similar:

$$Y(\omega) = X(\omega-\Omega)$$
$$\mathcal{F}^{-1}\{Y(\omega)\} = \mathcal{F}^{-1}\{X(\omega-\Omega)\}$$
$$y(t) = \int_{-\infty}^{+\infty} X(\omega-\Omega)e^{j\omega t}d\omega = e^{j\Omega t}\int_{-\infty}^{+\infty} X(p)e^{jpt}dp = e^{j\Omega t}\cdot\mathcal{F}^{-1}\{X(\omega)\} = e^{j\Omega t}\cdot x(t)$$
$$y(t) = e^{j\Omega t}\cdot x(t)$$

Here we see that real signal transforms into complex signal (spectrum now is not symmetric). We discuss such signals in Chapter 10 Analytic signal.

*1.7.6 Theorem of convolution*

Assume that:

$$\mathcal{F}\{y(t)\} = Y(\omega);\ \mathcal{F}\{x(t)\} = X(\omega);\ \mathcal{F}\{h(t)\} = H(\omega)$$

Then next equivalences are performed:

$$y(t) = (x*h)(t) \leftrightarrow Y(\omega) = X(\omega)\cdot H(\omega)$$
$$y(t) = x(t)\cdot h(t) \leftrightarrow Y(\omega) = \frac{1}{2\pi}\cdot(X*H)(\omega)$$

Let's prove it. Start with the first equivalence:

$$y(t) = (x*h)(t) = \int_{-\infty}^{+\infty} x(t-\tau)h(\tau)d\tau$$

$$Y(\omega) = \mathcal{F}\{y(t)\} = \mathcal{F}\{(x*h)(t)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x(t-\tau)h(\tau)d\tau\, e^{-j\omega t}dt =$$
$$= \int_{-\infty}^{+\infty} h(\tau)\left(\int_{-\infty}^{+\infty} x(t-\tau)\, e^{-j\omega t}dt\right)d\tau = \int_{-\infty}^{+\infty} h(\tau)\cdot e^{-j\omega\tau}\cdot X(\omega)d\tau = X(\omega)\int_{-\infty}^{+\infty} h(\tau)e^{-j\omega\tau}d\tau = X(\omega)\cdot H(\omega)$$

The second equivalence:

$$Y(\omega) = (X * H)(\omega) = \int_{-\infty}^{+\infty} X(\omega - \Omega)H(\Omega)d\Omega$$

$$y(t) = \mathcal{F}^{-1}\{Y(\omega)\} = \mathcal{F}^{-1}\left\{\frac{1}{2\pi} \cdot (X * H)(\omega)\right\} = \frac{1}{2\pi} \cdot \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} X(\omega - \Omega)H(\Omega)d\Omega\, e^{j\omega t}d\omega =$$

$$= \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(\Omega)\left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega - \Omega)\, e^{j\omega t}d\omega\right)d\Omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(\Omega) \cdot e^{j\Omega t} \cdot x(t)d\Omega = x(t) \cdot \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(\Omega)e^{j\Omega t}d\Omega$$

$$= x(t) \cdot h(t)$$

*1.7.7 General formulas*

Declaration

If $f(t) \leftrightarrow F(\omega)$, then the following equivalences are correct:

1. $\frac{df(t)}{dt} \leftrightarrow j\omega F(\omega)$
2. $\int f(t)dt \leftrightarrow \frac{F(\omega)}{j\omega}$
3. $f(\alpha t) \leftrightarrow \frac{1}{\alpha} F\left(\frac{\omega}{\alpha}\right)$

Proof

Initially we know that:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega\,;$$

1. *Differentiation*

$$\frac{df(t)}{dt} = \frac{d}{dt}\left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega\right) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{d}{dt}\left(F(\omega)e^{j\omega t}\right)d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} j\omega F(\omega)e^{j\omega t}d\omega = \mathcal{F}^{-1}\{j\omega F(\omega)\}$$

2. *Integration*

$$\int f(t)dt = \int\left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega\right)dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int F(\omega)e^{j\omega t}dt\right)d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{F(\omega)}{j\omega} e^{j\omega t}d\omega = \mathcal{F}^{-1}\left\{\frac{F(\omega)}{j\omega}\right\}$$

3. *Scaling*

$$\int_{-\infty}^{+\infty} f(\alpha t)e^{-j\omega t}dt = \left|\alpha t \to p; t \to \frac{p}{\alpha}; dt \to \frac{dp}{\alpha}\right| = \int_{-\infty}^{+\infty} f(p)e^{-j\omega\frac{p}{\alpha}}\frac{dp}{\alpha} = \frac{1}{\alpha} \int_{-\infty}^{+\infty} f(p)e^{-j\frac{\omega}{\alpha}p}dp = \frac{1}{\alpha}\, F\left(\frac{\omega}{\alpha}\right)$$

## §1.8 Laplace Transform

*1.8.1 Definition and properties*

Definition

Fourier Transform gives us information only about homogeneous in time processes. To analyze transition processes in a linear system, it needs to perform the Laplace transform.

*Direct Laplace Transform*

$$F(p) = \mathcal{L}\{f(t)\} = \int_{0}^{+\infty} f(t)e^{-pt}dt$$

*Inverse Laplace Transform*

$$f(t) = \mathcal{L}^{-1}\{f(t)\} = \frac{1}{2\pi j}\int_{\sigma_0 - j\infty}^{\sigma_0 + j\infty} F(p)e^{pt}dp$$

where $\sigma_0$ – an arbitrary real number, $p$ – a complex variable that is written usually as

$$p = \sigma + j\omega$$

The Laplace transform simplifies solving of differential linear equations by means of conversion them into regular linear equations.

Properties

3. *Linearity*

As we can see from the definition, the Laplace transform is a linear operator, because

$$\mathcal{L}\{\alpha x(t) + \beta y(t)\} = \alpha\mathcal{L}\{x(t)\} + \beta\mathcal{L}\{y(t)\}$$

4. *Invertibility*

This transform like the Fourier Transform is also invertible

$$\mathcal{L}^{-1}\{\mathcal{L}\{f(t)\}\} = f(t);\ \mathcal{L}\{\mathcal{L}^{-1}\{F(p)\}\} = F(p)$$

5. *Theorem of convolution*

The same theorem of convolution is fulfilled for the Laplace Transform:

$$y(t) = (x * h)(t) \leftrightarrow Y(p) = X(p) \cdot H(p)$$

$$y(t) = x(t) \cdot h(t) \leftrightarrow Y(p) = \frac{1}{2\pi j}(X * H)(p)$$

### *1.8.2 Impulse response and transfer function*

For linear systems, we can define a transfer function $T(p)$:

$$T(p) = \frac{Y(p)}{X(p)},$$

where $X(p)$ – the Laplace transform of the input function, $Y(p)$ – the Laplace transform of the output function. In other words, the output of our system can be calculated by:

$$Y(p) = T(p) \cdot X(p)$$

By theorem of convolution, it is equivalent to:

$$y(t) = (x * h)(t)$$

where $h(t)$ – the inverse Laplace Transform of $T(p)$ or an impulse response $h(t)$.

$$T(p) = \int_{0}^{+\infty} h(t)e^{-pt}dt$$

If the input is a delta function, then

$$y(t) = \int_{-\infty}^{+\infty} \delta(t - \tau)h(\tau)d\tau = h(t)$$

In other words, the output will be equaled to the impulse response, if the input is a delta function. Let's designated the following integral as $A$

$$\int_{-\infty}^{+\infty} h(\tau)d\tau = A$$

$A$ corresponds to the output of the system, if the input is a unity valued constant level. So that $A^{-1}$ corresponds to a transfer coefficient for the constant level:

$$\frac{1}{A} = \frac{1}{\int_{-\infty}^{+\infty} h(\tau)d\tau} = T(0).$$

*1.8.3 Poles and stability*

Typically, the transfer function is a fraction of polynomials

$$T(p) = \frac{A(p)}{B(p)} = \frac{\sum_{i=0}^{M} b_i p^i}{\sum_{i=0}^{N} a_i p^i}$$

Let $p_k$ are roots (also called as *poles*) of the polynomial $B(p)$. Then

$$T(p) = \frac{A(p)}{\sum_{k=0}^{K}(p - p_k)};$$

The Inverse Laplace transform from this expression will be

$$h(t) = \sum_{k=0}^{K} c_k e^{p_k t};$$

where $c_k$ – coefficients obtained by the Inverse Laplace transform. Now, analyze different cases for $p_k$. These cases are shown in figure 1.3.

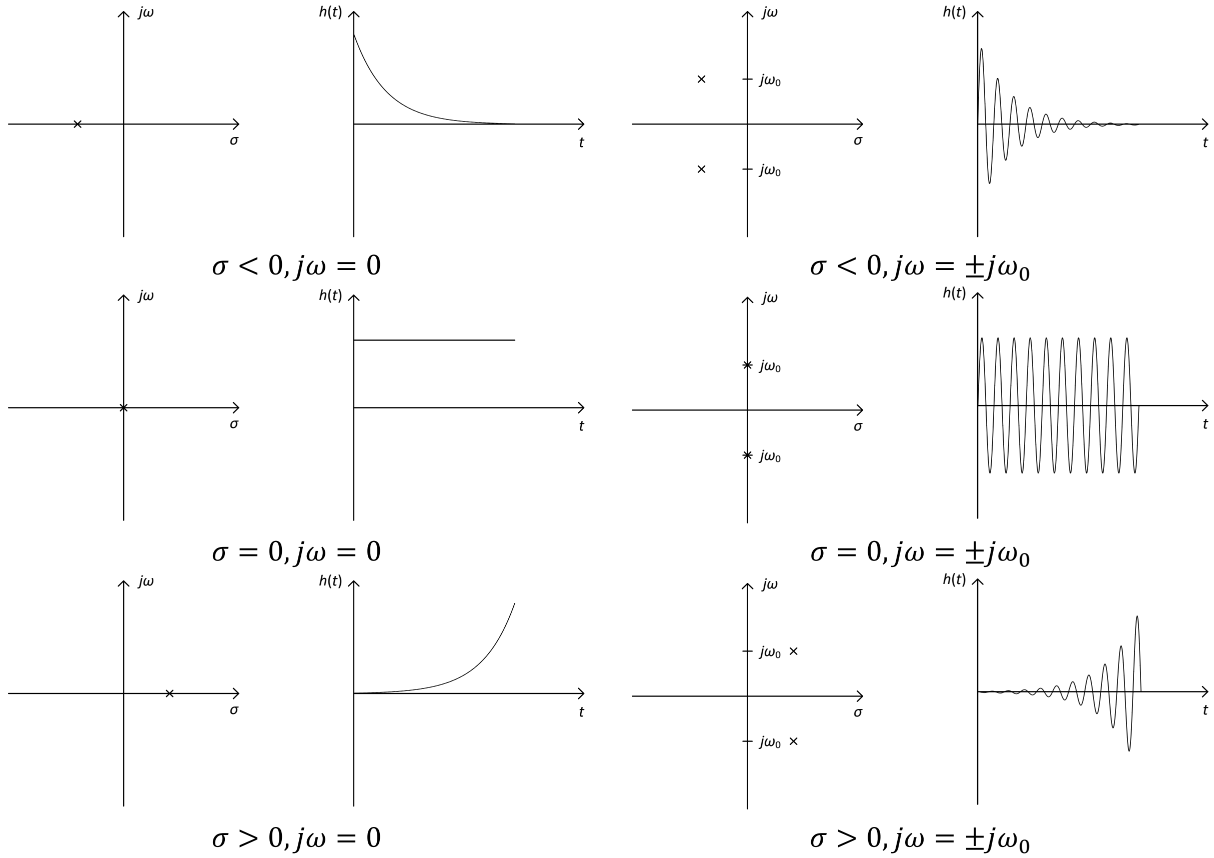



Figure 1.3 – Different cases of poles and their impulse responses

- If $\sigma < 0$ (a real part of $p_k$), then an impulse response $h(t)$ approaches zero and system is stable;
- If $\sigma = 0$, then $h(t)$ is constant and system is conditionally stable;
- If $\sigma < 0$, then $h(t)$ approaches the infinity and system is unstable.

Thus, if all poles of a transfer function are located in the left half of $p$-plane, then the system will be stable. If there is a pole in the right half of $p$-plane, then system will be unstable. Also note that transfer function with real coefficients always has conjugated pole pairs.

## §1.9 Z-transform

*1.9.1 Definition*

The Laplace transform helps to analyze continuous time system. What about discrete time system? For this purpose, there is a Z-transform.

*Direct Z-transform*

$$H(z) = \mathcal{Z}\{h(n)\} = \sum_{n=-\infty}^{+\infty} h(n)z^{-n},$$

*Inverse Z-transform*

$$h(n) = \mathcal{Z}^{-1}\{H(z)\} = \frac{1}{2\pi i}\oint_C H(z)z^{n-1}dz$$

where $H(z)$ – a transfer function of a discrete system, $z$ – a complex variable, $C$ – a counterclockwise closed path encircling the origin and entirely in the region of convergence (ROC). The complex variable $z^{-n}$, like $e^{-pt}$, is a general form of discrete equation solution. As we can see from the definition, the linearity property for Z-transform is performed, so Z-transform is also a linear operator.

A special case of the contour integral in the Inverse Transform occurs when $C$ is the unit circle. This contour can be used when the ROC includes the unit circle, which is always guaranteed when $H(z)$ is stable, that is, when all the poles are inside the unit circle. With this contour, the Inverse Z-transform simplifies to the Inverse Discrete-Time Fourier Transform, or Fourier series, of the periodic values of the Z-transform around the unit circle:

$$h(n) = \frac{1}{2\pi j}\int_{-\infty}^{+\infty} H\left(e^{j\omega t_s}\right)e^{j\omega t_s(n-1)}d\left(e^{j\omega t_s}\right) = \frac{1}{2\pi j}\int_{-\infty}^{+\infty} H\left(e^{j\omega t_s}\right)e^{j\omega t_s(n-1)}e^{j\omega t_s}jt_s d\omega$$

$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} H\left(e^{j\omega t_s}\right)e^{jn\omega t_s}d(\omega t_s).$$

*1.9.2 Connection with other transforms*

Variable $z$ may be represented as

$$z = e^{pt_s};\ t_s = \frac{1}{f_s}$$

where $f_s$ – sampling frequency of a discrete system. We remember that:

$$p = \sigma + j\omega$$

Then

$$z = e^{pt_s};\ t_s = \frac{1}{f_s}$$

and

$$z = e^{(\sigma+j\omega)t_s} = \overbrace{e^{\sigma t_s}}^{r} \cdot e^{j\omega t_s} = re^{j\omega t_s};\ |z| = r; \arg z = \omega t_s$$

$$H\left(re^{j\omega t_s}\right) = \sum_{n=-\infty}^{+\infty} h(n)\cdot z^{-n} = \sum_{n=-\infty}^{+\infty} h(n)\cdot\left(re^{j\omega t_s}\right)^{-n} = \sum_{n=-\infty}^{+\infty}\left(h(n)r^{-n}\right)\cdot e^{-jn\omega t_s}$$

If $r = 1$ ($\sigma = 0$), then

$$H\left(e^{j\omega t_s}\right) = H'(j\omega) = \sum_{n=-\infty}^{+\infty} h(n)e^{-jn\omega t_s}$$

It is a Discrete-Time Fourier Transform (DTFT). Y-axis (where $\sigma = 0$) from $p$-plane transforms to a unit circle on $z$-plane (figure 1.6). The left half of $p$-plane (where $\sigma < 0$ and $r < 1$) transforms into interior part of the unit circle, the right half of $p$-plane (where $\sigma > 0$ and $r > 1$) – into outer part of the unit circle.

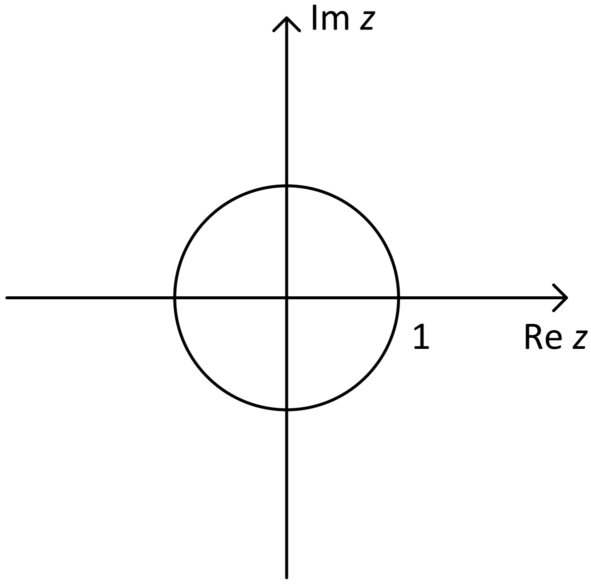



Figure 1.6 – A unit circle on the $z$-plane

Thus, if poles are inside the unit circle on the $z$-plane, then your system is stable; if there is at least one pole outside the unit circle – your system is unstable.

As $\omega t_s$ – an angle on $z$-plane, $z$ is periodic by frequency. Let's find this period:

$$\omega_T t_s = 2\pi \Leftrightarrow 2\pi f_T = \frac{2\pi}{t_s} \Leftrightarrow f_T = \frac{1}{t_s} = f_s$$

Thus, sampling frequency is a period of $z$ and, as consequence, of $H(z)$. Magnitude and phase responses of such a system (discrete-time system) is periodic by frequency with period equaled to sampling frequency.

If we choose interest interval as (-π; π], then:

$$\omega t_s \in (-\pi;\ \pi] \Leftrightarrow 2\pi f \in \left(-\frac{\pi}{t_s};\ \frac{\pi}{t_s}\right] \Leftrightarrow f \in \left(-\frac{f_s}{2};\ \frac{f_s}{2}\right]$$

We may choose any another interval with length $f_s$, but this interval is more convenient for us.

## §1.10 Dirac delta function

Definition

Delta function is a function that satisfy the following statements:

- $\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases};$
- $\int_{-\infty}^{+\infty} \delta(x)dx = 1.$

Properties

*1. Filter property*

$$\int_{-\infty}^{+\infty} f(x)\delta(x)dx = f(0)$$

In general case for any shift $T$:

$$\int_{-\infty}^{+\infty} f(x)\delta(x - T)dx = f(T)$$

*2. Discrete case*

For the discrete case, the definition of the delta function transforms into

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}, n \in \mathbb{Z}$$

3. *Unit and scaled delta function*

Let's take unit step function (figure 1.7). Take a derivation in point 0.

$$1(t) = \begin{cases} 1 \text{ for } t \geq 0 \\ 0 \text{ for } t < 0 \end{cases}$$

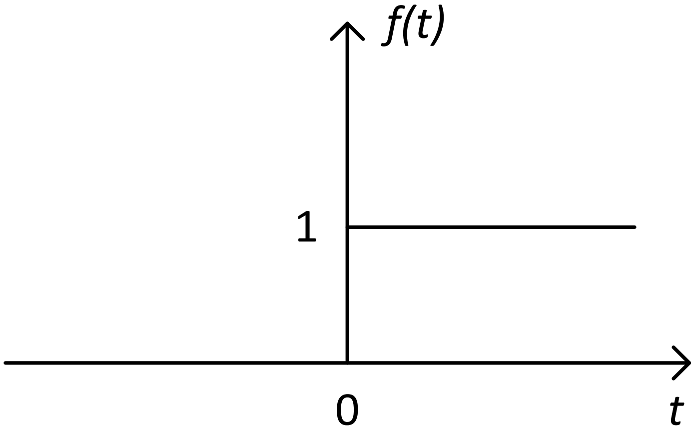



Figure 1.7 – A unit step function *f*(*t*)

A unit step function derivation gives a delta function and: for *t* > 0:

$$\int_{-\infty}^{t} \delta(x)dx = \int_{-\infty}^{+\infty} \delta(x)dx = 1$$

There is an issue with scaled unit step function, it is shown in figure 1.8.

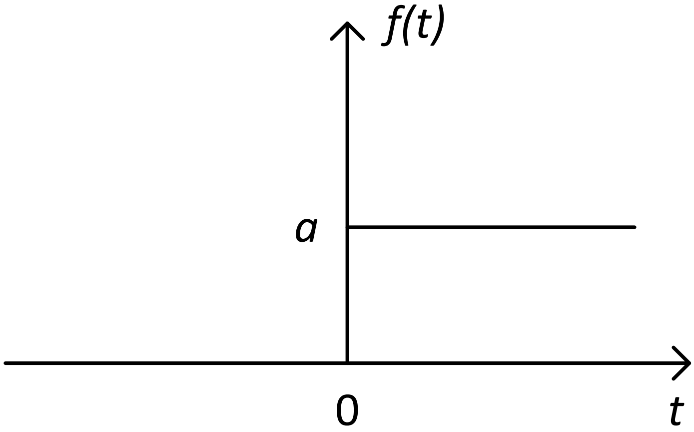



Figure 1.8 – A scaled unit step function *f*(*t*)

Formally, its derivation is the same: just a delta function. But if we take integral, we won't get *a*, we will also get 1. To solve this confusion, it is accepted to write down the such derivation as:

$$\frac{df(t)}{dt} = a\delta(t)$$

Then (for *t* > 0)

$$\int_{-\infty}^{t} a\delta(x)dx = a\int_{-\infty}^{+\infty} \delta(x)dx = a$$

Take into consideration this example, we will assume that in point 0 a delta function is equaled to 1.

4. *Spectrum*

Spectrum *S*(ω) of delta function is:

$$S(\omega) = \int_{-\infty}^{+\infty} \delta(t)e^{-j\omega t}dt = e^{-j\omega \cdot 0} = 1$$

So that it is independent of frequency. Delta function can be present by inverse Fourier Transform as:

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\omega)e^{j\omega t}d\omega = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{j\omega t}d\omega$$

*5. Fourier series*

Let's assume *f*(*t*) that is repeating delta functions with period *T*:

$$f(t) = \sum_{k=-\infty}^{+\infty} \delta(t-kT)$$

As *f*(*t*) is a periodic function, we can expand it in a Fourier series:

$$f(t) = \sum_{k=-\infty}^{+\infty} c_k \cdot e^{j\frac{2\pi}{T}kt}$$

where $c_k$ is:

$$c_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} \delta(t)e^{-j\frac{2\pi}{T}kt}dt = \frac{1}{T}\int_{-\infty}^{+\infty} \delta(t)e^{-j\frac{2\pi}{T}kt}dt = \frac{1}{T}\cdot e^{-j\frac{2\pi}{T}k\cdot 0} = \frac{1}{T}\cdot 1 = \frac{1}{T}$$

Then

$$f(t) = \sum_{k=-\infty}^{+\infty} \frac{1}{T}\cdot e^{j\frac{2\pi}{T}kt} = \frac{1}{T}\sum_{k=-\infty}^{+\infty} e^{j\frac{2\pi}{T}kt}$$

Finally, we got that:

$$\sum_{k=-\infty}^{+\infty} \delta(t-kT) = \frac{1}{T}\sum_{k=-\infty}^{+\infty} e^{j\frac{2\pi}{T}kt}$$

# Chapter 2 Discrete sequences and systems

## §2.1 Introduction

Let's start with understanding what digital signal processing is. A digital code – a code presented by means of digits. Signal processing – a transformation of a signal. Thus, digital signal processing – a transformation of a signal presented with numbers by arithmetic operations.

How can we get a signal presented with numbers? This process is illustrated in Figure 2.1. At first, we have some analog signal $s(t)$, which is continuous by time and by value. It goes through $N$-bit analog-to-digital converter (ADC) with sampling frequency $f_s$. The output of the ADC is a discrete sequence of numbers $x(n)$. The discrete sequence $x(n)$ is discretized signal $s(t)$ by time and by value.

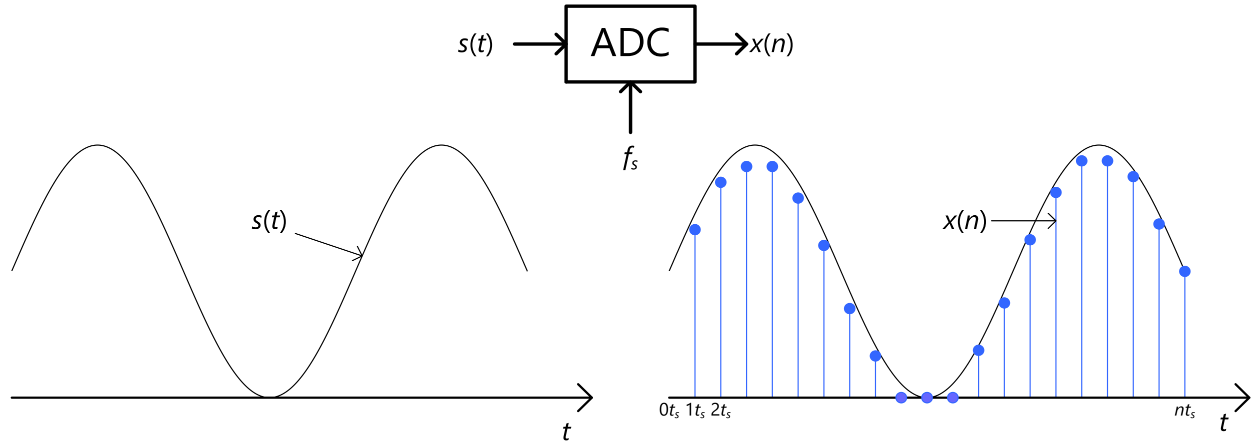



Figure 2.1 – Analog-to-digital conversion of signal $s(t)$ into discrete sequence $x(n)$

Interval between $x(n)$ samples equals sampling period $t_s$. Relation between sampling period and sampling frequency $f_s$ is obvious

$$t_s = \frac{1}{f_s}.$$

Sampling frequency determines time resolution, bits number of the ADC determines value resolution. Hereinafter, we will neglect quantization error and consider that samples exactly equal signal values, i.e

$$x(n) = s(nt_s).$$

Sometimes, you can find such an expression

$$x_{\text{cont}}(t) = s(t) \sum_{k=-\infty}^{+\infty} \delta(t - kt_s)$$

but it is not a sequence of samples ($x_{\text{cont}}$ is a continuous-time function) and it is not the output of the ADC. The output of the ADC is a sequence of index $n$. Such an expression is used to get continuous-time (analog) representation of the sequence $x$. Taking into account that delta function equals zero almost everywhere, we can rewrite the expression above as

$$x_{\text{cont}}(t) = s(t) \sum_{k=-\infty}^{+\infty} \delta(t - kt_s) = \sum_{k=-\infty}^{+\infty} s(kt_s) \cdot \delta(t - kt_s).$$

We got discrete convolution of $s$ and δ. As we know, from the theorem of convolution

$$y(t) = (x * h)(t) \leftrightarrow Y(\omega) = X(\omega) \cdot H(\omega)$$

Spectrum of the delta function is unit-valued frequency independent function.

$$(s * \delta)(t) = S(\omega) \cdot \underbrace{\Delta(\omega)}_{1} = S(\omega)$$

Thus, such a representation $x_{\text{cont}}$ of a discrete sequence does not change the spectrum. In terms of spectrum, $x_{\text{cont}}(t)$ and $x(n)$ are equivalent. In particular, this representation is assumed in Discrete-Time Fourier Transform (§3.2).

## §2.2 Operations on discrete sequences

In digital signal processing, there are only several operations available. By combination of these operations, all transformations are done. These three main operations are illustrated in figure 2.2. They are

- Sequence summation;
- Sequence multiplication;
- Delay unit (memory cell).

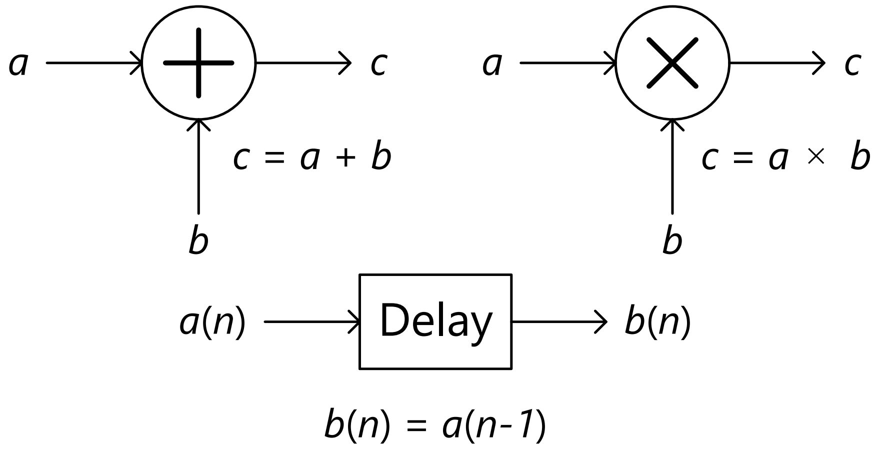



Figure 2.2 – Operations in digital signal processing

In analog processing, we know the next processing blocks: amplifiers, adders, mixers, generators, detectors and filters. In digital signal processing, their functions can also be done, namely:

- Amplifier – a multiplication of sequence by a constant;
- Adder – sequences summation;
- Mixer – sequences multiplication;
- Generator – a source of a determinate sequence;
- Filter – a digital filter;
- Detector – a Hilbert converter (digital filter) with an envelope selection.

All these devices, except digital filter and detector, have obvious realization.

Virtually, any transformation in DSP is done by means of a digital filter. Any combination of summations, multiplications and delays is a structure of some digital filter. That is, a digital filter is not only a frequency selection device, but a transformation device in general.

## §2.3 Unit delay element

Look at a unit delay element (Figure 2.3). We know a relationship between the input and the output for this element:

$$y(n) = x(n-1).$$

Do *z*-transform from the output:

$$Y(z) = \sum_{n=-\infty}^{+\infty} y(n)z^{-n} = \sum_{n=-\infty}^{+\infty} x(n-1)z^{-n} = \sum_{m=-\infty}^{+\infty} x(m)z^{-(m+1)} = z^{-1}\sum_{m=-\infty}^{+\infty} x(m)z^{-m} = z^{-1}X(z)$$

It means that a unit delay element has a transfer function equals to $z^{-1}$. Therefore, both representations depicted in Figure 2.3 are equivalent.

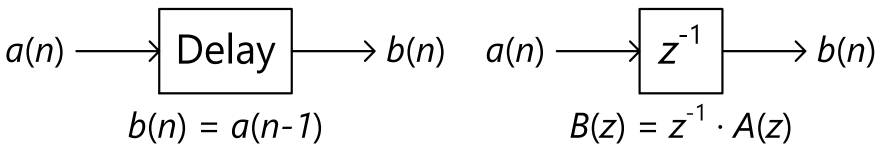


Figure 2.3 – Two ways of representation for a unit delay element

## §2.4 Systems in digital signal processing

*2.4.1 Discrete linear systems*

<u>Definition</u>

*Discrete linear system (DLS) – a discrete system that is a linear operator for an input-output transform.*

For example, The Fourier Transform is a linear operator. So that, a device performing Fourier Transform with discrete sequences is a discrete linear system. But systems that perform such functions:

$$y(n) = |x(n)|;\ y(n) = x^2(n);\ y(n) = \sin x(n)$$

are nonlinear, and linearity property is not executed. The output signal of such systems will have a distortion, that is spurious spectral components.

*2.4.2 Time-invariant systems*

Except linear systems, we are also interested in another independent class of systems – time-invariant systems. Time-invariance means that a time shift of the input signal cause the similar shift of the output signal. If input sequence *x* causes output sequence *y*, i. e.

$$x(n) \rightarrow y(n),$$

then we can write the time-invariance property as

$$x(n+k) \rightarrow y(n+k).$$

And for time-invariance systems, it is true for every *k*. In other words, a time-invariance system is invariant for a time reference point. As an example, look at the system, where

$$y(n) = \alpha x(n).$$

The input and the output test signals are presented in Figure 2.4. Shifting the input signal *x*(*n*) by a quarter period similarly changes shifting in time of the output sequence *y*(*n*).

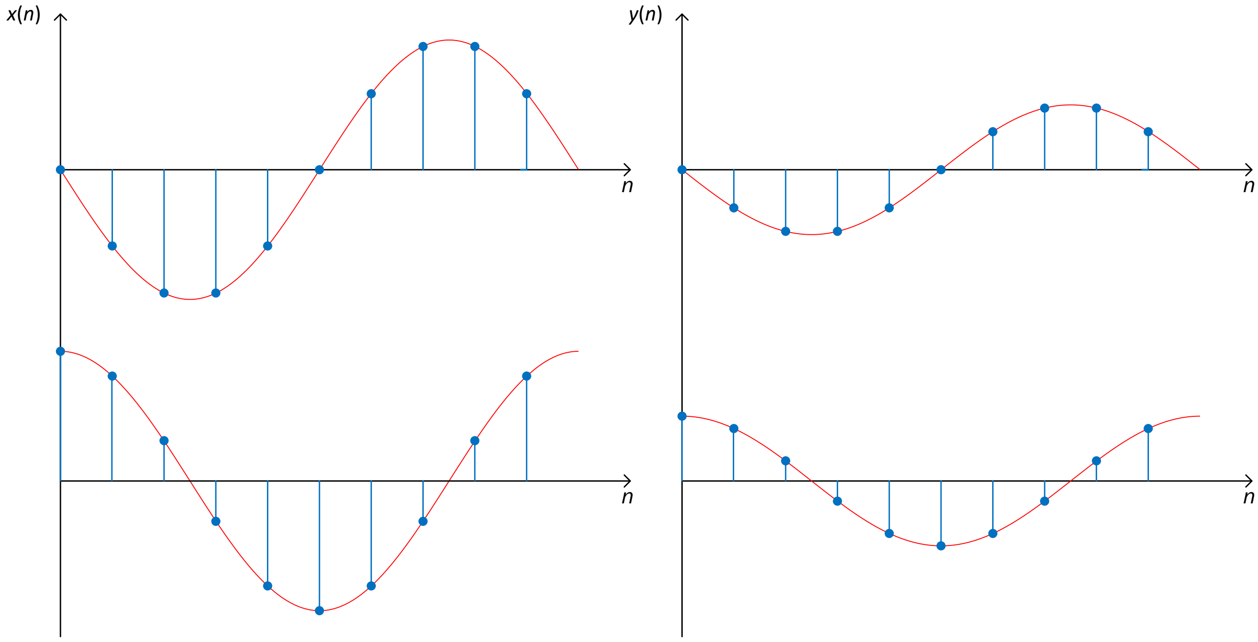



Figure 2.4 – A system response before (top) and after (bottom) time shift

It should be noticed that for time-variant systems an impulse response cannot be unambiguously defined. This is due to dependence of the output response from the initial time shift. From this point, time-variant systems are similar with nonlinear systems, where the output response depends on an absolute value of the input.

*2.4.3 Discrete Linear Time-Invariant (LTI) systems*

Combination of two mentioned independent properties (linearity and time-invariance) gives us a very convenient class of systems – discrete Linear Time-Invariant (LTI) systems.

Properties of discrete time-invariant linear systems

1. *Linearity (superposition)*

Assume that every input sequence has corresponding output sequence:

$$x_k(n) \to y_k(n)$$

Then a linear combination of the input sequences corresponds to the same linear combination of the output sequences:

$$x(n) = \sum a_k x_k(n) \to y(n) = \sum a_k y_k(n)$$

2. *Convolution with impulse response*

For LTI systems, impulse response is extremely important. If we know the impulse response, we can say that we know everything about our system behavior. In other words, it is possible to know the output signal for every input signal, because the output signal can be calculated as a convolution of the impulse response and the input signal. For LTI, it will be:

$$y(n) = \sum_{k=0}^{K-1} h(k) \cdot x(n-k),$$

where *y*(*n*) – the output sequence, *x*(*n*) – the input sequence, *h*(*k*) – the impulse response, *K* – the length of the impulse response. This relation is possible due to the properties of LTI system. In nonlinear systems response depends on an absolute value of the input, in time-variant systems it depends on time when sequence starts. In LTI systems both dependencies are absent, impulse response is unambiguity and can be applied with any input signal.

Let's proof that combination of linearity and time-invariance allow us to use convolution for getting output response. Let our system be an operator *A* that transforms space of *u* to space of *t*, i.e.

$$y(t) = A_t\{x(u); u\}.$$

Due to linearity the following statement is true

$$A_t\left\{\int_{-\infty}^{+\infty} c_\tau \cdot x(u) d\tau\,; u\right\} = \int_{-\infty}^{+\infty} c_\tau \cdot A_t\{x(u); u\} d\tau$$

Due to time-invariance the following statement is true

$$y(t-\tau) = A_{t-\tau}\{x(u); u\} = A_t\{x(u-\tau); u\}$$

Impulse response in this notation will be

$$h(t) = A_t\{\delta(u); u\}$$

Thus, the convolution is

$$x * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau = \int_{-\infty}^{+\infty} x(\tau) \cdot A_{t-\tau}\{\delta(u); u\} d\tau = \int_{-\infty}^{+\infty} x(\tau) \cdot A_t\{\delta(u-\tau); u\} d\tau$$

$$= A_t\left\{\int_{-\infty}^{+\infty} x(\tau) \cdot \delta(u-\tau) d\tau\,; u\right\} = A_t\{x(u); u\} = y(t)$$

3. *Commutativity*

For LTI systems, it is possible to change their order without affecting the output sequence (Figure 2.5). This is a consequence of the previous property. Let's prove it. Consider

$$x * h_1(t) = y_1; x * h_2(t) = y'_1; y_1 * h_2(t) = y_2$$

then

$$y_2(t) = y_1 * h_2(t) = \int_{-\infty}^{+\infty} h_2(\tau) \cdot y_1(t-\tau) d\tau = \int_{-\infty}^{+\infty} h_2(\tau) \cdot \left( \int_{-\infty}^{+\infty} h_1(p) \cdot x(t-\tau-p) dp \right) d\tau$$

$$= \int_{-\infty}^{+\infty} \left( \int_{-\infty}^{+\infty} h_2(\tau) \cdot h_1(p) \cdot x(t-\tau-p) dp \right) d\tau = \int_{-\infty}^{+\infty} h_1(p) \cdot \left( \int_{-\infty}^{+\infty} h_2(\tau) \cdot x(t-p-\tau)\, d\tau \right) dp$$

$$= \int_{-\infty}^{+\infty} h_1(p) \cdot y'_1(t-p) dp = y'_1 * h_1(t).$$

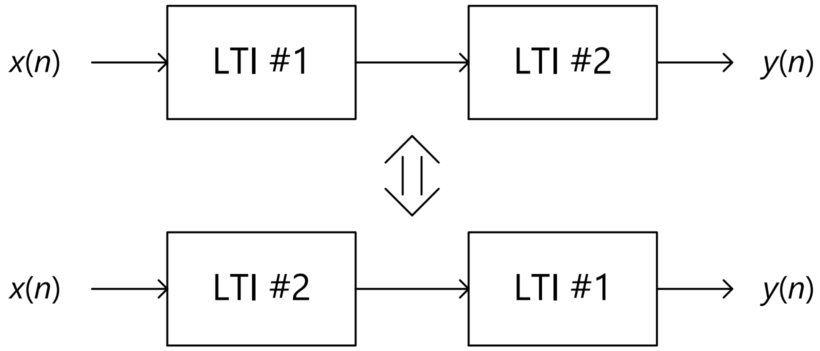



Figure 2.5 – Commutativity property

## §2.5 Real-time systems

Real-time system – a system that guarantees a response for each input sample at a fixed time interval. A real-time system may have a delay for an input action, but this delay is guaranteed and fixed. A typical real-time system has equal rate of input and output data streams, and each new input sample changes an output sample. An exception is systems for sample rate conversion. Systems that imply block processing (like microprocessors or CPU) are not inherently real-time systems. However, they can function as real-time systems if they are capable of processing a data block at least *N* times faster than input rate (*N* – a block size). For example, CPUs can process audio signals in real-time due to high performance in comparison with audio stream rate.

## §2.6 Complexity metrics

In comparison with digital systems, typically, we are interested in performance, power consumption and hardware costs (or area). From all the mentioned operations (summation, multiplication and delay), multiplication block is the most complex one. It requires plenty of logic gates, which results in more area, power consumption and time for processing signal. Thus, we will evaluate the complexity and performance of digital systems by multipliers number.

# Chapter 3 Sampling signals

## §3.1 Ambiguity of signal presentation

Let's take a discrete sequence *x*(*n*) that is illustrated in Figure 3.1 and try to restore a harmonic signal *x*(*t*). One person may see here harmonic signal with one period, another – with 3 periods. These situations are shown in Figure 3.2. And both variants are absolutely right without any additional information about the signal *x*(*t*).

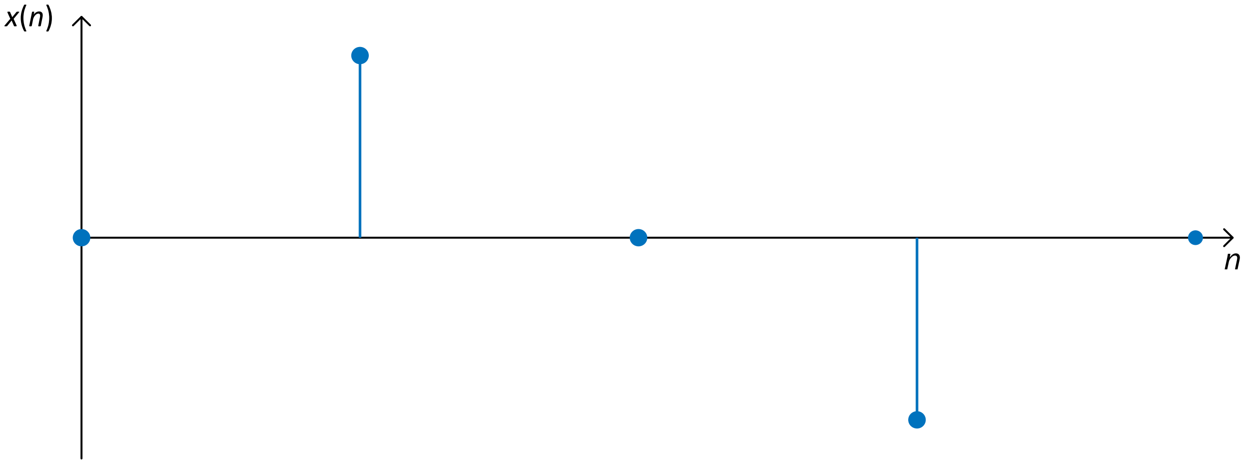



Figure 3.1 – A discrete sequence *x*(*n*)

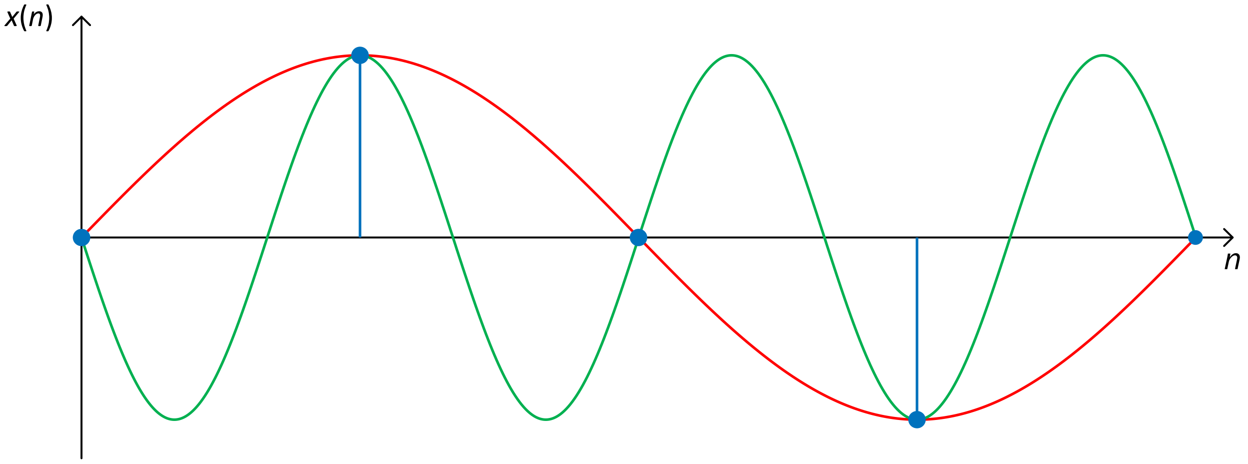



Figure 3.2 – Two different signals correspond to the sequence *x*(*n*)

It means that a discrete sequence without an additional information may represent an infinite number of frequencies. Let's determine this frequency family. At first, take a discrete sequence as

$$x(n) = \sin(2\pi f_0 n t_s)$$

Add up 2π*m* (*m*∈ℤ) to the phase. It doesn't change the sequence, so

$$x(n) = \sin(2\pi f_0 n t_s) = \sin(2\pi f_0 n t_s + 2\pi m) = \sin\big(2\pi(f_0 n t_s + m)\big) = \sin\big(2\pi(f_0 n t_s + m f_s t_s)\big)$$
$$= \sin\left(2\pi\left(f_0 + \frac{m}{n} f_s\right) n t_s\right).$$

From here, we see that each sample of *x*(*n*) corresponds to the family with frequencies

$$f_0 + \frac{m}{n} f_s.$$

To determine the family for the whole sequence, we need to exclude a family dependence of the sequence index *n*. For this, we choose *m* = *kn*. Then

$$x(n) = \sin(2\pi(f_0 + k f_s) n t_s)$$

Thus, the discrete sequence *x*(*n*) corresponds to signals with frequencies

$$f_0 + kf_s.$$

Such a family is demonstrated in Figure 3.3. This correspondence between the discrete sequence and the frequency family means that a spectrum of a discrete sequence is periodic. A spectrum repetition period is $f_s$. The images corresponded to $k \neq 0$ are called aliases.

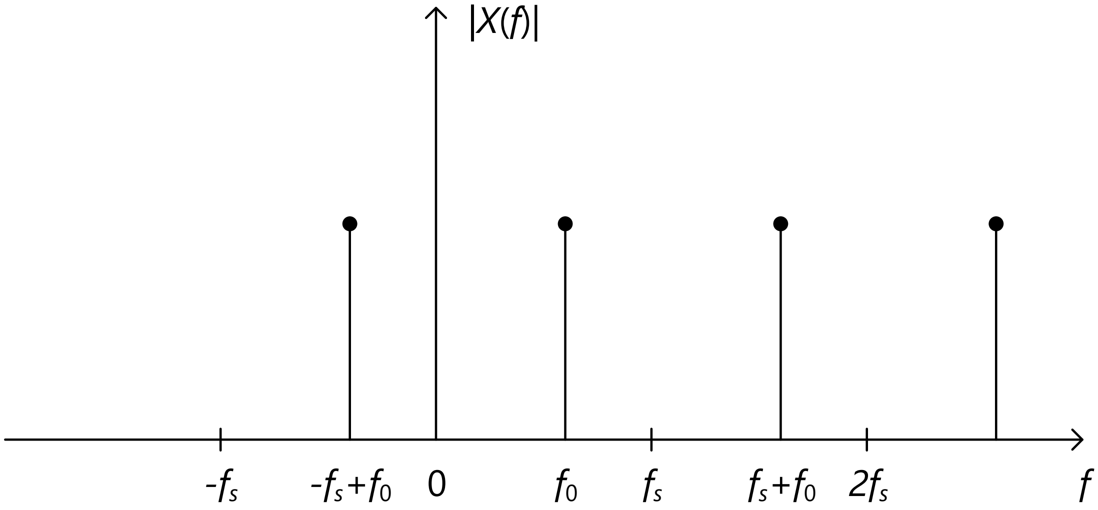



Figure 3.3 – A frequencies family represented by the sequence $x(n)$

## §3.2 Discrete-Time Fourier transform

We already know several instruments for getting a spectrum, namely: The Fourier Series and Integral Fourier Transform (IFT). But they have a requirement – function must be continuous in time. As a result, we cannot use them for discrete sequences. Therefore, we need other instrument that is capable to operate with discrete sequences. Such an instrument is called the Discrete-Time Fourier Transform (DTFT).

| *Direct Discrete-Time Fourier Transform* | *Inverse Discrete-Time Fourier Transform* |
|---|---|
| $X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) \cdot e^{-j\omega n t_s}$ | $x(n) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} X(\omega) \cdot e^{j\omega n t_s} d\omega$ |

This transform can be obtained from the Integral Fourier Transform. The IFT requires a continuous-time function, so we use the continuous-time representation of a discrete sequence from §2.1.

$$x_{\text{cont}}(t) = s(t) \sum_{k=-\infty}^{+\infty} \delta(t - kt_s) = \sum_{k=-\infty}^{+\infty} s(kt_s) \cdot \delta(t - kt_s) = \sum_{k=-\infty}^{+\infty} x(k) \cdot \delta(t - kt_s)$$

If we apply the IFT to $x_{\text{cont}}$, we will get

$$X(\omega) = \int_{-\infty}^{+\infty} x_{\text{cont}}(t) \cdot e^{-j\omega t} dt = \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} x(k) \cdot \delta(t - kt_s) \cdot e^{-j\omega t} dt = \sum_{k=-\infty}^{+\infty} x(k) \int_{-\infty}^{+\infty} \delta(t - kt_s) \cdot e^{-j\omega t} dt$$

$$= \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-j\omega k t_s} = \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-j\omega k t_s}.$$

Thus, we get the Direct DTFT expression. Note, a spectrum of DTFT is periodic with period $\omega_s$, i.e.

$$X(\omega + k\omega_s) = X(\omega), k \epsilon \mathbb{Z}.$$

▲ Home exercise: check that spectrum is periodic.

The spectrum of the DTFT will be discussed in detail in the next section.

## §3.3 Discrete sequence spectrum

From the previous sections, we have learned a relation between a continuous signal $s(t)$ and its discrete sequence $x(n)$ and known that $X(\omega)$ is periodic, i.e.

$$s(nt_s) \to x(n);\ X(\omega + k\omega_s) = X(\omega), k\epsilon\mathbb{Z}.$$

But still there is a question. What is relation between spectrums $S(\omega)$ and $X(\omega)$?

$$S(\omega) \overset{?}{\to} X(\omega)$$

To know it, let's calculate $X(\omega)$ using DTFT.

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) \cdot e^{-j\omega n t_s} = \sum_{n=-\infty}^{+\infty} s(nt_s) \cdot e^{-j\omega n t_s}$$

Present $s(t)$ using its Inverse Integral Fourier Transform (see §1.7.1)

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot e^{j\Omega t} d\Omega \to s(nt_s) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot e^{j\Omega n t_s} d\Omega$$

where $S(\Omega)$ – the original signal spectrum. Then

$$\sum_{n=-\infty}^{+\infty} s(nt_s) \cdot e^{-j\omega n t_s} = \sum_{n=-\infty}^{+\infty} \left( \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot e^{j\Omega n t_s} d\Omega \right) \cdot e^{-j\omega n t_s} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot \sum_{n=-\infty}^{+\infty} e^{j(\Omega-\omega) n t_s}\, d\Omega$$

From §1.10 we know that

$$\sum_{k=-\infty}^{+\infty} \delta(t - kT) = \frac{1}{T}\sum_{k=-\infty}^{+\infty} e^{j\frac{2\pi}{T}kt}$$

So summation of exponents can be replaced by

$$\sum_{k=-\infty}^{+\infty} e^{j(\Omega-\omega)kt_s} = \omega_s \sum_{k=-\infty}^{+\infty} \delta(\Omega - \omega - k\omega_s)$$

where

$$t \leftrightarrow \Omega - \omega; k \leftrightarrow k; \frac{2\pi}{T} \leftrightarrow t_s;$$

$$T \leftrightarrow \frac{2\pi}{t_s} = 2\pi f_s = \omega_s$$

Using this, we get

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot \sum_{n=-\infty}^{+\infty} e^{j(\Omega-\omega)nt_s}\, d\Omega = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot \omega_s \sum_{n=-\infty}^{+\infty} \delta(\Omega - \omega - n\omega_s)\, d\Omega$$

$$= \frac{\omega_s}{2\pi}\int_{-\infty}^{+\infty} S(\Omega) \cdot \sum_{n=-\infty}^{+\infty} \delta(\Omega - \omega - n\omega_s)\, d\Omega = \frac{\omega_s}{2\pi}\sum_{n=-\infty}^{+\infty}\int_{-\infty}^{+\infty} S(\Omega) \cdot \delta(\Omega - \omega - n\omega_s) d\Omega$$

$$= \frac{\omega_s}{2\pi}\sum_{k=-\infty}^{+\infty} S(\omega + n\omega_s)$$

Finally, the result is

$$X(\omega) = \frac{\omega_s}{2\pi}\sum_{n=-\infty}^{+\infty} S(\omega + n\omega_s)\, , n \in \mathbb{Z}$$

It means that spectrum of a discrete sequence is a summation of periodically repeating original signal spectrum. This effect is illustrated in Figure 3.4.

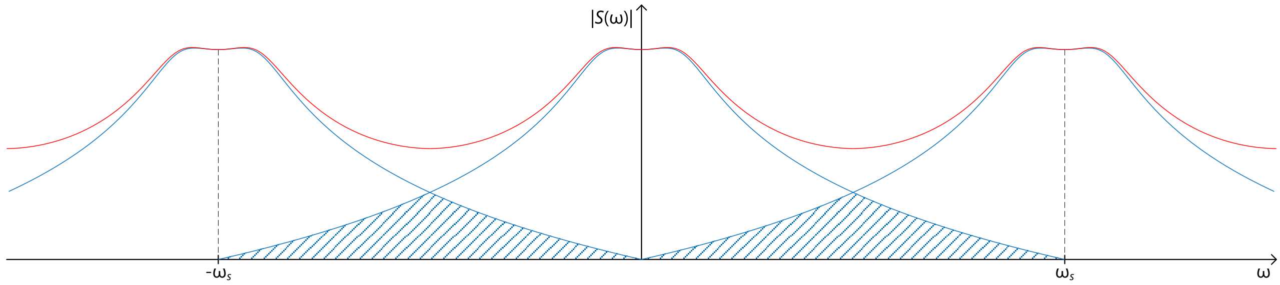



Figure 3.4 – The result of a spectrum summation after sampling

## §3.4 Signal reconstruction

We have discussed aspects concerning transition from continuous-time signals to discrete sequences. What about inverse transitions? That is, we want to reconstruct a continuous-time signal $r(t)$ from a discrete sequence $x(n)$. How can we do this? Will $r(t)$ be equal $s(t)$?

To make a reconstruction, we need some reconstruction function $h(t)$. Virtually, we have already done a reconstruction by means of the Dirac delta function for $x_{cont}(t)$ in §3.2. In general case, a reconstructed signal $r(t)$ will be a function defined as

$$r(t) = \sum_{k=-\infty}^{+\infty} x(k) \cdot h(t - kt_s) = (x * h)(t).$$

Thus, $h(t)$ can be interpreted as an impulse response of a reconstruction device. What about spectrum? By the convolution theorem, we get that

$$R(\omega) = X(\omega) \cdot H(\omega).$$

Take as an example a digital-to-analog converter. Typically, it has a rectangular reconstruction function. Example of corresponding convolution is illustrated in Figure 3.5.

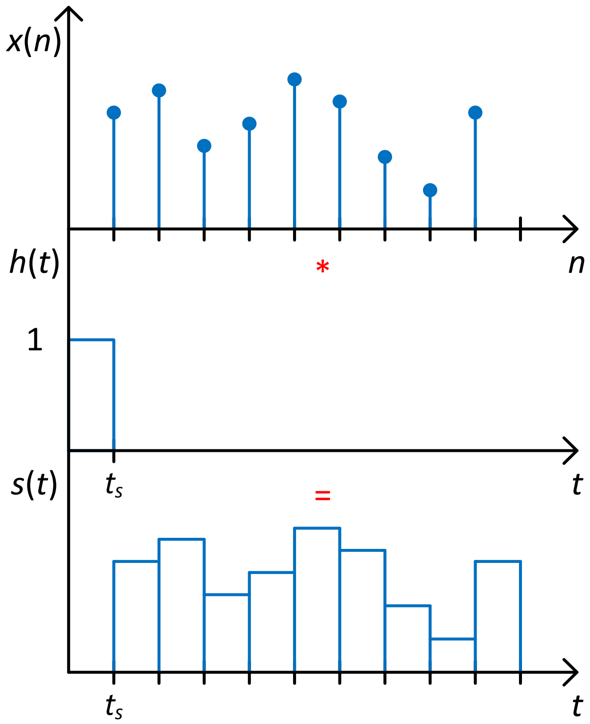



Figure 3.5 – Reconstruction function an ideal DAC

▲ Home exercise: get spectrum of the ideal DAC.

## §3.5 Sampling low-pass signals

In this paragraph we talk about low-pass signals, i.e. signals having the most their energy around 0 frequency. Consider frequency range of interest from $-f_s/2$ to $f_s/2$, which is known as “baseband”. In general case, we can consider any frequency range having width $f_s$. Look at a spectrum of a low-pass signal with bandwidth *B* (Figure 3.6). This spectrum is symmetric because of the properties of a real signal.

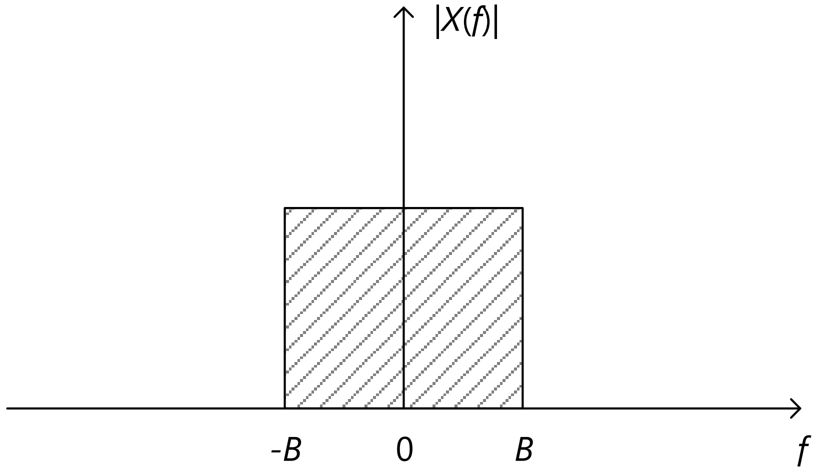


Figure 3.6 – A spectrum of a low-pass real signal

If we sample this signal with a sampling frequency $f_s$, we will get a spectrum illustrated in Figure 3.7. The spectrum is periodic with period $f_s$. From here, we can understand requirements for the sampling frequency. In shown example, a repetition of the spectrum doesn't distort our signal. But if we take the sampling frequency smaller as demonstrated in Figure 3.8, we will get a distortion for our signal because the spectrum copies (or "aliases") overlap one another. Such an effect is called aliasing.

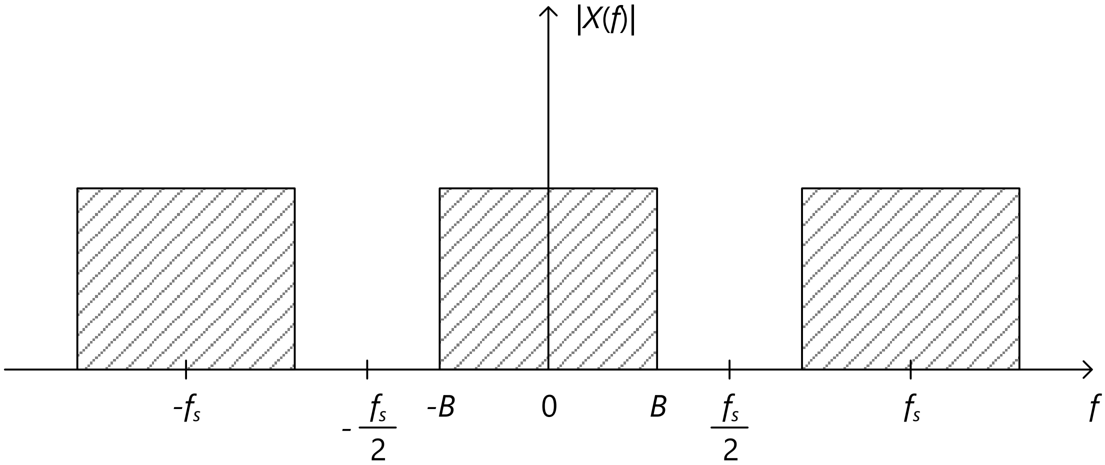


Figure 3.7 – A spectrum after sampling with $f_s$ frequency

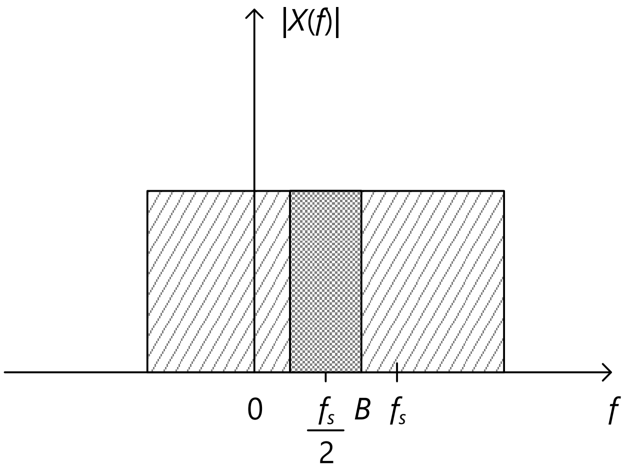


Figure 3.8 – Aliasing of spectrums

From here we see that for distortion absence, we need that:

$$\frac{f_s}{2} > B \Leftrightarrow f_s > 2B$$

This condition is called Nyquist criteria. The fulfilment of this condition provides us a non-distorted signal. To choose sampling frequency more than doubled signal bandwidth is not enough to exclude aliasing. If our signal has noise like in Figure 3.9, then after sampling we still will have a distortion that is illustrated in figure 3.10.

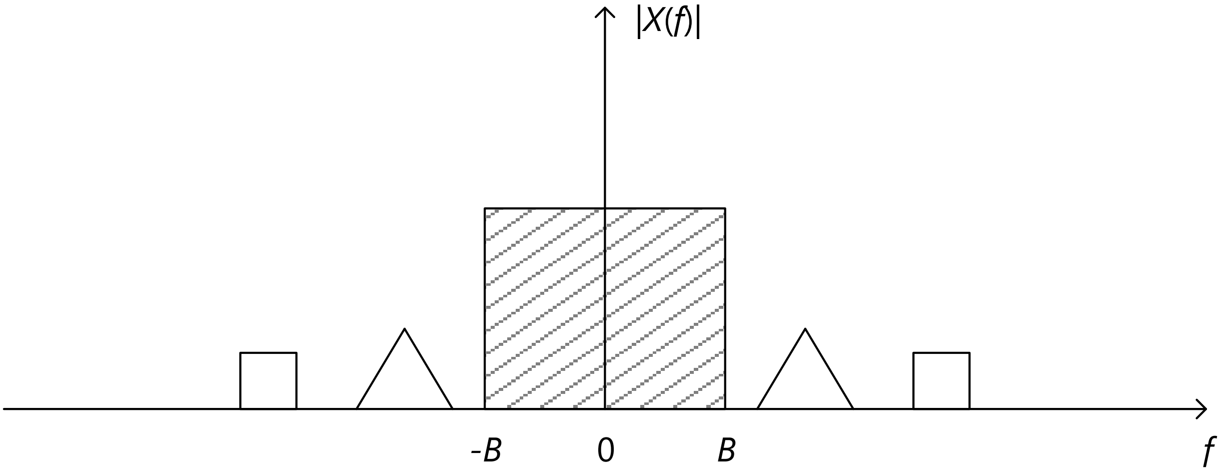



Figure 3.9 – Spectrum of the input signal with noise

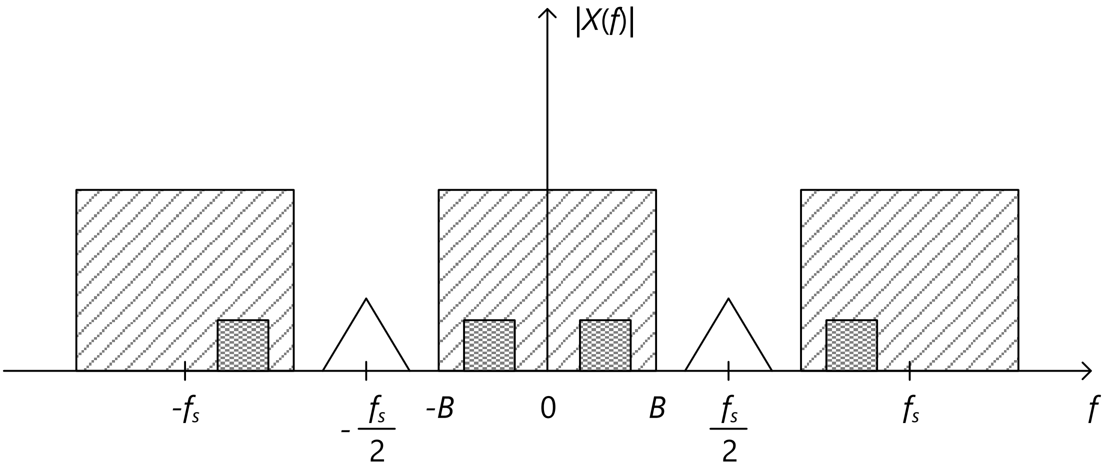



Figure 3.10 – Spectrum of the input signal with noise after sampling

The resulting spectrum will be a sum of repeating spectrums (see §3.3). To avoid this, it is necessary to put a low-pass filter before an ADC. Such a diagram is presented in Figure 3.11. The low-pass filter has to cut-off all out of a signal bandwidth (Figure 3.12). The low-pass filter before the ADC is called antialiasing filter.

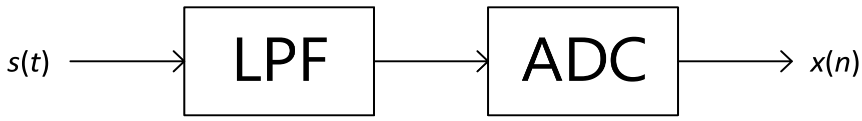



Figure 3.11 – Using a low-pass filter for antialiasing

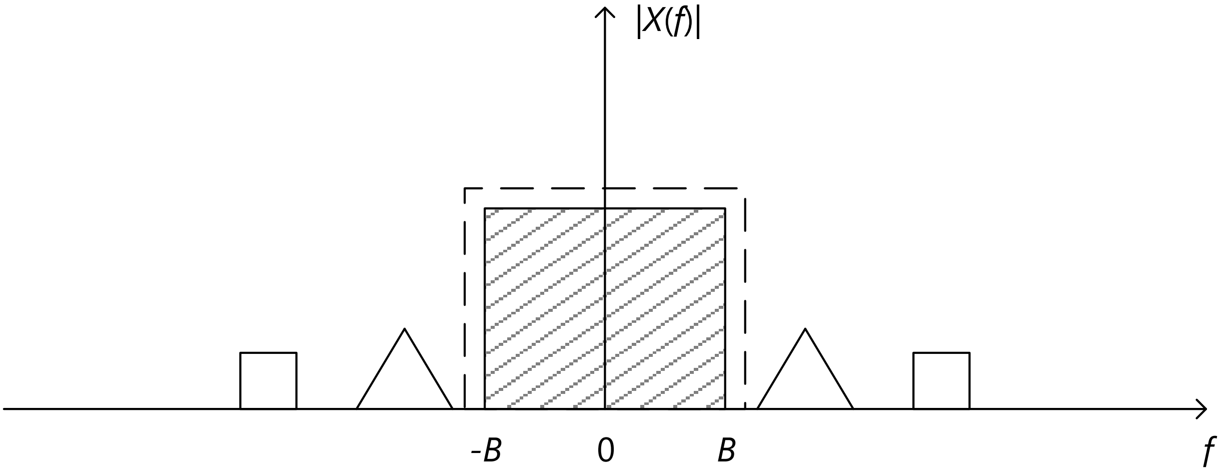



Figure 3.12 – Required magnitude response of the low-pass filter.

## §3.6 Sampling band-pass signals

### *3.6.1 Limits for a band-pass sampling*

Now we discuss the sampling of a band-pass signal. Consider a band-pass signal with carrier $f_0$ and band $B$. A spectrum of such a signal is depicted in Figure 3.13. Using the Nyquist criteria, the sampling frequency for such signal should be

$$f_s > 2f_{max} = 2\left(f_0 + \frac{B}{2}\right) = 2f_0 + B$$

But such sampling frequency may be very high. Moreover, we have an empty band from $-f_0+B/2$ to $f_0-B/2$. Let's employ aliasing of spectrum to our advantage.

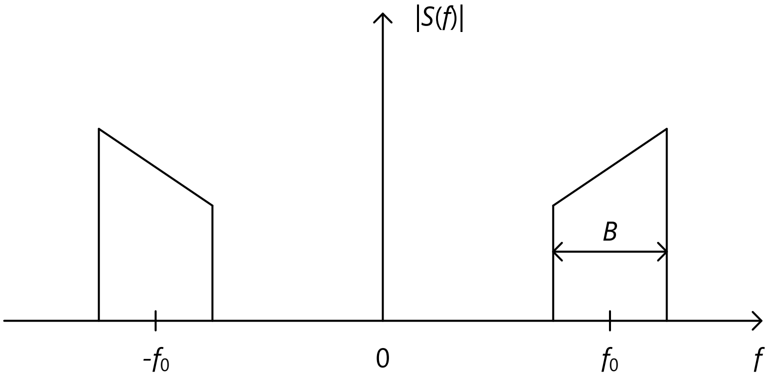



Figure 3.13 – A spectrum of a pass-band signal

Choose sampling as the lowest frequency in spectrum, i.e.

$$f_s = f_0 - \frac{B}{2}$$

Then the spectrum after the sampling will be as illustrated in Figure 3.14. Blue spectrum – a spectrum of the original signal; green spectrum – a spectrum of the alias in the main band; grey spectrums – spectrums of other aliases.

Let's calculate possible sampling frequencies for such case. We know that the empty band has the width

$$f_0 - \frac{B}{2} - \left(-f_0 + \frac{B}{2}\right) = 2f_0 - B.$$

In this band, we may have only integer number $m$ of spectrum copies. A width of each copy is $f_s$. So

$$mf_s = 2f_0 - B \Leftrightarrow f_s = \frac{2f_0 - B}{m}.$$

This is an upper limit for the sampling frequency. And the situation, depicted in Figure 3.14, corresponds to $m$ = 2, i. e.

$$f_s = \frac{2f_0 - B}{2} = f_0 - \frac{B}{2}$$

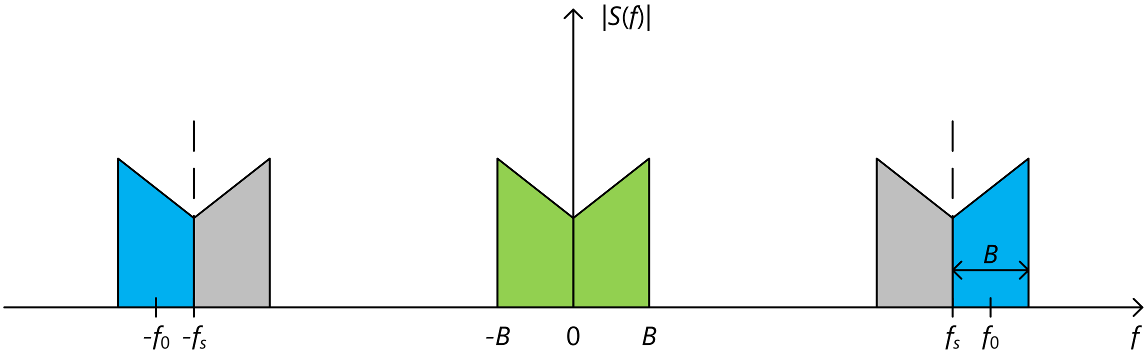



Figure 3.14 – A spectrum of the signal for $f_s$ = $f_0$-$B$/2 ($m$ = 2)

Can we increase $f_s$? No, we will have distortions. Can we decrease $f_s$? Yes. An example of such a situation is illustrated in Fig. 3.15. Here we see that there is no possibility to decrease $f_s$ further. That is, it is a lower limit for the sampling frequency.

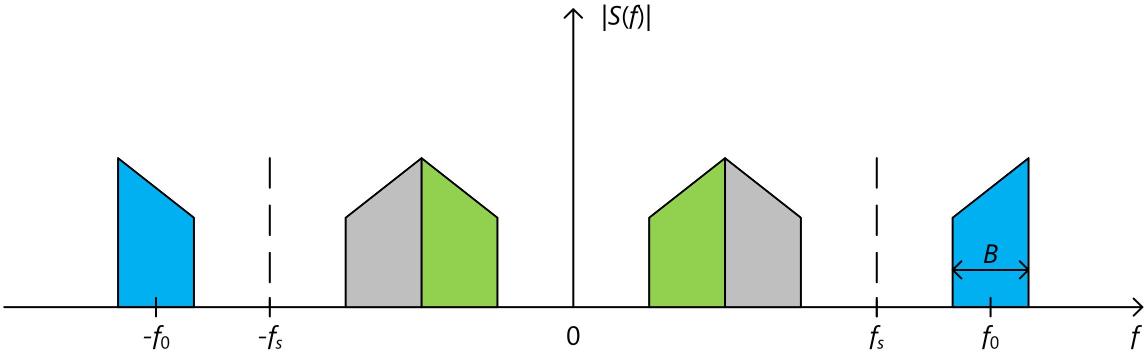



Figure 3.15 – A spectrum of the signal for lower limit of $f_s$ ($m$ = 2)

Let's estimate the lower limit. We still have the same number of copies, but they present in the narrower band. The band is decreased by value 2$x$, i. e.

$$mf_s = 2f_0 - B - 2x$$

Value $x$ is shown in Figure 3.16 and can be calculated, for example, using a copy of the spectrum in the band from 0 to $f_s$ as

$$2x = f_s - 2B.$$

Thus,

$$mf_s = 2f_0 - B - (f_s - 2B) \Leftrightarrow (m+1)f_s = 2f_0 + B \Leftrightarrow f_s = \frac{2f_0 + B}{m+1}$$

And

$$\frac{2f_0 + B}{m+1} \le f_s \le \frac{2f_0 - B}{m}$$

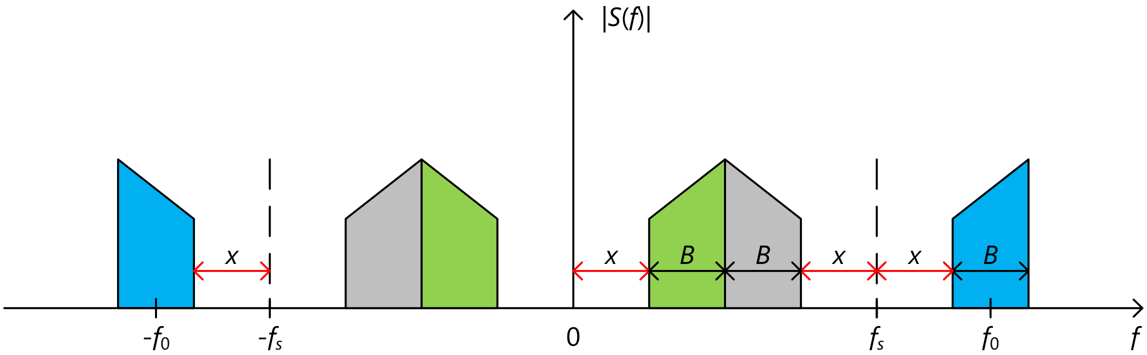



Figure 3.16 – Calculation of $x$ value

It is recommended not to choose the sampling frequency exactly at the upper/lower limit because frequency at edges will be distorted by the aliasing. In other words, requirements for the band-pass sampling is

$$\frac{2f_0 + B}{m+1} < f_s < \frac{2f_0 - B}{m}$$

Moreover, it can be noticed that for $m = 0$, we obtain our first estimation for the sampling frequency, namely

$$\frac{2f_0 + B}{0+1} < f_s < \frac{2f_0 - B}{0} \Leftrightarrow 2f_0 + B < f_s < \infty \Leftrightarrow f_s > 2f_0 + B$$

That is, low-pass sampling is included into conditions for the band-pass sampling.

What is the lowest sampling frequency? Here we need to remind Nyquist criteria and restrict lower limit by the doubled band of the signal, that is

$$\frac{2f_0 + B}{m+1} > 2B \Leftrightarrow \frac{2f_0 + B}{2B} > m + 1 \Leftrightarrow m < \frac{2f_0 + B}{2B} - 1 = \frac{2f_0 - B}{2B} = \frac{f_0}{B} - \frac{1}{2}$$

So, we obtain an upper limit for *m* as

$$m < \frac{f_0}{B} - \frac{1}{2}$$

*3.6.2 Spectrum inversion*

Let's have a look at a spectrum for $m = 3$; it is shown in Figures 3.17 and 3.18 (upper and lower limit cases respectively). The spectrum for lower limit in the main band is flipped. This effect is called "an inversion of a spectrum" and illustrated additionally in Figure 3.17. You can detect the inversion by an orientation of spectrums that are closest to the 0 Hz. The inversion happens only for **odd *m*** and is important only for signals with an asymmetric band.

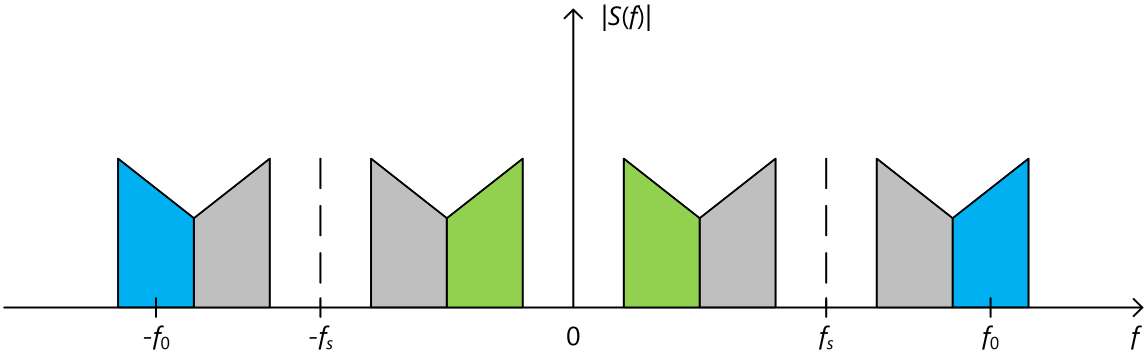



Figure 3.17 – A spectrum for $m = 3$ and the upper limit

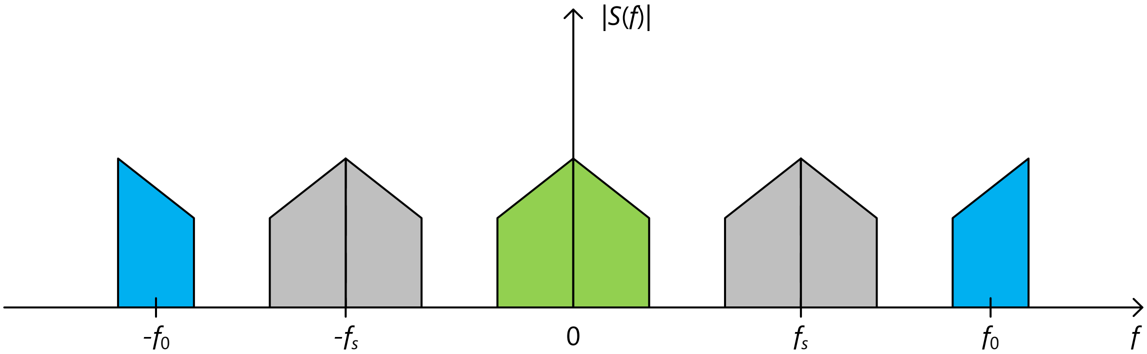



Figure 3.18 – A spectrum for $m = 3$ and the lower limit

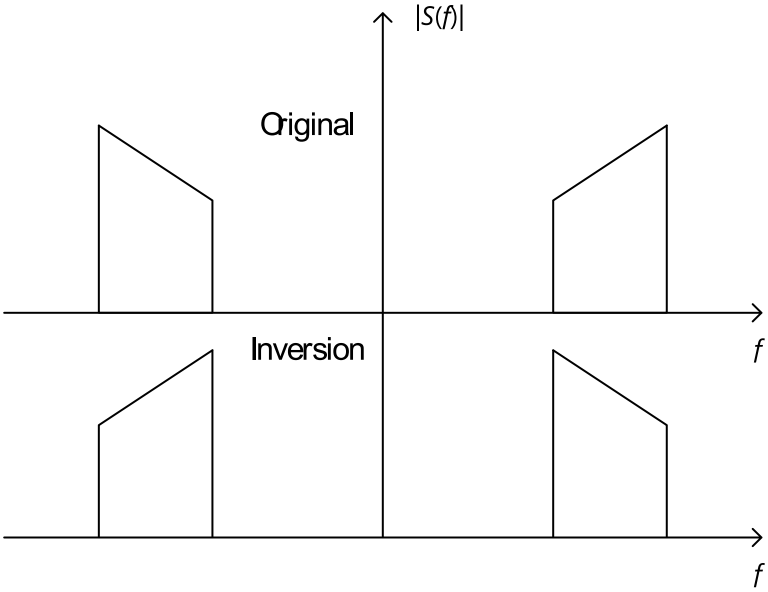



Figure 3.19 – Illustration of an inversion of a spectrum

If you've got an undesirable inversion, it is not a critical problem. The inversion can be cured by the following operation

$$x'(n) = x(n) \cdot (-1)^n$$

This operation is equivalent to a shift of the spectrum by $f_s/2$. Why? Let's show this. The second term of the multiplication can be presented as

$$\cos\left(2\pi \frac{f_s}{2} n t_s\right) = \cos(\pi n) = (-1)^n$$

For example, if

$$x(n) = \cos(2\pi f n t_s)$$

Then

$$x(n) \cdot (-1)^n = \cos(2\pi f n t_s) \cdot \cos\left(2\pi \frac{f_s}{2} n t_s\right) = \frac{\cos\left(2\pi f n t_s + 2\pi \frac{f_s}{2} n t_s\right) + \cos\left(2\pi f n t_s - 2\pi \frac{f_s}{2} n t_s\right)}{2}$$

$$= \frac{\cos\left(2\pi \left(f + \frac{f_s}{2}\right) n t_s\right) + \cos\left(2\pi \left(f - \frac{f_s}{2}\right) n t_s\right)}{2}$$

After some evaluations

$$\cos\left(2\pi \left(f - \frac{f_s}{2}\right) n t_s\right) = \cos\left(2\pi \left(f - \frac{f_s}{2}\right) n t_s + 2\pi n\right) = \cos\left(2\pi \left(f - \frac{f_s}{2}\right) n t_s + 2\pi f_s n t_s\right)$$

$$= \cos\left(2\pi \left(f - \frac{f_s}{2} + f_s\right) n t_s\right) = \cos\left(2\pi \left(f + \frac{f_s}{2}\right) n t_s\right)$$

Thus,

$$x(n) \cdot (-1)^n = \frac{\cos\left(2\pi \left(f + \frac{f_s}{2}\right) n t_s\right) + \cos\left(2\pi \left(f - \frac{f_s}{2}\right) n t_s\right)}{2} = \frac{2\cos\left(2\pi \left(f + \frac{f_s}{2}\right) n t_s\right)}{2} = \cos\left(2\pi \left(f + \frac{f_s}{2}\right) n t_s\right)$$

After applying the mentioned operation, the modified spectrum will be as in Figure 3.20. There is no inversion and the spectrum in the main band locates at 0 frequency.

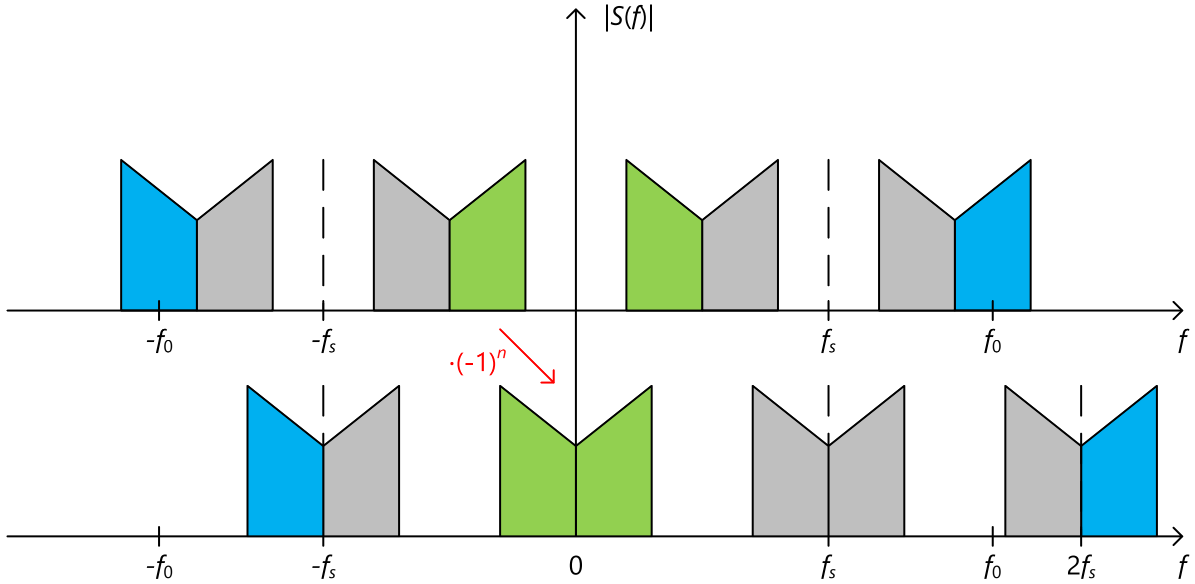



Figure 3.20 – The cure of the spectrum inversion

### *3.6.3 Recommendations*

Situations, in which the spectrums are in contact at $\pm f_s/2$ frequency (as in Figures 3.15 and 3.17), are undesirable for the following processing (since your signal is still band-pass and, as consequence, your band for processing is wider than it is necessary). So, to get an asymmetric spectrum at 0 frequency without the inversion, there are 2 options:

1. Use the upper limit for even *m*;
2. Use the upper limit for odd *m* and then multiply the signal by $(-1)^n$;

For a symmetric spectrum (the inversion doesn't matter), there are 2 additional options:

1. Use the lower limit for odd *m*.
2. Use the lower limit for even *m* and then multiply the signal by $(-1)^n$;

Moreover, some other approaches exist for choosing the sampling frequency.

1. Take the average value of lower and upper limits, then $f_s$ is expressed by
$$f_s = \frac{1}{2}\left(\frac{2f_0 + B}{m+1} + \frac{2f_0 - B}{m}\right)$$
2. Take the sampling frequency as
$$f_s = \frac{4f_0}{m}, m - \text{odd}$$

Then spectrum will be centered at $f_s/4$ frequency.

▲ Home exercise: prove the statement above.

# Chapter 4 Discrete Fourier Transform

## §4.1 Derivation of the formula

In digital signal processing, we deal with finite time discrete sequences. So we can't use the Discrete Time Fourier Transform (DTFT) or even more the Integral Fourier Transform (IFT) to get a spectrum of a signal. But there is the Discrete Fourier Transform (DFT) that can help us. To get it, let's start with the Fourier series. Present our original continuous signal as the Fourier series:

$$s(t) = \sum_{k=-\infty}^{+\infty} c_k e^{j\omega_k t};\ \omega_k = \frac{2\pi k}{T}$$

This equation performs for a **periodic** continuous signal with a period equals *T*. Let's sample this signal.

$$s(nt_s) = x(n) = x_n = \sum_{k=-\infty}^{+\infty} c_k e^{+j\omega_k n t_s} = \sum_{k=-\infty}^{+\infty} c_k e^{j\frac{2\pi k}{T} n t_s} = \sum_{k=-\infty}^{+\infty} c_k e^{j\frac{2\pi k}{N t_s} n t_s} = \sum_{k=-\infty}^{+\infty} c_k e^{j\frac{2\pi kn}{N}}$$

Then rewrite the summation in such a way:

$$x_n = \sum_{k=-\infty}^{+\infty} c_k e^{j\frac{2\pi kn}{N}} = \sum_{m=-\infty}^{+\infty}\sum_{k=0}^{N-1} c_{k+mN}\cdot e^{j\frac{2\pi(k+mN)n}{N}} = \sum_{m=-\infty}^{+\infty}\sum_{k=0}^{N-1} c_{k+mN}\cdot e^{j\frac{2\pi kn}{N}}\cdot e^{j\frac{2\pi mNn}{N}}$$

$$= \sum_{m=-\infty}^{+\infty}\sum_{k=0}^{N-1} c_{k+mN}\cdot e^{j\frac{2\pi kn}{N}}\cdot \underbrace{e^{j2\pi mn}}_{1} = \sum_{k=0}^{N-1} e^{j\frac{2\pi kn}{N}}\cdot \underbrace{\sum_{m=-\infty}^{+\infty} c_{k+mN}}_{X_k} = \sum_{k=0}^{N-1} X_k\cdot e^{j\frac{2\pi kn}{N}}$$

That is

$$x_n = \sum_{k=0}^{N-1} X_k\cdot e^{j\frac{2\pi kn}{N}};\ X_k = \sum_{m=-\infty}^{+\infty} c_{k+mN}$$

It is the formula for the inverse Discrete Fourier Transform. Go further

$$\sum_{n=0}^{N-1} x_n\cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} X_k\cdot e^{j\frac{2\pi kn}{N}}\cdot e^{-j\frac{2\pi mn}{N}} = \sum_{k=0}^{N-1} X_k\cdot\sum_{n=0}^{N-1} e^{j\frac{2\pi(k-m)n}{N}}$$

The last summation is a geometric progression, where

$$b_0 = 1; q = e^{j\frac{2\pi(k-m)}{N}}$$

Thus, according to material from §1.1, the summation, can be transformed to

$$\sum_{n=0}^{N-1} e^{j\frac{2\pi(k-m)n}{N}} = \begin{cases} 1\cdot\dfrac{1-q^N}{1-q}, \text{if } k\neq m \\ N, \text{if } k = m\end{cases} = \begin{cases} \dfrac{1-e^{j2\pi(k-m)}}{1-e^{j\frac{2\pi(k-m)}{N}}}, \text{if } k\neq m \\ N, \text{if } k = m\end{cases} = \begin{cases} 0, \text{if } k\neq m \\ N, \text{if } k = m\end{cases}$$

Because if *k* = *m*

$$e^{j\frac{2\pi(k-m)n}{N}} = e^{j0} = 1;\ \sum_{n=0}^{N-1} e^{j\frac{2\pi(k-m)n}{N}} = \sum_{n=0}^{N-1} 1 = N$$

If *k* ≠ *m*, then

$$e^{j2\pi(k-m)} = e^{j2\pi l} = 1;\ 1 - e^{j2\pi(k-m)} = 0$$

Thus, we have

$$\sum_{n=0}^{N-1} x_n\cdot e^{-j\frac{2\pi mn}{N}} = N\cdot X_m \Leftrightarrow X_m = \frac{1}{N}\cdot\sum_{n=0}^{N-1} x_n\cdot e^{-j\frac{2\pi mn}{N}}$$

This is direct DFT.

Typically, we are of interest in ratio between spectrum components (and use dBs for that), and there is no need to have multiplication by 1/*N* factor in Direct transform. So, in DSP this factor is set at Inverse transform and our formulas finally become

| | |
|---|---|
| $X_m = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi mn}{N}}$ | Direct DFT |
| $x_n = \frac{1}{N} \cdot \sum_{m=0}^{N-1} X_m \cdot e^{j\frac{2\pi mn}{N}}$ | Inverse DFT |

## §4.2 Example of a DFT calculation

Take, for example, sequence $x(n)$ as:

$$x(n) = \sin(2\pi f_1 n t_s) + 0{,}5 \sin\left(2\pi f_2 n t_s + \frac{3\pi}{4}\right)$$

where $f_1$ = 1000 Hz, $f_2$ = 2000 Hz, $t_s$ = 1/8000 s. The main period of this signal is $T$ = 1/1000 s. As a consequence

$$N = T \cdot f_s = \frac{1}{1000} \cdot 8000 = 8$$

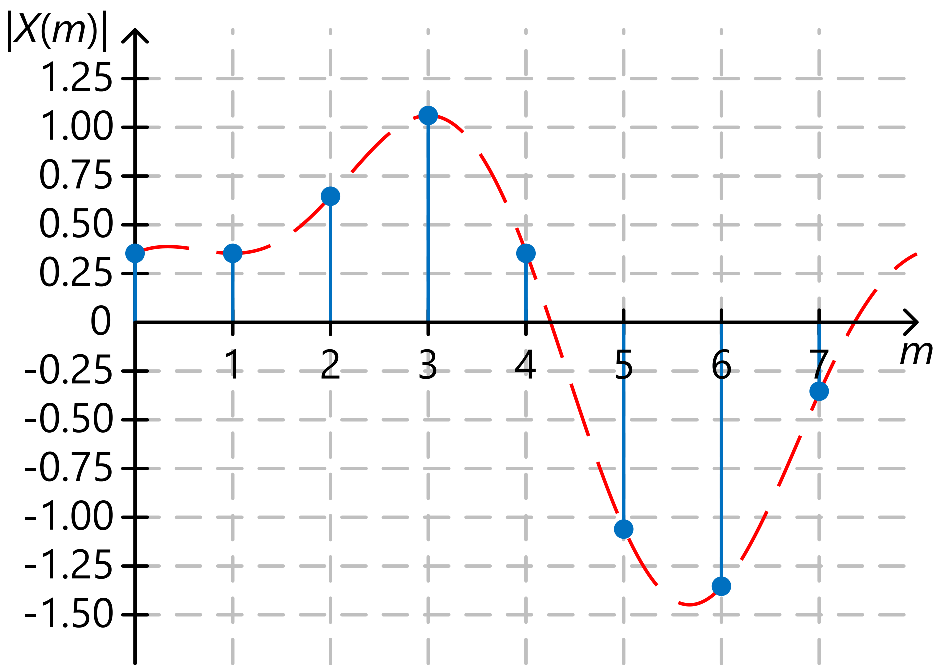



Figure 4.1 – Input sequence samples

Calculate input sequence samples:

| $n/m$ | $x(n)$ | $X(m)$ | $\lvert X(m)\rvert$ | arg $X(m)$ |
|---|---|---|---|---|
| 0 | 0,3536 | 0 + 0i | 0 | 0 |
| 1 | 0,3536 | 0 - 4$j$ | 4 | -π/2 |
| 2 | 0,6464 | 1,4142 + 1,4142$j$ | 2 | π/4 |
| 3 | 1,0607 | 0 + 0$j$ | 0 | 0 |
| 4 | 0,3536 | 0 + 0$j$ | 0 | 0 |
| 5 | -1,0607 | 0 + 0$j$ | 0 | 0 |
| 6 | -1,3536 | 1,4142 - 1,4142$j$ | 2 | -π/4 |
| 7 | -0,3536 | 0 + 4$j$ | 4 | π/2 |

Plots of spectrum magnitude and phase are presented in Figures 4.2 and 4.3.

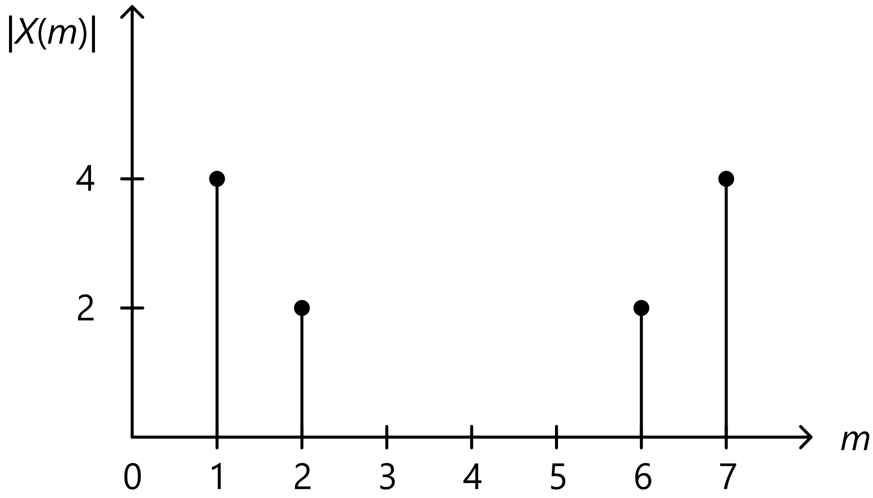



Figure 4.2 – Magnitude of spectrum

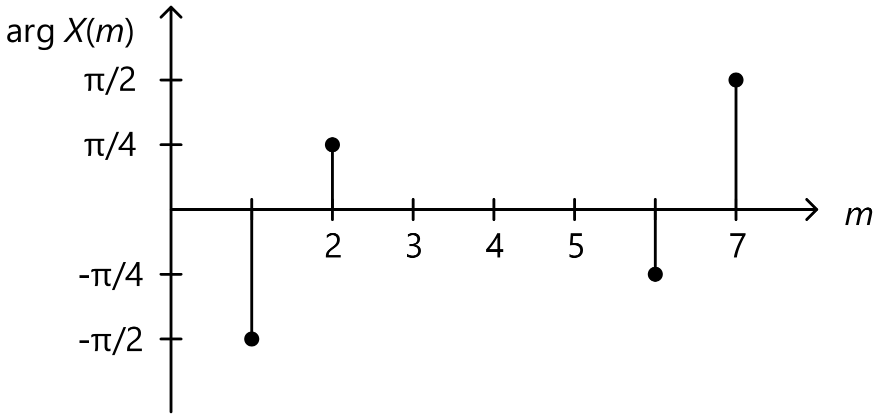



Figure 4.3 – Phase of spectrum

## §4.3 Properties of DFT

*4.3.1 Axes conversion (magnitude and frequency)*

As we have seen from the previous paragraph, a spectrum sample $X_m$ depends on a dimensionless index $m$. So, there is a question: how to convert an index into a frequency? The answer is at the beginning of our derivation. We expanded our signal into a Fourier series with frequencies:

$$\omega_k = \frac{2\pi k}{T} \Leftrightarrow f_k = \frac{k}{T} = \frac{k}{Nt_s} = \frac{kf_s}{N};\ m \in \mathbb{Z}$$

That is, $k$ corresponds to a frequency $f_k$, as well as, $n$ corresponds to a time point $t_n$:

$$k \to f_k = \frac{k}{T}; n \to t_n = nt_s$$

Moreover, from the example we have seen that an amplitude of our signal in a spectrum isn't 1. There are 2 reasons for it:

1. After derivation, we have changed a position of the factor $1/N$ from direct to inverse FT. As usually, we are interested in a ratio between spectrum components and plot it in dB, there is no need in the division by $N$;
2. As our signal is real, it has a symmetric spectrum. It means that the energy and the amplitude divide into 2 parts: with negative frequencies and with positive frequencies.

It results in the next equations for actual spectrum component amplitude $A_m$:

$$A_m = \frac{X_m}{N/2} = \frac{2X_m}{N} \quad \text{for real signal}$$

$$A_m = \frac{X_m}{N} \qquad \text{for complex signal}$$

*4.3.2 How T, $f_s$ and N effect on spectrum?*

Let's start from a basic equation for a DFT calculation:

$$T \cdot f_s = N$$

There are only 2 independent variables: analysis time *T* and sampling frequency $f_s$. A number of samples *N* can be obtained from these 2 parameters. Now, let's discuss 2 cases of changing these variables.

*1. Constant $f_s$*

We know that sampling frequency determines our repetition period of a spectrum. In other words, it determines our frequency band for observation. If we fix this value and increase samples number, it will result into an analysis time increase.

$$f_s - const; \uparrow N \Leftrightarrow \uparrow T$$

How does it change spectrum? Let's have a look at frequencies at in a spectrum. From the previous point we know that frequencies are:

$$f_k = \frac{k}{T}$$

That is, an interval between spectrum samples is:

$$\Delta f = \frac{1}{T}$$

This is a spectrum resolution. So that, if we increase *T* or *N* with a constant $f_s$, we will increase the resolution of our spectrum.

*2. Constant T*

If analysis time is constant, an increase of a samples number results into an increase of sampling frequency.

$$T - const; \uparrow N \Leftrightarrow \uparrow f_s$$

In this case, a spectrum resolution remains the same, but an observation frequency band becomes wider.

*4.3.3 Linearity*

Remind formulas for DFT:

$$X_m = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi mn}{N}}; \; x_n = \frac{1}{N} \cdot \sum_{m=0}^{N-1} X_k \cdot e^{j\frac{2\pi mn}{N}}$$

From here, it is seen that DFT, as other Fourier Transforms, is linear operator:

$$\mathcal{F}\{\alpha x(n) + \beta y(n)\} = \alpha \mathcal{F}\{x(n)\} + \beta \mathcal{F}\{y(n)\}$$

▲ Home exercises: proof linearity.

*4.3.4 Shifting theorem*

If we shift our signal in time by *k* samples, it will affect only phase shift of the spectrum.

$$x(n-k) \leftrightarrow e^{-j\frac{2\pi mk}{N}} \cdot X(m)$$

Consider shifted sequence $x'(n)$ is:

$$x'(n) = x(n-k)$$

Then its DFT:

$$X'(m) = \sum_{n=0}^{N-1} x'(n) \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=0}^{N-1} x(n-k) \cdot e^{-j\frac{2\pi mn}{N}} = \begin{vmatrix} n-k \to l \\ n \to l+k \end{vmatrix} = \sum_{l=-k}^{N-1-k} x(l) \cdot e^{-j\frac{2\pi m(l+k)}{N}}$$

$$= e^{-j\frac{2\pi mk}{N}} \cdot \sum_{l=-k}^{N-1-k} x(l) \cdot e^{-j\frac{2\pi ml}{N}} = e^{-j\frac{2\pi mk}{N}} \cdot X(m)$$

For the last step it has been assumed that:

$$\sum_{l=-k}^{N-1-k} x(l) \cdot e^{-j\frac{2\pi ml}{N}} = \sum_{l=0}^{N-1} x(l) \cdot e^{-j\frac{2\pi ml}{N}} = X(m)$$

And it is true, because our sequence for DFT is assumed to be periodic with period of *N* samples.

▲ Home exercise: proof equivalence above.

*4.3.5 Theorem of convolution*

For DFT, theorem of convolution is the same as for other transforms. That is

$$x * h(n) \leftrightarrow X(m) \cdot H(m).$$

$$y(n) = x * h(n) = \sum_{k=0}^{N-1} x(k) \cdot h(n-k)$$

▲ Home exercise: proof theorem of convolution.

*4.3.6 Symmetry*

For a real signal there is a following property:

$$X(-m) = X^*(m)$$

It requires an easy proof. We know for real signal:

$$x^*(n) = x(n)$$

So:

$$X(-m) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi(-m)n}{N}} = \sum_{n=0}^{N-1} x(n) \cdot e^{j\frac{2\pi mn}{N}}$$

$$X^*(m) = \left(\sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi mn}{N}}\right)^* = \sum_{n=0}^{N-1} x^*(n) \cdot e^{j\frac{2\pi mn}{N}} = \sum_{n=0}^{N-1} x(n) \cdot e^{j\frac{2\pi mn}{N}} = X(-m)$$

As our spectrum is periodic

$$X(-m) = X(N-m) = X(kN-m), \text{where } k \in \mathbb{Z}$$

In other words:

$$X(N-m) = X^*(m)$$

This property may help in DFT calculation: half of spectrum samples can be obtained by complex conjugating another half.

## §4.4 Symmetric DFT forms

There are symmetric DFT forms.

| | |
|---|---|
| $X_m = \frac{1}{\sqrt{N}} \cdot \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi mn}{N}}$ | Direct DFT |
| $x_n = \frac{1}{\sqrt{N}} \cdot \sum_{m=0}^{N-1} X_k \cdot e^{j\frac{2\pi mn}{N}}$ | Inverse DFT |

They have an identical scale factor, and we see that DFT forms differ only by a sign in the exponent. This feature can slightly simplify design of systems utilizing DFT in the both directions (you can use exactly the same block without need of scale adjustment). Magnitude scale of $X_m$ and $x_n$ is still not the same as in ordinary DFT, but more close to one another (differs by $\sqrt{N}$ times instead of *N* times). Truly the same scale of $X_m$ and $x_n$ has form that has been obtained from the derivation of DFT.

## §4.5 DFT matrix

A calculation of DFT can be done by means of matrixes. A vector of spectrum samples $\bar{X}(m)$ is defined from a vector of input samples $\bar{x}(n)$ as:

$$\bar{X}(m) = \begin{bmatrix} X(0) \\ X(2) \\ \dots \\ X(N-1) \end{bmatrix} = \underbrace{\begin{bmatrix} e^{-j2\pi\frac{0\cdot 0}{N}} & e^{-j2\pi\frac{0\cdot 1}{N}} & \dots & e^{-j2\pi\frac{0\cdot (N-1)}{N}} \\ e^{-j2\pi\frac{1\cdot 0}{N}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ e^{-j2\pi\frac{(N-1)\cdot 0}{N}} & e^{-j2\pi\frac{(N-1)\cdot 1}{N}} & \dots & e^{-j2\pi\frac{(N-1)\cdot (N-1)}{N}} \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} x(0) \\ x(2) \\ \dots \\ x(N-1) \end{bmatrix}}_{\bar{x}(n)} = D\bar{x}(n)$$

where $D$ – a DFT matrix.

## §4.6 DFT of typical functions

### *4.6.1 General rectangular function*

In this section we calculate DFT for several common functions. At first, let's take general rectangular function depicted in Figure 4.4.

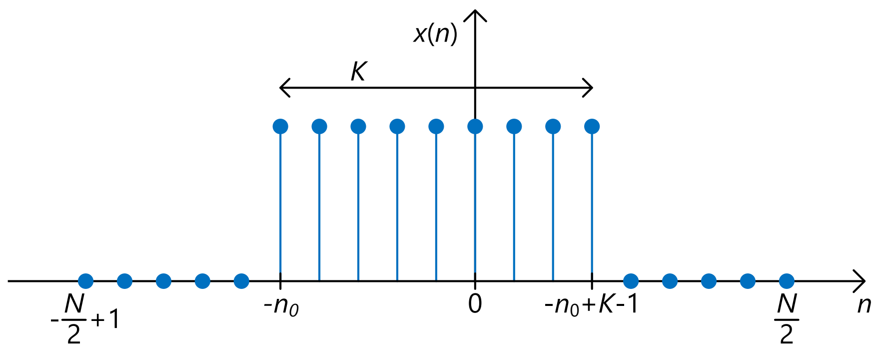



Figure 4.4 – General rectangular function

This function can be expressed by

$$x(n) = \begin{cases} 0, & n < -n_0 \\ 1, & -n_0 \le n \le -n_0 + (K-1) \\ 0, & n > -n_0 \end{cases}$$

Calculate DFT for it.

$$X_m = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} x_n \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=-n_0}^{-n_0+(K-1)} x_n \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=-n_0}^{-n_0+(K-1)} e^{-j\frac{2\pi mn}{N}} = e^{-j\frac{2\pi m(-n_0)}{N}} \cdot \frac{1-e^{-j2\pi m\frac{K}{N}}}{1-e^{-j\frac{2\pi m}{N}}}$$

$$= e^{-j\frac{2\pi m(-n_0)}{N}} \cdot \frac{e^{-j\frac{2\pi mK}{2N}}}{e^{-j\frac{2\pi m}{2N}}} \cdot \frac{e^{j\frac{2\pi mK}{2N}} - e^{-j\frac{2\pi mK}{2N}}}{e^{j\frac{2\pi m}{2N}} - e^{-j\frac{2\pi m}{2N}}} = e^{-j\frac{2\pi m}{N}\left(-n_0+\frac{K-1}{2}\right)} \cdot \frac{\sin\left(\pi m\frac{K}{N}\right)}{\sin\left(\pi m\frac{1}{N}\right)}$$

Finally, we get

$$X_m = e^{-j\frac{2\pi m}{N}\left(-n_0+\frac{K-1}{2}\right)} \cdot \frac{\sin\left(\pi m\frac{K}{N}\right)}{\sin\left(\pi m\frac{1}{N}\right)}.$$

This expression has indeterminate form at $m = 0$. Use L'Hospital's rule

$$X_0 = \lim_{m\to 0} \frac{\sin\left(\pi m\frac{K}{N}\right)}{\sin\left(\pi m\frac{1}{N}\right)} = \lim_{m\to 0} \frac{\left(\sin\left(\pi m\frac{K}{N}\right)\right)'}{\left(\sin\left(\pi m\frac{1}{N}\right)\right)'} = \lim_{m\to 0} \frac{\pi\frac{K}{N}\cdot\cos\left(\pi m\frac{K}{N}\right)}{\pi\frac{1}{N}\cdot\cos\left(\pi m\frac{1}{N}\right)} = K$$

Magnitude of the spectrum is expressed by

$$|X_m| = \left|\frac{\sin\left(\pi m\frac{K}{N}\right)}{\sin\left(\pi m\frac{1}{N}\right)}\right|$$

and is illustrated in Figure 4.5.

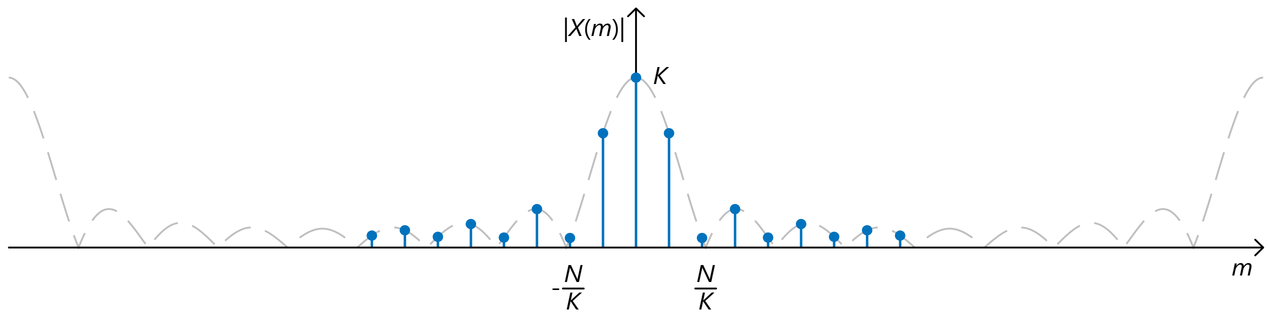



Figure 4.5 – Spectrum of general rectangular function

for $N \to \infty$,

$$|X_m| = \left|\frac{\sin\left(\pi m \frac{K}{N}\right)}{\sin\left(\pi m \frac{1}{N}\right)}\right| \approx \left|\frac{\sin\left(\pi m \frac{K}{N}\right)}{\pi m \frac{1}{N}}\right|$$

Find zeroes of the spectrum

$$\pi m \frac{K}{N} = \pi l \Leftrightarrow m = l \cdot \frac{N}{K}, l \in \mathbb{Z}\backslash\{0\}$$

### *4.6.2 Symmetric rectangular function*

Now, let's consider symmetric rectangular function, i.e.

$$x(n) = x(-n).$$

Calculate shift $n_0$ for this case. Due to the symmetry, the left and the right boundaries for $n$ should have opposite values, that is

$$-n_0 + K - 1 = -(-n_0)$$

Then

$$n_0 = -n_0 + K - 1; n_0 = \frac{K-1}{2}$$

An example of the symmetric rectangular function is depicted in Figure 4.6.

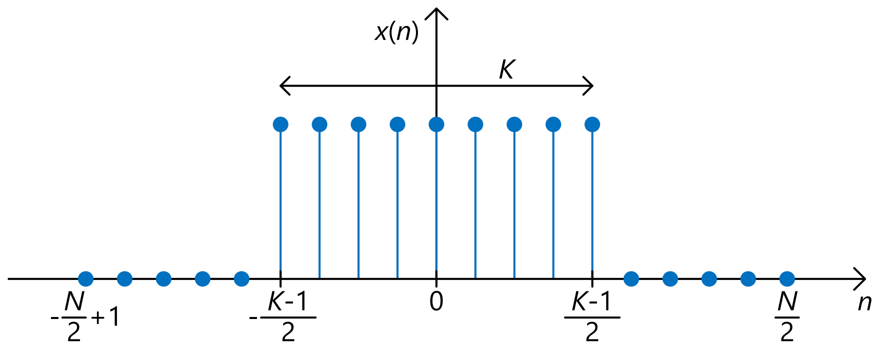



Figure 4.6 – Symmetric rectangular function

So the result will be

$$X_m = \underbrace{e^{-j\frac{2\pi m}{N}\left(-\frac{K-1}{2}+\frac{K-1}{2}\right)}}_{1} \cdot \frac{\sin\left(\pi m \frac{K}{N}\right)}{\sin\left(\pi m \frac{1}{N}\right)} = \frac{\sin\left(\pi m \frac{K}{N}\right)}{\sin\left(\pi m \frac{1}{N}\right)}$$

As seen, the spectrum is real. A graph of the spectrum coincides with its magnitude in figure 4.5.

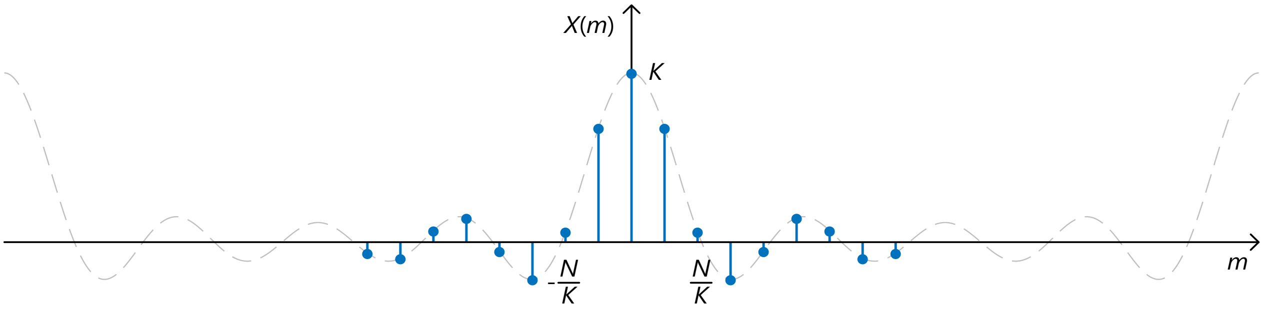



Figure 4.7 – Spectrum of symmetric rectangular function

*4.6.3 Constant level*

Now let's have a look at constant level (Figure 4.8). Spectrum of a constant level can be obtained from the previous result. If we assume

$$K = N,$$

then the DFT transforms to

$$X_m = \frac{\sin\left(\pi m \frac{N}{N}\right)}{\sin\left(\pi m \frac{1}{N}\right)} = \frac{\sin(\pi m)}{\sin\left(\pi m \frac{1}{N}\right)} = \begin{cases} N, m = 0 \\ 0, m \neq 0 \end{cases}$$

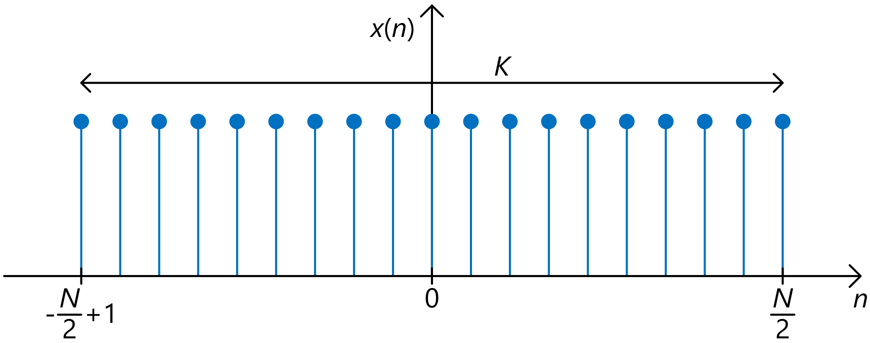



Figure 4.8 – Constant level

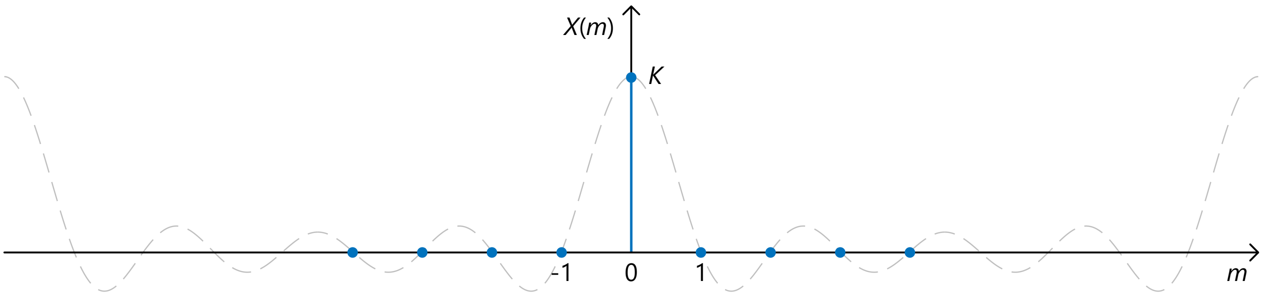



Figure 4.9 – Spectrum of the constant level

*4.6.4 IDFT of rectangular function*

Consider Inverse DFT. Also, take general rectangular function (Figure 4.10). This function can be expressed by

$$X(m) = \begin{cases} 0, & m < -m_0 \\ 1, & -m_0 \leq m \leq -m_0 + (K-1) \\ 0, & m > -m_0 \end{cases}$$

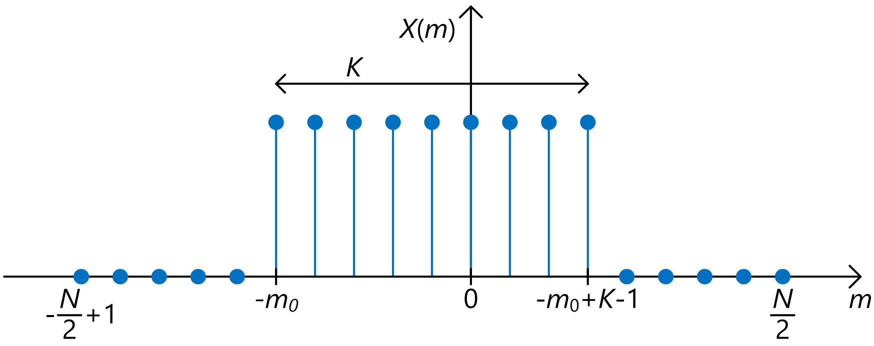



Figure 4.10 – Rectangular function in the frequency domain

Calculate DFT for it.

$$x_n = \frac{1}{N} \sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}} X_m \cdot e^{j\frac{2\pi mn}{N}}$$

Calculation flow is the same as in point 4.6.1. So we can just exchange *n* and *m* and change sign in the exponent. Result will be

$$x_n = \frac{1}{N} \cdot e^{+j\frac{2\pi n}{N}\left(-m_0+\frac{K-1}{2}\right)} \cdot \frac{\sin\left(\pi n \frac{K}{N}\right)}{\sin\left(\pi n \frac{1}{N}\right)}$$

Such a spectrum is depicted in Figure 4.11 and is similar to another one in Figure 4.5. You can see that sequence *x*(*n*) is complex. It is due-to asymmetric spectrum. If we imply symmetric spectrum (like in point 4.6.2), sequence becomes

$$x_n = \frac{1}{N} \cdot \frac{\sin\left(\pi n \frac{K}{N}\right)}{\sin\left(\pi n \frac{1}{N}\right)},$$

that is real sequence.

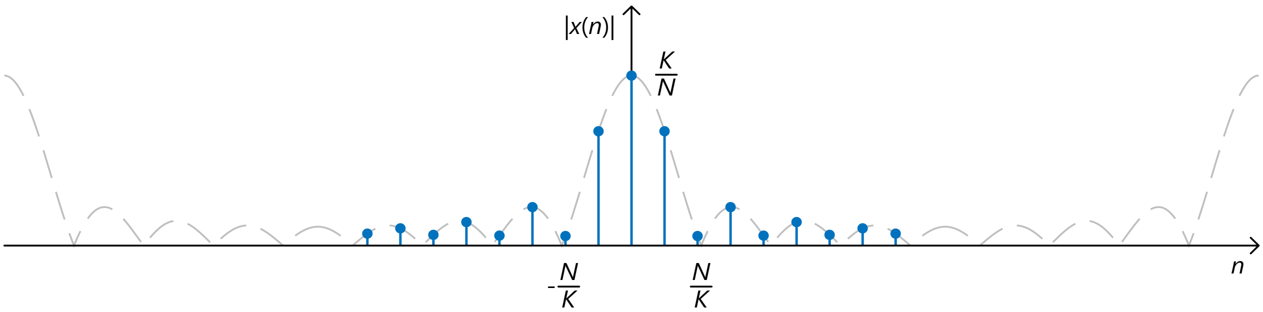



Figure 4.11 – Signal *x*(*n*) with the rectangular spectrum

### *4.6.5 Complex signal*

Introduce complex signal $x_n$ with frequency $\omega_k$

$$x_n = e^{j\omega_k t_n}; \omega_k = 2\pi \cdot \frac{k}{T}.$$

Rewrite the exponent argument

$$\omega_k t_n = 2\pi \cdot \frac{k}{T} \cdot n t_s = 2\pi \cdot \frac{kn}{T \cdot f_s} = 2\pi \frac{kn}{N}.$$

So signal can be expressed as

$$x_n = e^{j2\pi\frac{kn}{N}}$$

where *k* – number of periods, *N* – number of samples. Illustration of signal $x_n$ is presented in Figure 4.12.

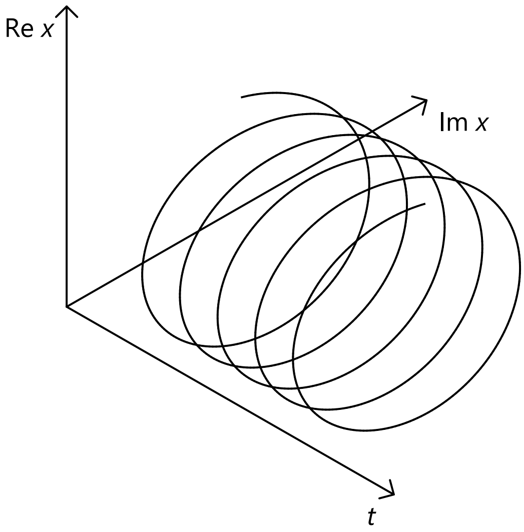



Figure 4.12 – Illustration of complex signal $x_n$

Now calculate DFT of this signal. We take it with symmetric bounds to eliminate phase shift (as we know from 4.6.2) and, as consequence, $N$ should be odd.

$$X_m = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} x_n \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j2\pi\frac{kn}{N}} \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{-j\frac{2\pi(m-k)n}{N}}$$

$$= e^{-j\frac{2\pi(m-k)}{N}\left(-\frac{N-1}{2}\right)} \cdot \frac{1 - e^{-j\frac{2\pi(m-k)N}{N}}}{1 - e^{-j\frac{2\pi(m-k)}{N}}}$$

$$= e^{-j\frac{2\pi(m-k)}{N}\cdot\left(-\frac{N-1}{2}\right)} \cdot \frac{e^{-j\frac{2\pi(m-k)N}{2N}}}{e^{-j\frac{2\pi(m-k)}{2N}}} \cdot \frac{e^{j\frac{2\pi(m-k)}{2}} - e^{-j\frac{2\pi(m-k)}{2}}}{e^{j\frac{2\pi(m-k)}{2N}} - e^{-j\frac{2\pi(m-k)}{2N}}}$$

$$= e^{-j\frac{2\pi(m-k)}{N}\cdot\overbrace{\left(-\frac{N-1}{2}+\frac{N-1}{2}\right)}^{0}} \cdot \frac{\sin\left(\frac{2\pi(m-k)}{2}\right)}{\sin\left(\frac{2\pi(m-k)}{2N}\right)} = \frac{\sin\big(\pi(m-k)\big)}{\sin\left(\pi\frac{m-k}{N}\right)}.$$

Spectrum of such a signal is depicted in Figure 4.13. For the expressions above, we can see that for $k \neq 0$ this function is not even. That is, the spectrum is not symmetrical about the $y$-axis. It is a feature of a complex signal.

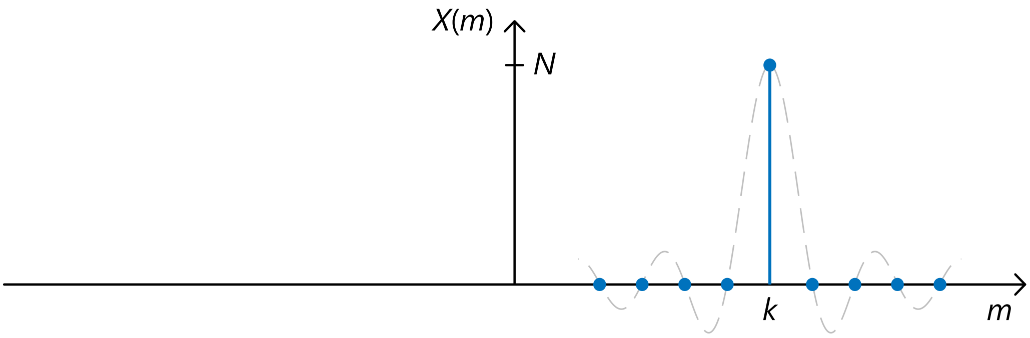



Figure 4.13 – Spectrum of the complex signal

*4.6.6 Real signal*

Take real signal, for instance

$$x_n = \cos\omega_k t_n\,;\, \omega_k = 2\pi\cdot\frac{k}{T}.$$

To simplify DFT calculations, we can present it as

$$x_n = \cos\omega_k t_n = \frac{e^{j\omega_k t_n} + e^{-j\omega_k t_n}}{2}$$

Result for the complex signal is already known. As consequence, we can immediately write the spectrum

$$X_m = \frac{1}{2}\cdot\frac{\sin\left(\pi(m-k)\right)}{\sin\left(\pi\frac{m-k}{N}\right)} + \frac{1}{2}\cdot\frac{\sin\left(\pi(m+k)\right)}{\sin\left(\pi\frac{m+k}{N}\right)}$$

Spectrum in all point, except $m = \pm k$, equals 0. Show it for $m = k$.

$$X_k = \frac{1}{2}\cdot\frac{\sin\left(\pi(k-k)\right)}{\sin\left(\pi\frac{k-k}{N}\right)} + \frac{1}{2}\cdot\frac{\sin\left(\pi(k+k)\right)}{\sin\left(\pi\frac{k+k}{N}\right)} = \frac{1}{2}\cdot 1 + 0 = \frac{1}{2}.$$

Thus, we get

$$X_m = \begin{cases} 0, & m \neq \pm k \\ \frac{1}{2}, & m = \pm k \end{cases}$$

We see that signal has real spectrum due to evenness of a cosine function. The spectrum is depicted in Figure 4.14.

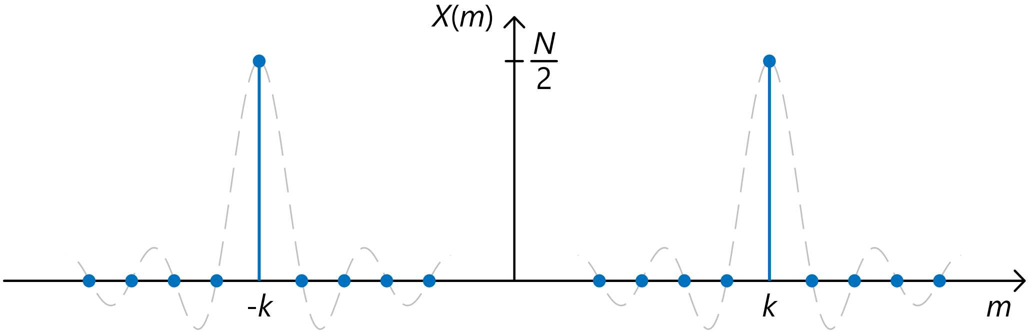



Figure 4.14 – Spectrum of the real cosine signal

A DFT for the signal of sine can be obtained similarly. Present such a signal as

$$x_n = \sin\omega_k t_n = \frac{e^{j\omega_k t_n} - e^{-j\omega_k t_n}}{2j}$$

Then DFT will be

$$X_m = \frac{1}{2j}\cdot\frac{\sin\left(\pi(m-k)\right)}{\sin\left(\pi\frac{m-k}{N}\right)} - \frac{1}{2j}\cdot\frac{\sin\left(\pi(m+k)\right)}{\sin\left(\pi\frac{m+k}{N}\right)} = \begin{cases} 0, & m \neq \pm k \\ \mp\frac{j}{2}, & m = \pm k \end{cases}$$

The spectrum is illustrated in Figure 4.15.

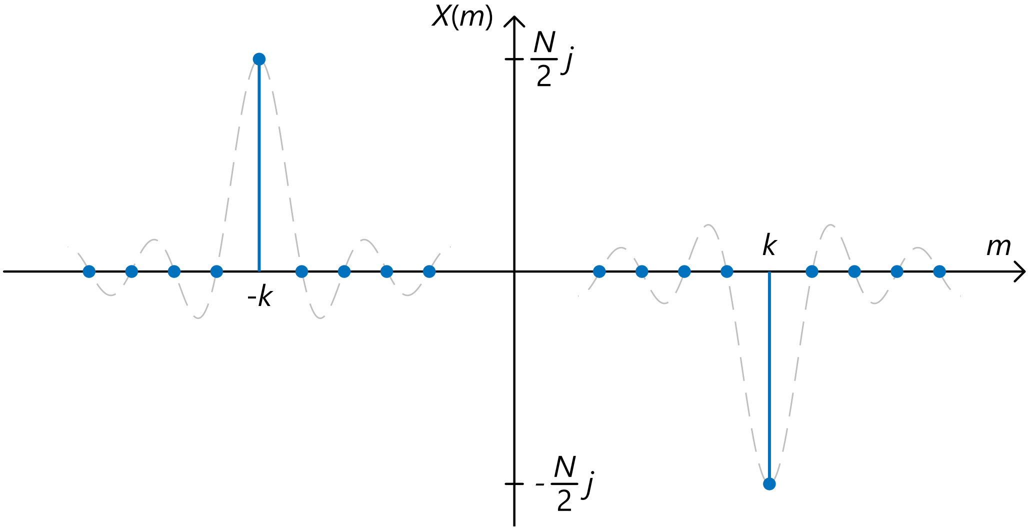



Figure 4.15 – Spectrum of the real sine signal

## §4.7 Leakage

Now we have a look at very important feature of the DFT. Take two discrete sequences with different number of periods (Figure 4.16). A calculation of their DFTs give us spectrums shown in Figure 4.17.

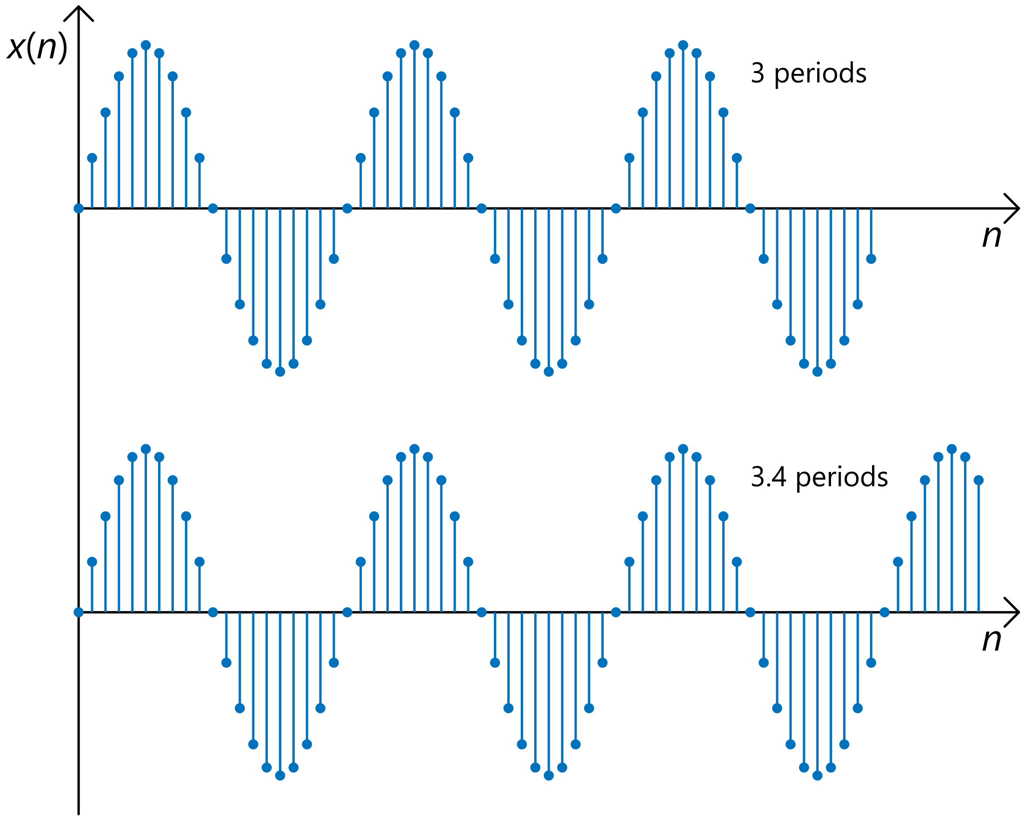
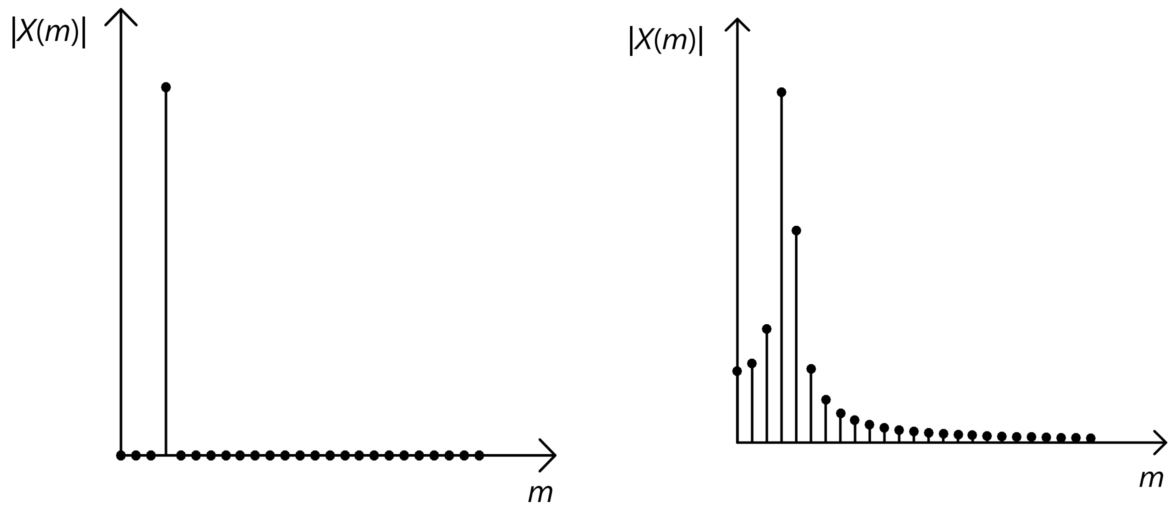



Figure 4.16 – Discrete sequences with 3 and 3.4 periods




a) b)

Figure 4.17 – DFT module of a discrete sequence a) with 3 periods, b) with 3.4 periods

You see that spectrum in Figure 4.17a looks normal, but another one in Figure 4.17b looks strange. Why is it that? Because there was an effect called leakage. Let's have a deep look to their DFTs.

As we know from §4.6.6, a DFT of a harmonic discrete sequence with $k$ periods is the following function:

$$X(m) = \frac{1}{2}e^{j\pi\left(k-m-\frac{k-m}{N}\right)}\frac{\sin\left(\pi(k-m)\right)}{\sin\left(\pi\frac{k-m}{N}\right)} + \frac{1}{2}e^{j\pi\left(k+m-\frac{k+m}{N}\right)}\frac{\sin\left(\pi(k+m)\right)}{\sin\left(\pi\frac{k+m}{N}\right)}$$

There will be only two non-zero values – for $m = \pm k$ (see Figure 4.14). However, $m$ unlike $k$ can be only integer. As consequence, we can get different spectrum images in dependence of $k$, even though $X(m)$ is the same. These possible situations are depicted in Figure 4.18. If $k$ – an integer number, then non-zero values coincide with spectrum samples and we get a typical spectrum for a harmonic function. On the other hand, if *k* – not an integer number, then picture will be different and zero samples are absent. In the both cases the spectrum is a sampled function

$$\frac{\sin(\pi x)}{\sin\left(\pi \frac{x}{N}\right)}.$$

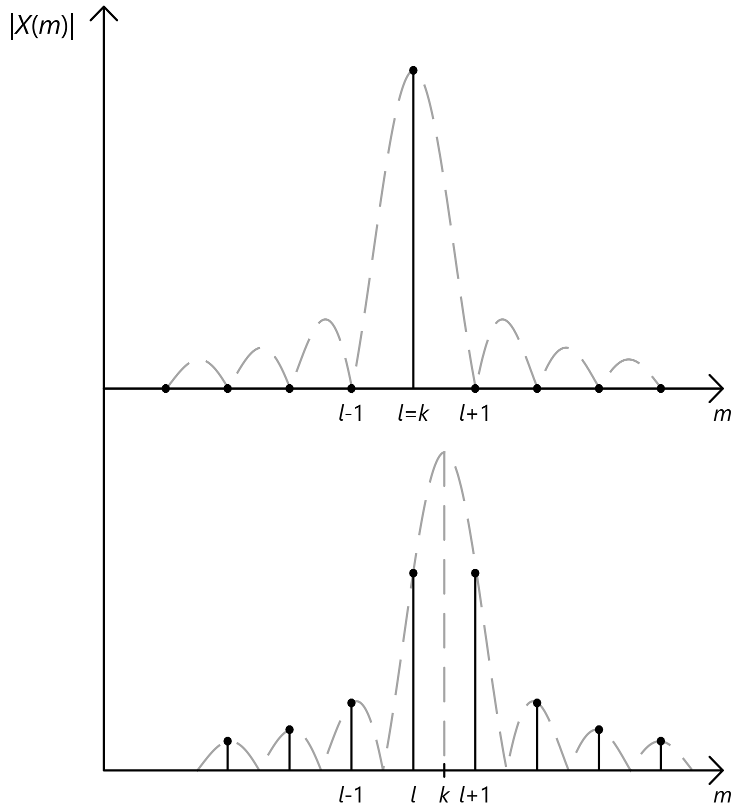



Figure 4.18 – Spectrum of a harmonic signal for integer *k* (top) and non-integer *k* (bottom)

You may notice that spectrum in Figure 4.17b is asymmetric unlike function in Figure 4.18. This is because non-zero samples of *X*(*m*) terms are summed up.

## §4.8 Windows

We have already understood that the leakage is due to the form of a function enveloping each signal frequency in the spectrum. Herein, the main cause deteriorates the spectrum is a level of side lobes. Can we change this enveloping function and reduce side lobes? Yes. For this purpose, there are window functions or just windows.

The most common windows are a triangular, a Hanning and a Hamming ones depicted in Figure 4.19. In addition to them a rectangular window is introduced representing window absence. From the figure, you can see that all windows have lower side lobes in comparison with the rectangular window. And this aspect very important in terms of leakage. In time domain, these function looks like in Figure 4.20 and can be expressed as

| Window | Function *w*(*t*) |
|---|---|
| Rectangular | $1 \text{ for } n \in \{0, \dots, N-1\}$ |
| Triangular | $\frac{n}{N/2}$ and $2 - \frac{n}{N/2}$ for $n \in \left\{0, \dots, \frac{N}{2}\right\}$ and $n \in \left\{\frac{N}{2}+1, \dots, N-1\right\}$ respectively |
| Hanning | $0.5 - 0.5 \cdot \cos\left(2\pi \frac{n}{N-1}\right) \text{ for } n \in \{0, \dots, N-1\}$ |
| Hamming | $0.54 - 0.46 \cdot \cos\left(2\pi \frac{n}{N-1}\right) \text{ for } n \in \{0, \dots, N-1\}$ |

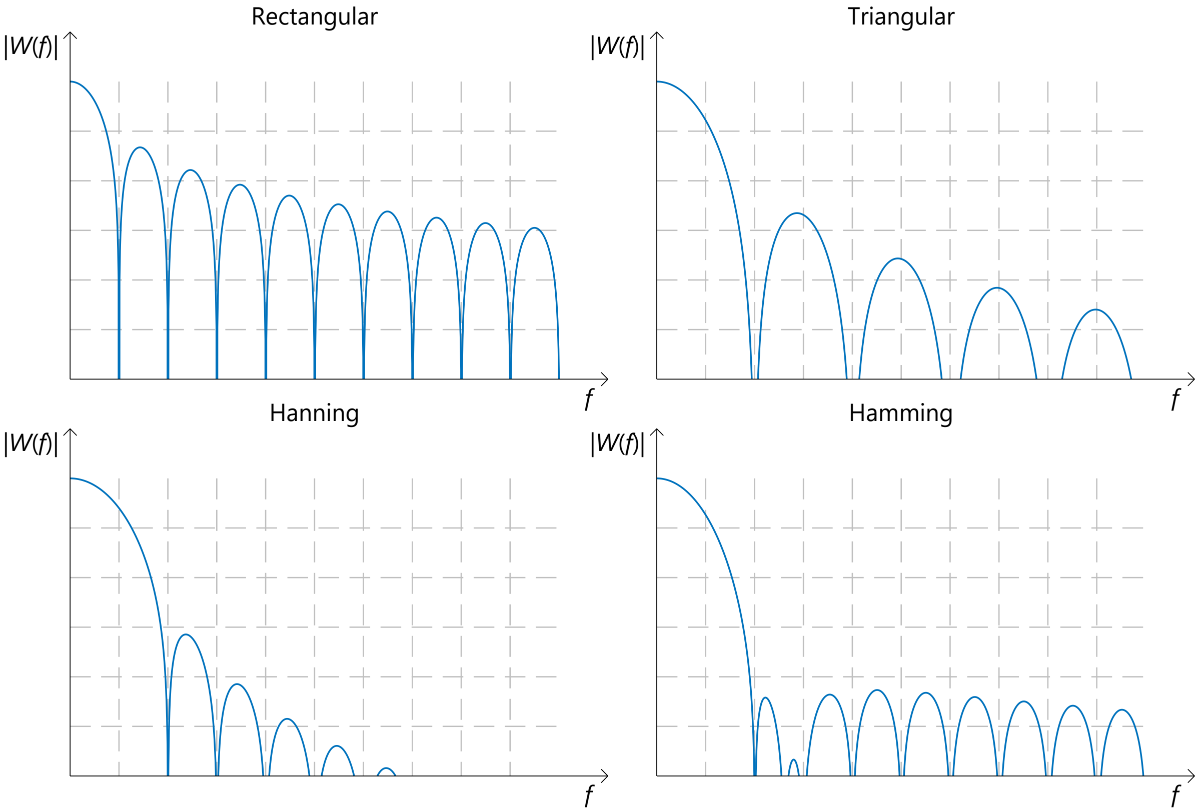



Figure 4.19 – Spectrums of common window functions

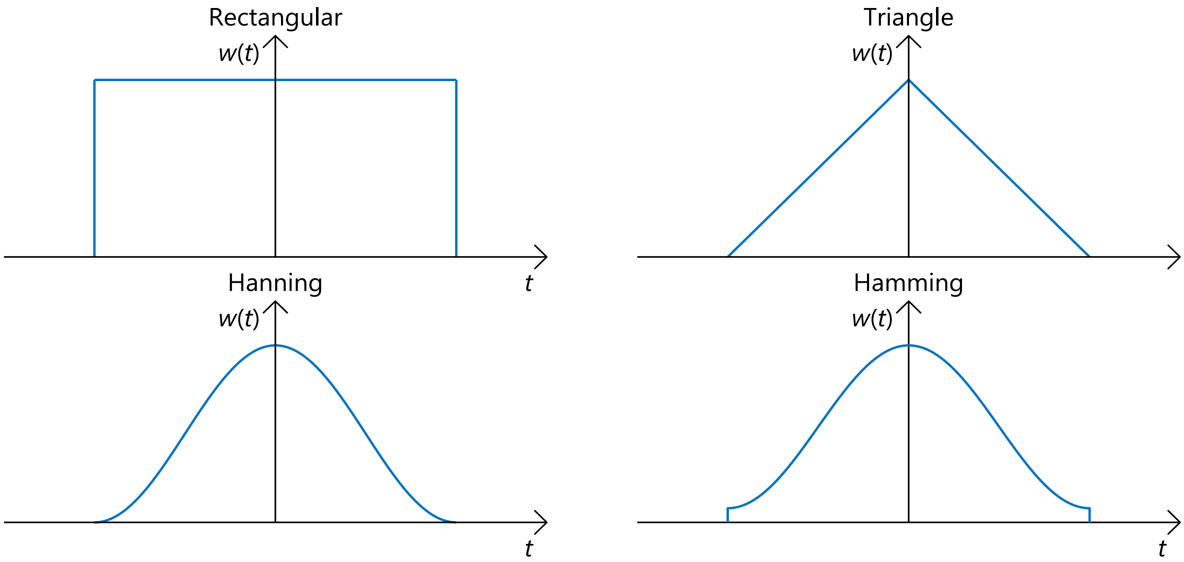



Figure 4.20 – Common window functions in time domain

How do we apply window? For this purpose, we just multiply our signal $x(n)$ by the window function $w(n)$, i.e.

$$x_{win}(n) = w(n) \cdot x(n)$$

In case of the rectangular window, we get

$$x_{win}(n) = w(n) \cdot x(n) = 1 \cdot x(n) = x(n),$$

that is, signal itself. Examples of applying window are illustrated in Figure 4.21.

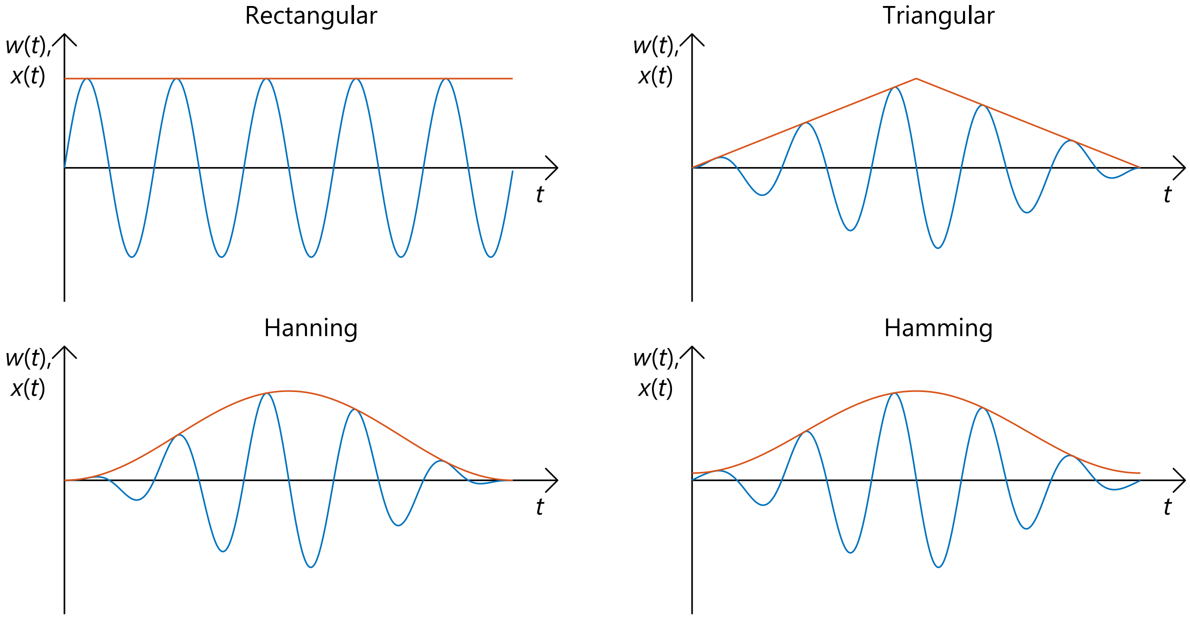



Figure 4.21 – Examples of applying window function

How does it work? We know that a multiplication in time domain means a convolution in frequency domain, i.e.

$$x_{win}(n) = w(n) \cdot x(n) \leftrightarrow X_{win}(m) = (W * X)(m).$$

Here *W*(*m*) is a sampled window spectrum (see its envelope in Figure 4.19). If we assume *x*(*n*) as a harmonic signal with frequency $f_k$, then a convolution result $X_{win}(m)$ will be a spectrum of window *W*(*m*) shifted by signal frequency $f_k$. In other words, $X_{win}(m)$ – a sampled window spectrum, which is centered at $f_k$. We have seen it for the rectangular window in Figure 4.18. There, a spectrum of the rectangular window has form

$$\frac{\sin(\pi x)}{\sin\left(\pi \frac{x}{N}\right)},$$

and it Is centered at *k* and sampled (*k* corresponds to frequency $f_k = k/T$).

Let's illustrate an effect of changing window. Take $f_k$ so that signal has

$$k = \frac{p+1}{2}; p \in \mathbb{N}$$

periods. In such a case, spectrum $X_{win}(m)$ will have leakage, since spectrum samples don't get to zeroes of an envelope function. You can see it in Figure 4.22 for the rectangular window, which is similar to Figure 4.18. Herewith, spectrums for the other windows even though have leakage, but allow to distinct our frequency better due to lower side lobes. However, we didn't get this enhancement for free. Side lobes lowering widens the main lobe, and now it contains 4 samples instead of 2. Nevertheless, in spite of the main component blur, windows are still an effective instrument in fight against the leakage.

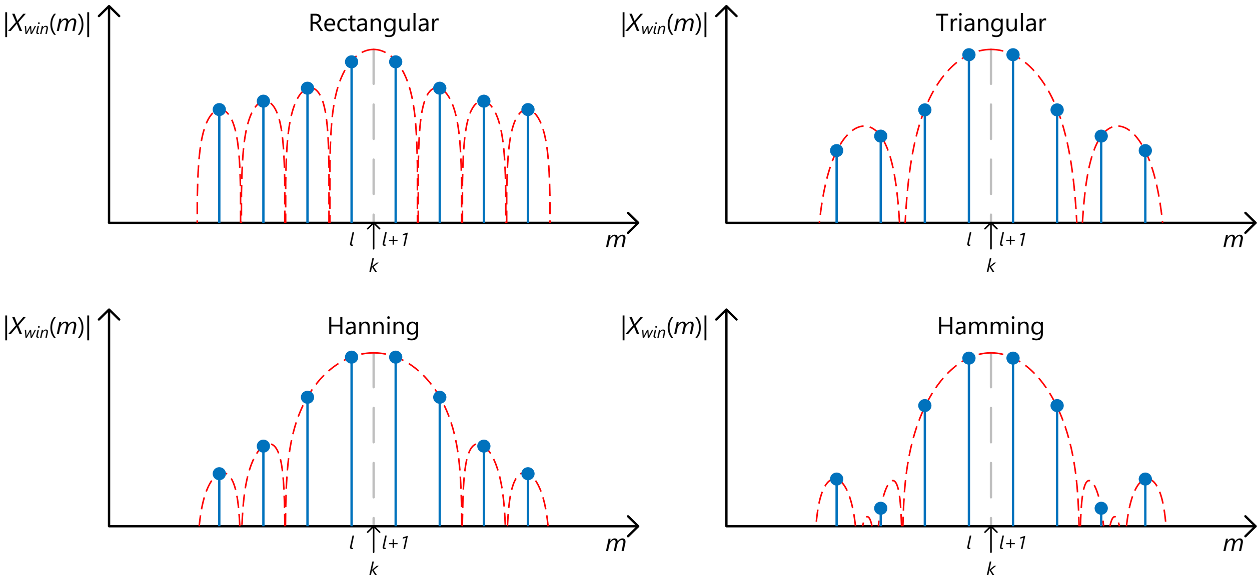



Figure 4.22 – Leakage effect for different windows

## §4.9 Signal to noise ratio in DFT

The Signal-to-Noise Ratio (SNR) indicates relation between wanted signal and unwanted signals (noise). Typical, expression for the SNR is

$$SNR = \frac{\text{Power of wanted signal}}{\text{Power of noise}}.$$

On the one hand, an amplitude of a harmonic signal increases in a proportion of $N$. An amplitude of noise is described as standard deviation, which increases as $\sqrt{N}$ due to its random nature (see dispersion of random values sum). So their ratio will increase as $N/\sqrt{N}=\sqrt{N}$. On the other hand, it is known that noise signal has random nature (both in terms of amplitude and frequency), so the probability of its frequency coinciding with a certain sample on the spectrum axis tends to 0. Thus, noise energy is spread across finite number of spectrum samples and noise level occurs quite high. An increase of analysis time enlarges a number of spectrum samples and resolution. It results in lower amplitude of noise sample and higher SNR value. The relation between number of samples and SNR enhancement is the following:

$$SNR_N = SNR_{N'} + 20\log_{10}\sqrt{\frac{N}{N'}} = SNR_{N'} + 10\log_{10}\frac{N}{N'}$$

This effect is illustrated in Figure 4.23. The top spectrum has noise sample at -40 dB. Enlargement of samples number by 100 times give lowering of noise sample to -60 dB level for the bottom spectrum.

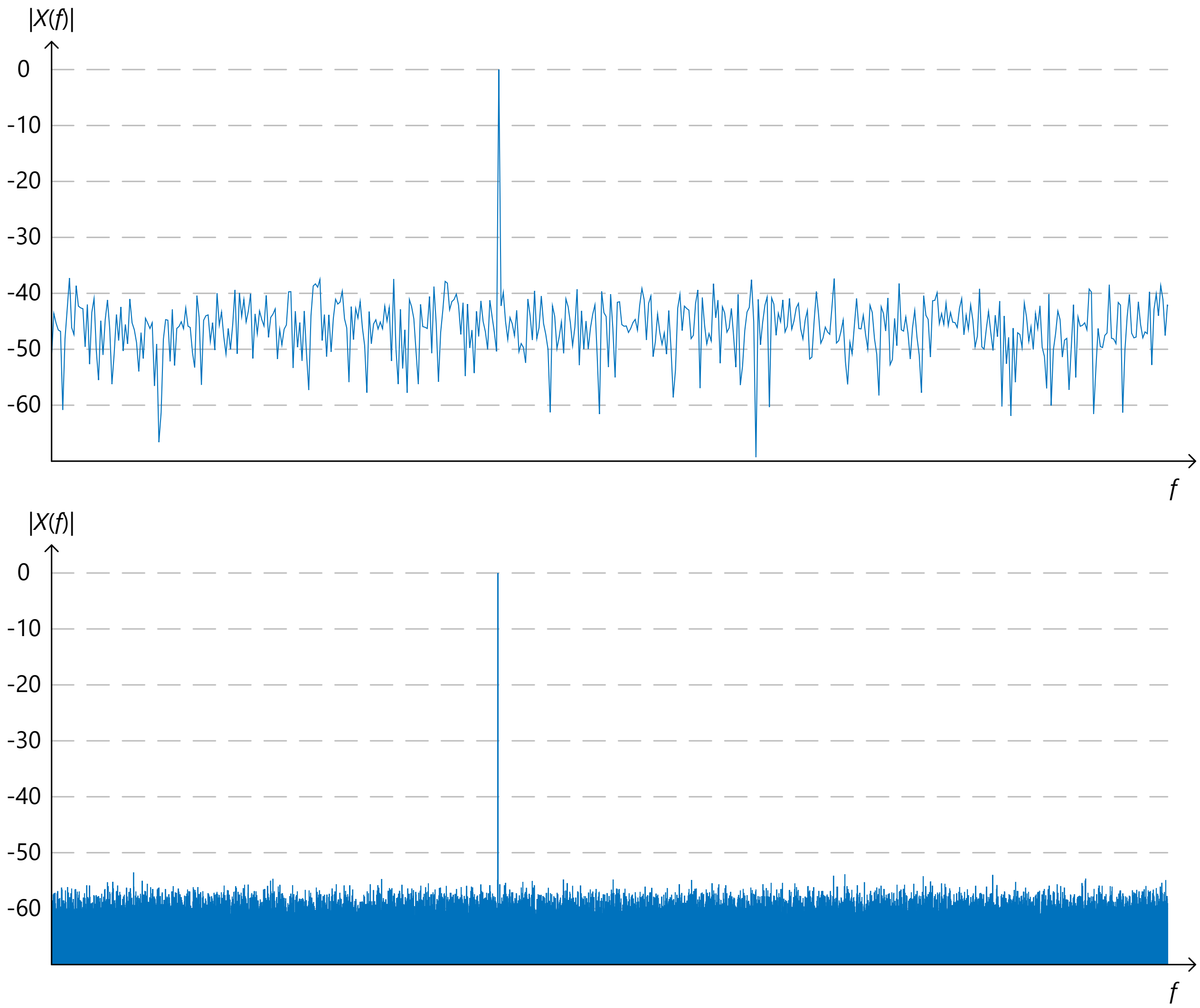



Figure 4.23 – Example of SNR enhancement for $N/N' = 100$

## §4.10 Conclusion

Here we conclude essential information about discrete sequence spectrum.

1) Relationship between periodicity and discreteness

| Signal | | Spectrum |
|---|---|---|
| Discrete | ↔ | Periodic |
| Periodic | ↔ | Discrete |

2) Relation between signal, spectrum and transformation tool

Set up a correspondence between type of the signal and type of the spectrum.

| | |
|---|---|
| IFT | Integral Fourier Transform |
| FS | Fourier Series |
| DTFT | Discrete-Time Fourier Transform |
| DFT | Discrete Fourier Transform |

| Signal | Periodic | Aperiodic | |
|---|---|---|---|
| Discrete | Periodic, discrete | Periodic, continuous | Spectrum |
| | DFT | DTFT | Tool |
| Continuous | Aperiodic, discrete | Aperiodic, continuous | Spectrum |

| | FS | IFT | Tool |
|---|---|---|---|

3) Spectrum enhancement FAQ

| | |
|---|---|
| - How to increase observable frequency band?<br>- Increase $f_s$. | - How to increase spectrum resolution?<br>- Increase $T$. |

4) Leakage essence

The cause of leakage is <u>finite</u> time of analysis

How to avoid or reduce the leakage?

1. Choose correct time of analysis;
2. Enhance resolution in frequency;
3. Use windows.

# Chapter 5 Fast Fourier Transform

## §5.1 Algorithm

### *5.1.1 Derivation*

The Fast Fourier Transform (FFT) is an algorithm of DFT calculation. This algorithm stands out for its high efficiency and low hardware costs. To achieve this, FFT introduces the following limitation on the number of samples *N*, i.e.

$$N = 2^l; l \in \mathbb{N}$$

Taking into account this limitation let's derivate the FFT algorithm.

$$X(m) = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi mn}{N}} = \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \cdot e^{-j\frac{2\pi m\cdot 2k}{N}} + \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} \cdot e^{-j\frac{2\pi m\cdot(2k+1)}{N}} = \left| W_N = e^{-j\frac{2\pi}{N}} \right|$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \cdot W_N^{2km} + W_N^m \cdot \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} \cdot W_N^{2km} = \left| (W_N)^2 = e^{-j\frac{2\pi}{N}\cdot 2} = e^{-j\frac{2\pi}{N/2}} = W_{\frac{N}{2}} \right|$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \cdot W_{\frac{N}{2}}^{km} + W_N^m \cdot \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} \cdot W_{\frac{N}{2}}^{km} = A(m) + W_N^m \cdot B(m).$$

Do the same operations for the second half of spectrum samples.

$$X\left(m + \frac{N}{2}\right) = \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \cdot W_{\frac{N}{2}}^{k\left(m+\frac{N}{2}\right)} + W_N^{m+\frac{N}{2}} \cdot \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} \cdot W_{\frac{N}{2}}^{k\left(m+\frac{N}{2}\right)}$$

$$= \left| \begin{array}{l} (W_N)^{m+N} = e^{-j\frac{2\pi}{N}\cdot(m+N)} = e^{-j\frac{2\pi m}{N}} \cdot e^{-j\frac{2\pi N}{N}} = W_N^m \cdot \underbrace{e^{-j2\pi}}_{1} = W_N^m \\ (W_N)^{m+\frac{N}{2}} = e^{-j\frac{2\pi}{N}\cdot\left(m+\frac{N}{2}\right)} = W_N^m \cdot e^{-j\frac{2\pi N}{N\cdot 2}} = W_N^m \cdot \underbrace{e^{-j\pi}}_{-1} = -W_N^m \end{array} \right|$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} \cdot W_{\frac{N}{2}}^{km} - W_N^m \cdot \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} \cdot W_{\frac{N}{2}}^{km} = A(m) - W_N^m \cdot B(m).$$

Summarizing it, we will get

$$X(m) = A(m) + W_N^m \cdot B(m)$$

$$X\left(m + \frac{N}{2}\right) = A(m) - W_N^m \cdot B(m)$$

### *5.1.2 Illustration of calculation flow*

Now let's illustrate the obtained result. For *N* = 8, it can be illustrated as in Figure 5.1.

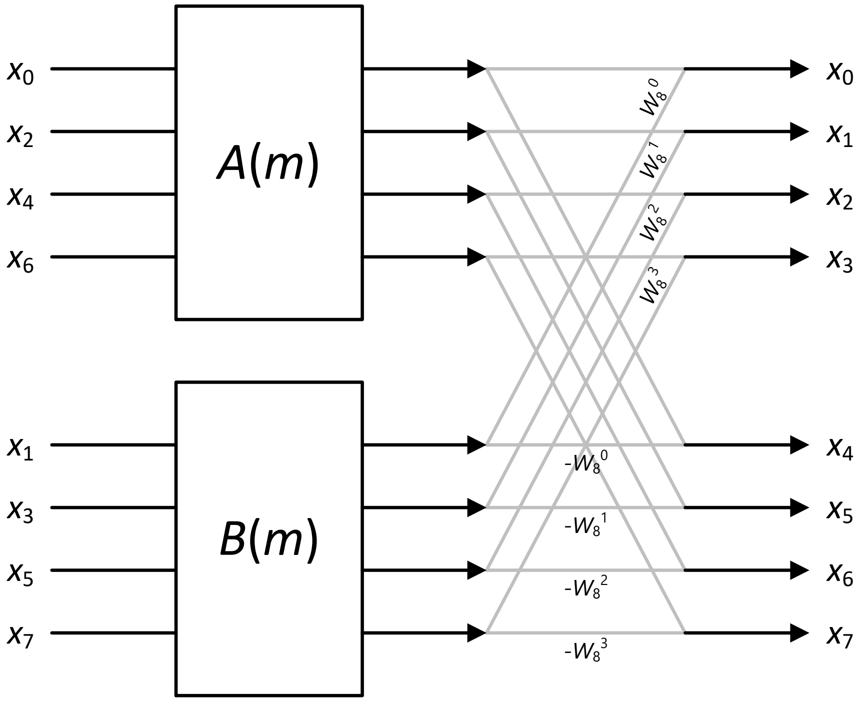



Figure 5.1 – $N$-point DFT calculation using two $N/2$-point DFTs

As number of samples is still even (thanks to the introduced limitation), we can continue splitting each DFT into even and odd samples. Finally, we stop the evaluation at 2-point DFT structure with coefficients

$$W_2^0 = W_N^0 = e^{-j\frac{2\pi}{N}\cdot 0} = 1; W_2^1 = W_N^{N/2} = e^{-j\frac{2\pi}{N}\cdot\frac{N}{2}} = e^{-j\pi} = -1$$

And whole calculation flow shown will be as in Figure 5.2.

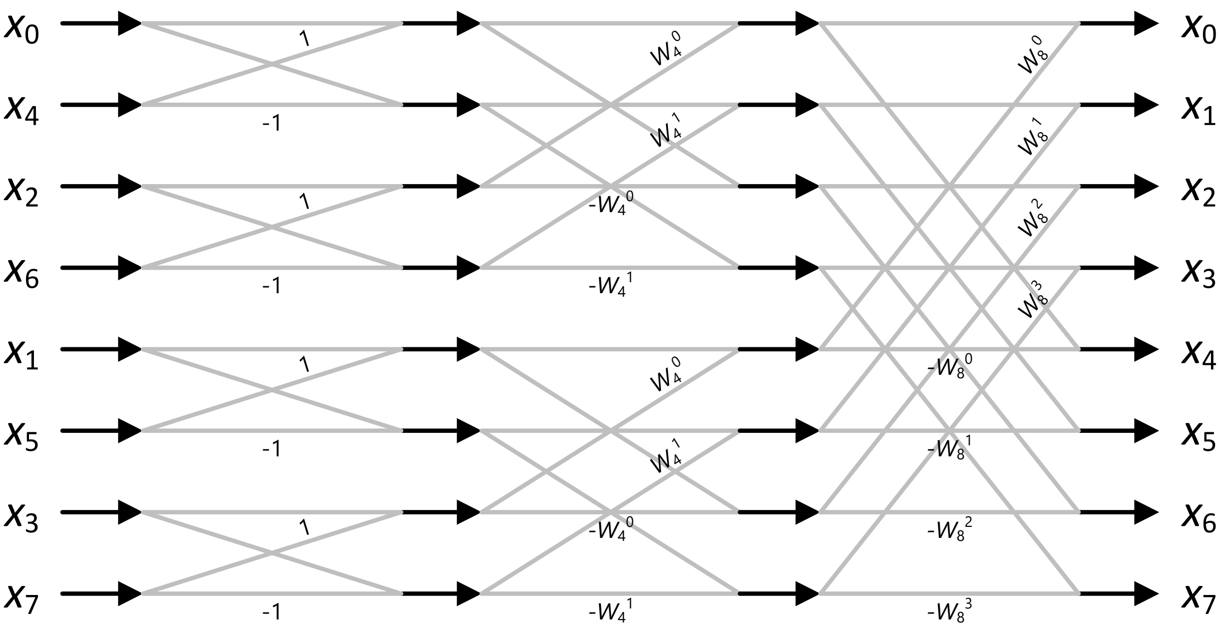



Figure 5.2 – 8-point Fast Fourier Transform calculation flow

### *5.1.3 Complexity of calculation*

We can ask a question: why do we need it? The answer is low number of multiplications and, as a consequence, higher efficiency of calculation. Expressions for the typical DFT flow and FFT algorithm are presented in the table below.

Number of complex multiplications $M$

| Conventional DFT | Fast Fourier Transform |
|---|---|
| $N^2$ | $\frac{N}{2}\cdot\log_2 N$ |

You can see in Figure 5.3 graphical representation of these expressions, where advantage of FFT is evident.

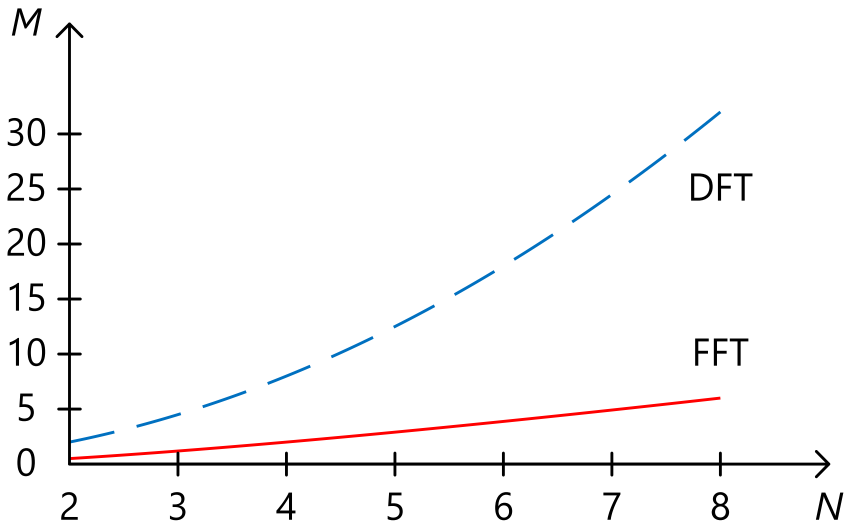



Figure 5.3 – Graphical representation of expressions for number of complex multiplications in DFT and FFT.

## §5.2 Bit-reversed order

Now we try to answer the question: how to get this magic order of input sequence indices. The way is the following:

1. Write index in ascending order from 0 to *N*-1;
2. Convert each index into the binary code;
3. Revert order of bits in each code;
4. Convert each code into decimal format.

In the table below, you can see an example of this way for *N* = 8, where *n* – initial order of samples, *n'* – bit-reversed order of samples.

| *n* | Binary code | Bit-reversed order | *n'* |
|---|---|---|---|
| 0 | 000 | 000 | 0 |
| 1 | 001 | 100 | 4 |
| 2 | 010 | 010 | 2 |
| 3 | 011 | 110 | 6 |
| 4 | 100 | 001 | 1 |
| 5 | 101 | 101 | 5 |
| 6 | 110 | 011 | 3 |
| 7 | 111 | 111 | 7 |

Bit-reversed order can be used not only for input samples, but also for the output samples. Let's take into account the fact that

$$W_{N\cdot k}^{m\cdot k} = e^{-j2\pi\frac{m\cdot k}{N\cdot k}} = e^{-j2\pi\frac{m}{N}} = W_N^m$$

Assuming for all coefficients *N* = 8, the FFT with regular order of input samples and bit-reversed order of the output samples will look like in Figure 5.4.

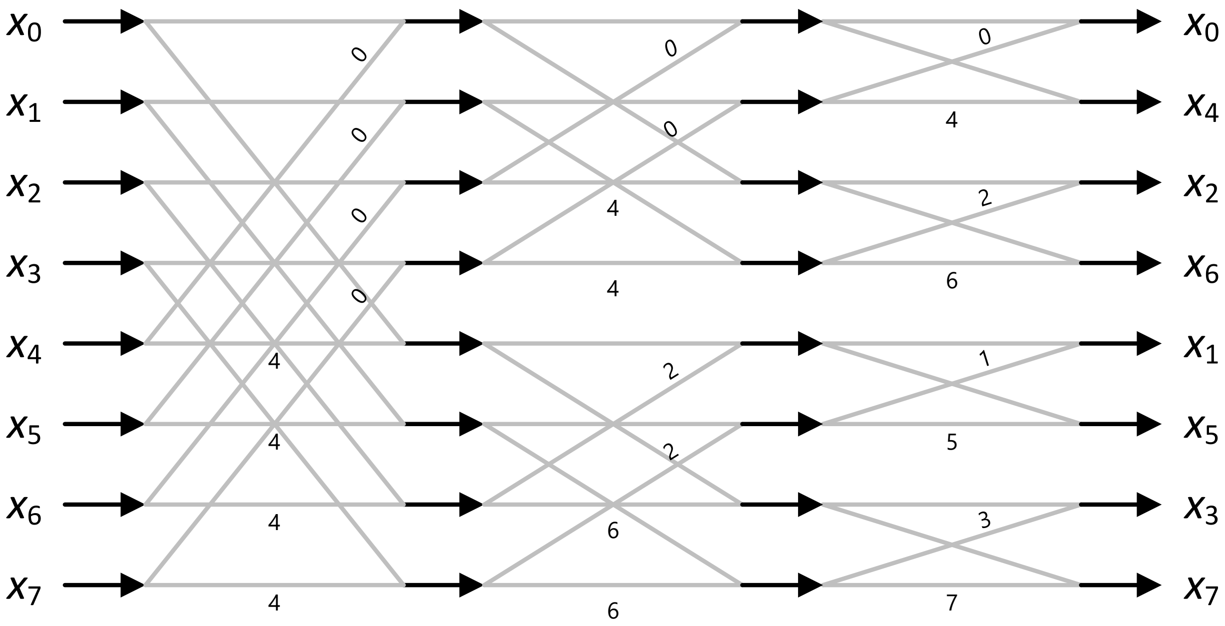


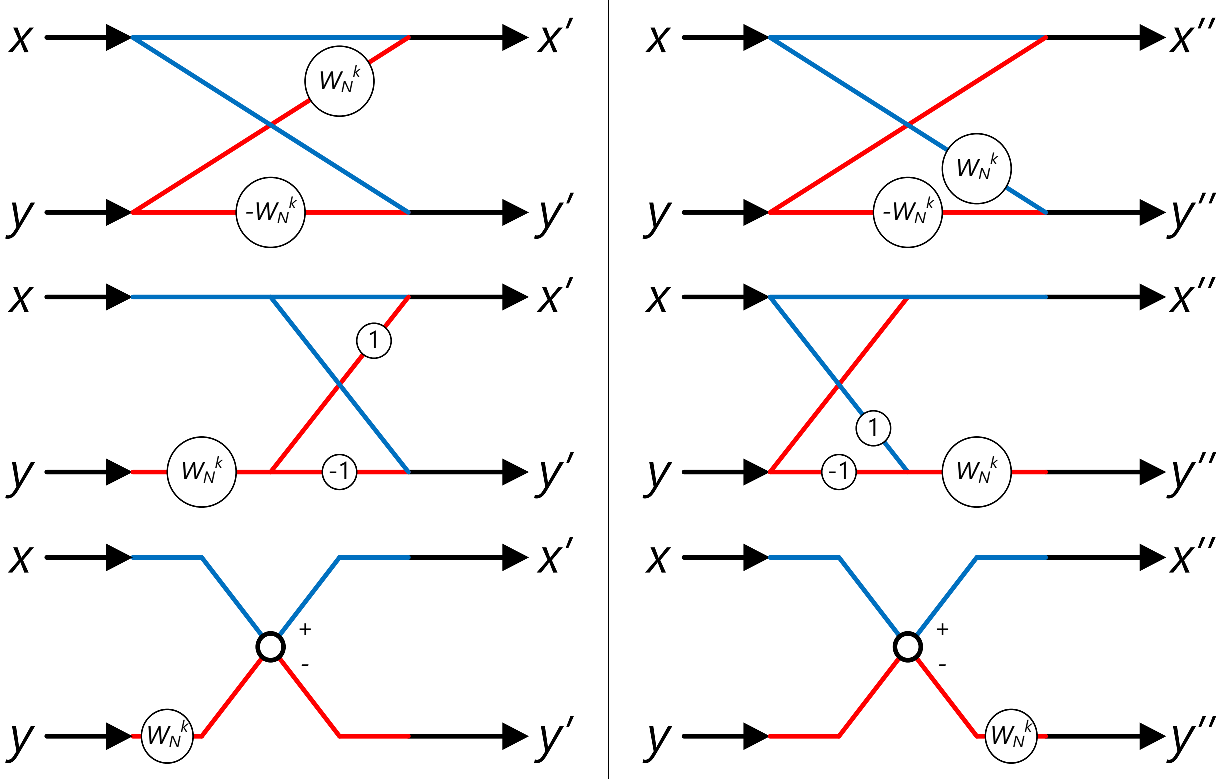

Figure 5.4 – The FFT flow for decimation-in-time and bit-reversed order of the output samples

During the derivation of the algorithm, we divided samples in time domain into even and odd. This approach is called “decimation-in-time”. However, the similar derivation can be done with a division of samples into even and odd in frequency domain – “decimation-in- frequency”.

## §5.3 Butterfly structures

There we will introduce different forms of butterfly structure.

| Decimation-in-time | Decimation-in-frequency |
|---|---|
| $x' = x + W_N^k \cdot y; y' = x - W_N^k \cdot y$ | $x'' = x + y; y'' = W_N^k \cdot (x - y)$ |




In the last figure, circle with plus and minus signs means branch with plus is a sum of branches on the left, branch with minus – a subtraction of branches on the left.

# Chapter 6 Finite impulse response filters

## §6.1 Introduction

At first, let's have a look at the example: some device that averages 5 samples of the input sequence. The data for the input *x* and output *y* is presented below (*n* – time index).

| *n* | *x* | *y* |
|---|---|---|
| 1 | 2 | 0.4 |
| 2 | 3 | 1 |
| 3 | 3 | 1.6 |
| 4 | 4 | 2.4 |
| 5 | 6 | 3.6 |
| 6 | 2 | 3.6 |
| 7 | 0 | 3 |
| 8 | 0 | 2.4 |
| 9 | 0 | 1.6 |
| 10 | 0 | 0.4 |
| 11 | 0 | 0 |

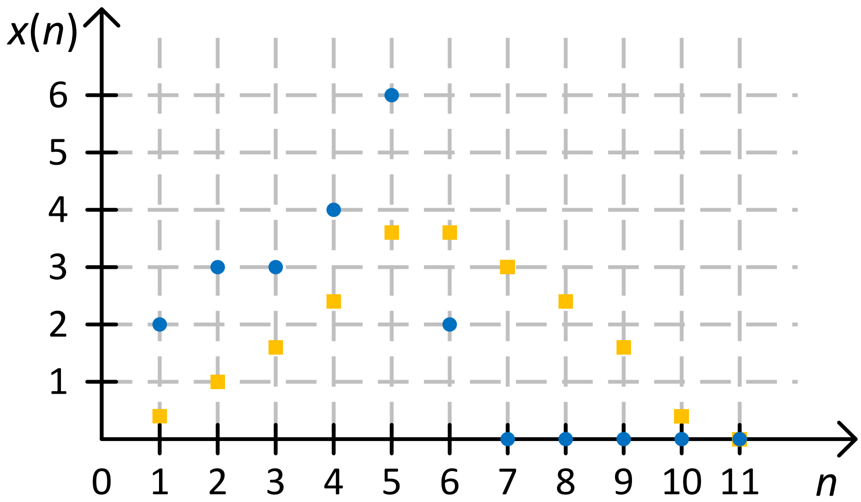



Figure 6.1 – Illustration for input (blue circle) and output (yellow square) signals

The equation for the output is

$$y(n) = \frac{x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4)}{5}$$

Structure of such a device can be presented as in Figure 6.2.

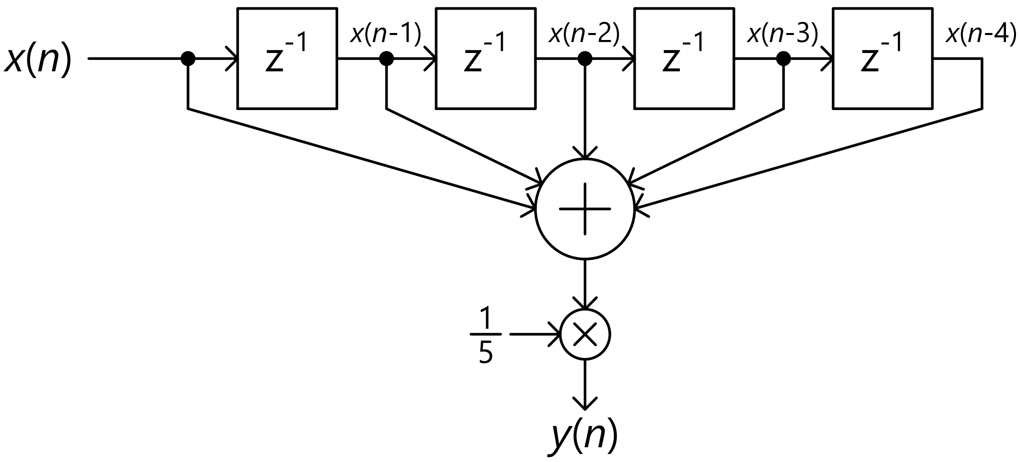


Figure 6.2 – Structure of the averaging device

Such a device can be generalized with the following expression for the output

$$y(n) = \sum_{k=0}^{K-1} a_k x(n-k).$$

and structure depicted in Figure 6.3. This structure is a Finite Impulse Response (FIR) filter.

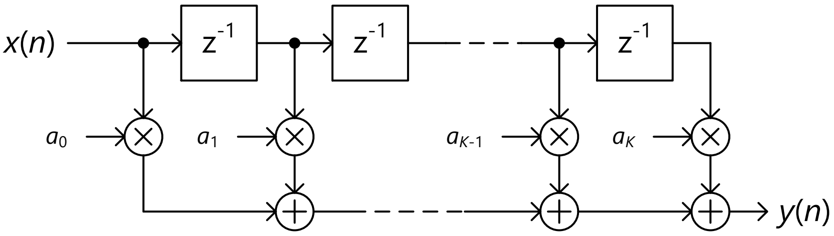



Figure 6.3 – Structure of a Finite Impulse Response (FIR) filter

From the presented expression the following conclusion can be drawn

1. The output of the filter is a convolution of the input $x(n)$ and the filter impulse response $h(k)$;
2. If the input samples represent delta function, then the output will be the filter coefficients sequence $a_k$;
3. The filter coefficients sequence $a_k$ of the FIR filter is impulse response $h(k)$;

Strictly speaking, convolution of the filter input and its impulse response is

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

However, $h(k)$ equals to 0 for $k$ that are not between 0 and $K$-1. So

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k) = \sum_{k=0}^{K-1} h(k)x(n-k).$$

The name Finite Impulse Response comes from the fact that the impulse response becomes 0 at a finite period of time, i.e.

$$\lim_{k\to K} h(k) = 0$$

where $K$ – number of coefficients, i.e. some finite number. As a result, the output of a FIR Filter will equal 0 in $K$ samples after input signal termination.

## §6.2 Filter analysis

For analog systems, we know that magnitude and impulse responses of a filter can be obtained from its transfer function $T(p)$ (reminder: $|T(p)|$ – magnitude response, arg $T(p)$ – phase response). And there is a connection between a transfer function and an impulse response:

$$T(p) = \mathcal{L}\{h(t)\}.$$

As a FIR filter is a discrete system, to get its "transfer function" we should use Z-transform. In §1.9 Z-transform was introduced as

$$H(z) = \mathcal{Z}\{h(n)\} = \sum_{n=-\infty}^{+\infty} h(n)z^{-n},$$

where $H(z)$ – a **transfer function** of a digital filter, $h(n)$ – an impulse response of the digital filter. The FIR filter impulse response is not equaled to zero only for $n$ = 0...$K$-1. So

$$H(z) = \sum_{n=-\infty}^{+\infty} h(n)z^{-n} = \sum_{n=0}^{K-1} h(n)z^{-n}$$

Since $h(n)$ is just filter coefficients, we can rewrite it as

$$H(z)=\sum_{n=0}^{K-1}h(n)z^{-n}=\sum_{n=0}^{K-1}a_n z^{-n}$$

The next step is to substitute *z* with

$$z\rightarrow e^{j\omega t_s}.$$

So we obtain

$$H\left(e^{j\omega t_s}\right)=\sum_{n=0}^{K-1}h(n)z^{-n}=\sum_{n=0}^{K-1}a_n e^{-jn\omega t_s}$$

And the last step is to get magnitude or phase of transfer function *H*(*z*) to plot corresponding responses

$$\left|H\left(e^{j\omega t_s}\right)\right|=\left|\sum_{n=0}^{K-1}a_n e^{-jn\omega t_s}\right|-\text{magnitude response}$$

$$\arg H\left(e^{j\omega t_s}\right)=\arg\sum_{n=0}^{K-1}a_n e^{-jn\omega t_s}\ -\text{phase response}$$

Example of magnitude and phase response for the filter from §6.1 are presented in Figure 6.4. The **order of a FIR filter** is the maximum power of $z^{-1}$ in the transfer function. It can be easily determined from the structure because the order is equal to the number of delays.

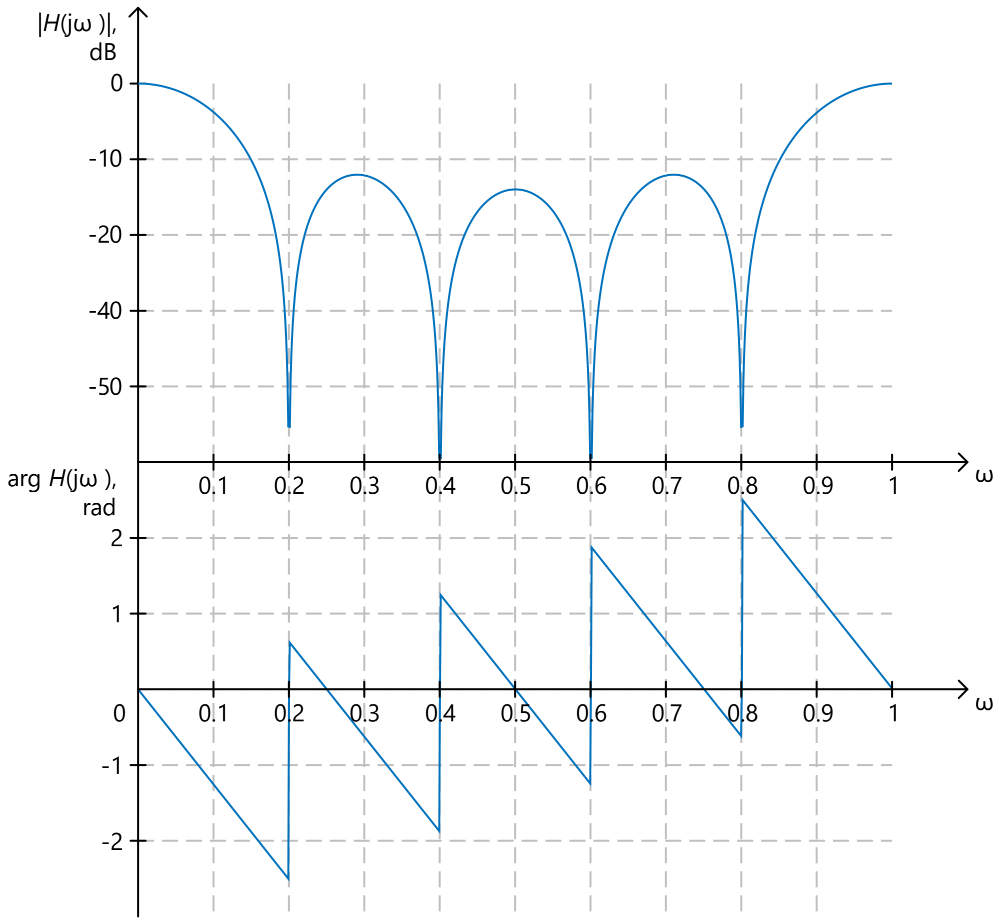



Figure 6.4 – Magnitude and phase responses for averaging filter.

## §6.3 Phase response

### *6.3.1 Introduction*

A phase response of a FIR filter will be linear in the pass-band if and only if coefficients (or impulse response) are symmetric or antisymmetric. That is a symmetric or antisymmetric impulse response is a necessary and sufficient condition for the FIR filter linear phase response in the pass-band.

### *6.3.2 Sufficiency condition*

Now look at sufficiency of this condition. Transfer function of a FIR filter is:

$$H(z) = \sum_{n=0}^{N-1} a_n z^{-n}$$

and we have 4 cases for combination of parity and symmetry. We illustrate this condition with case when impulse responses is symmetric and *N* is odd. In this case

$$a_n = a_{N-1-n}$$

And transfer function can be presented as:

$$H(z) = \sum_{n=0}^{N-1} a_n z^{-n} = \sum_{n=0}^{\frac{N-3}{2}} a_n z^{-n} + a_{\frac{N-1}{2}} z^{-\frac{N-1}{2}} + \sum_{n=\frac{N+1}{2}}^{N-1} a_n z^{-n}$$

Change for the last sum index:

$$n \to N-1-n$$

$$H(z) = \sum_{n=0}^{\frac{N-3}{2}} a_n z^{-n} + a_{\frac{N-1}{2}} z^{-\frac{N-1}{2}} + \sum_{n=\frac{N-3}{2}}^{0} a_{N-1-n} z^{-(N-1-n)} = \sum_{n=0}^{\frac{N-3}{2}} a_n z^{-n} + a_{\frac{N-1}{2}} z^{-\frac{N-1}{2}} + \sum_{n=0}^{\frac{N-3}{2}} a_n z^{-(N-1-n)} =$$

$$= \sum_{n=0}^{\frac{N-3}{2}} a_n \left(z^{-n} + z^{-(N-1-n)}\right) + a_{\frac{N-1}{2}} z^{-\frac{N-1}{2}} = z^{-\frac{N-1}{2}} \sum_{n=0}^{\frac{N-3}{2}} a_n \left(z^{-n+\frac{N-1}{2}} + z^{-(N-1-n)+\frac{N-1}{2}}\right) + a_{\frac{N-1}{2}} z^{-\frac{N-1}{2}} =$$

$$= z^{-\frac{N-1}{2}} \left( \sum_{n=0}^{\frac{N-3}{2}} a_n \left(z^{-\left(n-\frac{N-1}{2}\right)} + z^{n-\frac{N-1}{2}}\right) + a_{\frac{N-1}{2}} \right)$$

Transfer function is obtained by substitution:

$$H(z) \xrightarrow{z=e^{j\omega t_s}} T(\omega)$$

And finally we get:

$$T(\omega) = e^{-j\omega t_s \frac{N-1}{2}} \left( \sum_{n=0}^{\frac{N-3}{2}} a_n \left(e^{-j\omega t_s\left(n-\frac{N-1}{2}\right)} + e^{j\omega t_s\left(n-\frac{N-1}{2}\right)}\right) + a_{\frac{N-1}{2}} \right)$$

$$= e^{-j\omega t_s \frac{N-1}{2}} \left( \sum_{n=0}^{\frac{N-3}{2}} a_n \cdot 2\cos \omega t_s \left(n - \frac{N-1}{2}\right) + a_{\frac{N-1}{2}} \right)$$

Thus, we can determine frequency response and phase response:

$$|T(\omega)| = \left| 2 \sum_{n=0}^{\frac{N-3}{2}} a_n \cdot \cos \omega t_s \left(n - \frac{N-1}{2}\right) + a_{\frac{N-1}{2}} \right|$$

$$\arg T(\omega) = -\omega t_s \frac{N-1}{2}$$

As we can see, phase response is a linear function of frequency ω.

▲ Home exercise: proof the rest 3 cases.

Finally, you will get the following result

| | Odd | Even |
|---|---|---|
| Symmetric | $\|T(\omega)\| = \left\| 2 \sum_{n=0}^{\frac{N-3}{2}} a_n \cdot \cos \omega t_s \left(n - \frac{N-1}{2}\right) + a_{\frac{N-1}{2}} \right\|$ | $\|T(\omega)\| = \left\| 2 \sum_{n=0}^{\frac{N}{2}-1} a_n \cdot \cos \omega t_s \left(n - \frac{N-1}{2}\right) \right\|$ |

| | | |
|---|---|---|
| | $\arg T(\omega) = -\omega t_s \frac{N-1}{2}$ | $\arg T(\omega) = -\omega t_s \frac{N-1}{2}$ |
| Antisymmetric | $\lvert T(\omega)\rvert = \left\lvert 2\sum_{n=0}^{\frac{N-3}{2}} a_n \cdot \sin \omega t_s \left(n - \frac{N-1}{2}\right)\right\rvert$ <br> $\arg T(\omega) = -\frac{\pi}{2} - \omega t_s \frac{N-1}{2}$ | $\lvert T(\omega)\rvert = \left\lvert 2\sum_{n=0}^{\frac{N}{2}-1} a_n \cdot \sin \omega t_s \left(n - \frac{N-1}{2}\right)\right\rvert$ <br> $\arg T(\omega) = -\frac{\pi}{2} - \omega t_s \frac{N-1}{2}$ |

### *6.3.3 Conclusion*

We have discussed all cases for a symmetric impulse response. A linear phase response means that group delay is a constant of frequency. Group delay *G* is evaluated by:

$$G = -\frac{d\varphi}{d\omega}$$

And, for all discussed cases, a group delay is:

$$G = -\frac{d\varphi}{d\omega} = t_s \frac{N-1}{2}$$

It is a constant of frequency, so that the phase response of a FIR filter is linear.

Also let's note that at points

$$\omega = \pm\frac{\omega_s}{2}$$

the argument of trigonometric function in the magnitude response becomes

$$\theta = \omega t_s \left(n - \frac{N-1}{2}\right) = \pm 2\pi \frac{f_s}{2} t_s \left(n - \frac{N-1}{2}\right) = \pm\pi \left(n - \frac{N-1}{2}\right)$$

For odd and even *N,* we can rewrite phase φ, correspondingly, as

$$\theta_{odd} = \pi \frac{2k}{2} = \pi k;\ \theta_{even} = \pi \left(\frac{2k+1}{2}\right) = \pi k + \frac{\pi}{2}$$

Then we get that

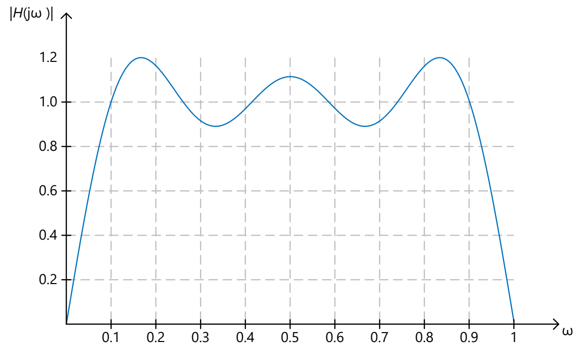
$$\cos \theta_{even} = \cos\left(\pi k + \frac{\pi}{2}\right) = 0; \sin\theta_{odd} = \sin \pi k = 0$$

Thus, for symmetric-even and antisymmetric-odd cases:

$$\left\lvert T\left(\pm\frac{\omega_s}{2}\right)\right\rvert = 0$$

| | Odd | Even |
|---|---|---|
| Symmetric | $\left\lvert T\left(\pm\frac{\omega_s}{2}\right)\right\rvert \neq 0$ | $\left\lvert T\left(\pm\frac{\omega_s}{2}\right)\right\rvert = 0$ |
| Antisymmetric | $\left\lvert T\left(\pm\frac{\omega_s}{2}\right)\right\rvert = 0$ | $\left\lvert T\left(\pm\frac{\omega_s}{2}\right)\right\rvert \neq 0$ |

Examples of magnitude responses illustrating such a property are presented in Figure 6.5.



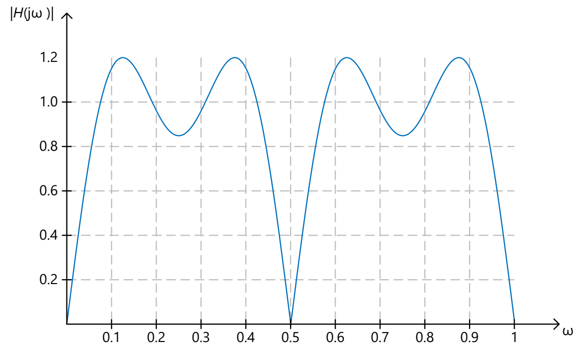




Figure 6.5 – Examples of magnitude response for antisymmetric impulse response with even (left) and odd (right) number of samples

## §6.4 Structures

### *6.4.1 Direct Forms*

In Figure 6.5, you can see already known Direct Form of a FIR filter. It has a critical path that is shown with red color. The critical path – the longest path for signal passing between two registers (delay elements). The critical path in Direct form has 1 multiplication and *K* adders (*K* – a filter order);

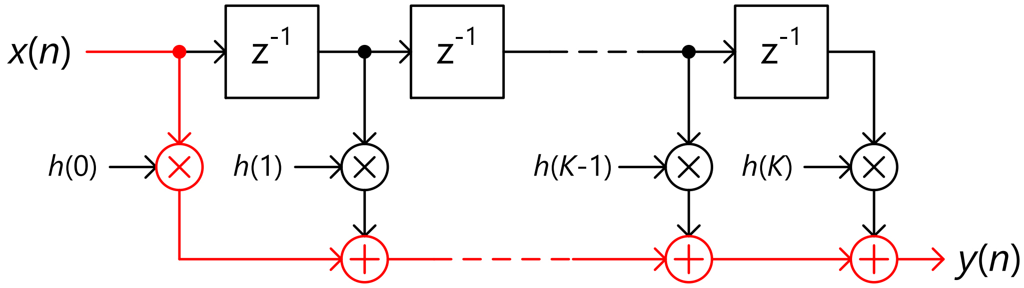



Figure 6.6 – Direct Form of a FIR filter

### *6.4.2 Transposed forms*

Transposed Form can be obtained from the Direct Form by the following operations:

- Nodes are replaced by adders;
- Adders are replaced by nodes;
- Arrows changes its direction on opposite.

The results of the transposition is shown in Figure 6.6. The critical path of the transposed form has only 1 multiplication and 1 adder regardless of the filter order.

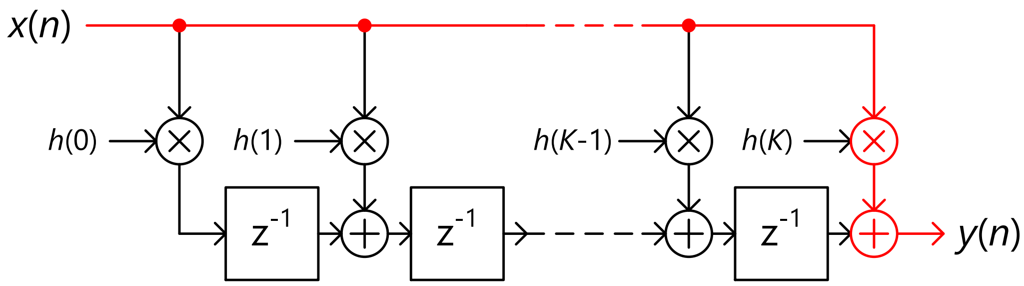



Figure 6.7 – Transposed Form of a FIR filter

### *6.4.3 Folded Form*

A FIR filter structure can be optimized in case of symmetric or anti-symmetric coefficients. As there are identical coefficients multiplication it can be done only once. Let's have a look at expression for the filter output

$$y(n)=\sum_{k=0}^{N-1}h(k)\cdot x(n-k)=\sum_{k=0}^{\frac{N-3}{2}}h(k)\cdot x(n-k)+h\left(\frac{N-1}{2}\right)+\sum_{k=\frac{N+1}{2}}^{N-1}h(k)\cdot x(n-k)=\begin{vmatrix}k=N-1-p\\p=N-1-k\end{vmatrix}$$
$$=\sum_{k=0}^{\frac{N-3}{2}}h(k)\cdot x(n-k)+h\left(\frac{N-1}{2}\right)+\sum_{p=0}^{\frac{N-3}{2}}\underbrace{h(N-1-p)}_{h(p)}\cdot x\big(n-(N-1-k)\big)$$
$$=\sum_{k=0}^{\frac{N-3}{2}}h(k)\cdot\Big(x(n-k)+x\big(n-(N-1-k)\big)\Big)+h\left(\frac{N-1}{2}\right)=\left|K=\frac{N-1}{2}\right|$$
$$=\sum_{k=0}^{K-1}h(k)\cdot\Big(x(n-k)+x\big(n-(2K-k)\big)\Big)+h(K).$$

There we see that corresponding samples can be summed up before multiplication instead of individual multiplication. Such a simplification reduces required number of multipliers and, consequently, hardware costs. Corresponding filter structure is called “folded” and is illustrated in Figure 6.7.

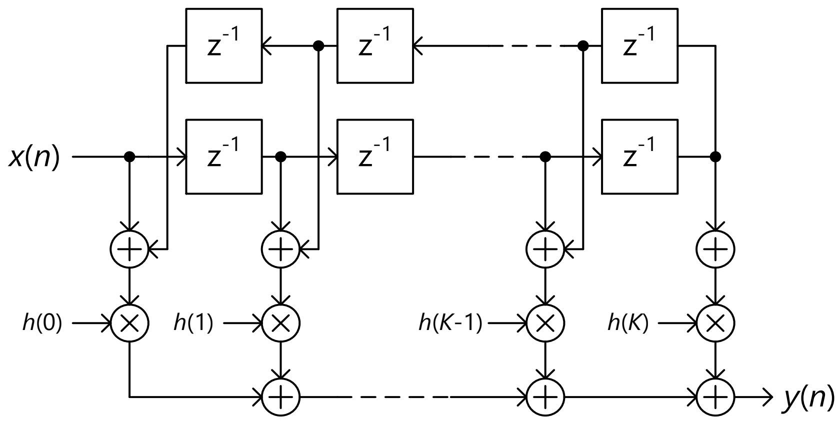



Figure 6.8 – Folded Form of a FIR filter with symmetric coefficients

## §6.5 Half-band filters

Half-band filter is a specialized FIR filter, whose magnitude response is symmetric relative to point (0.5; 0.25×$\omega_s$). An advantage of such a filter is that impulse response has every second filter coefficient being zero, except the center one. This allows us to reduce number of multiplications approximately by 2 times. Such filters are popular in sample rate conversion applications like decimation or interpolation. An example of an impulse response for the 10-th order half-band filter is presented in Figure 6.8. Its magnitude response is shown in Figure 6.9.

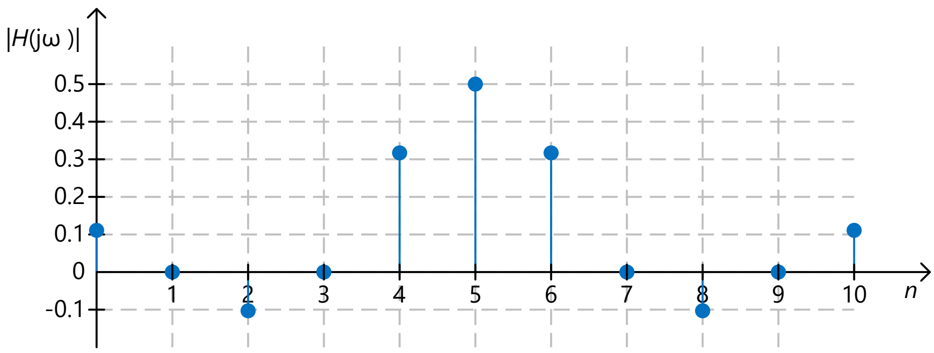



Figure 6.9 – An impulse response of a half-band filter

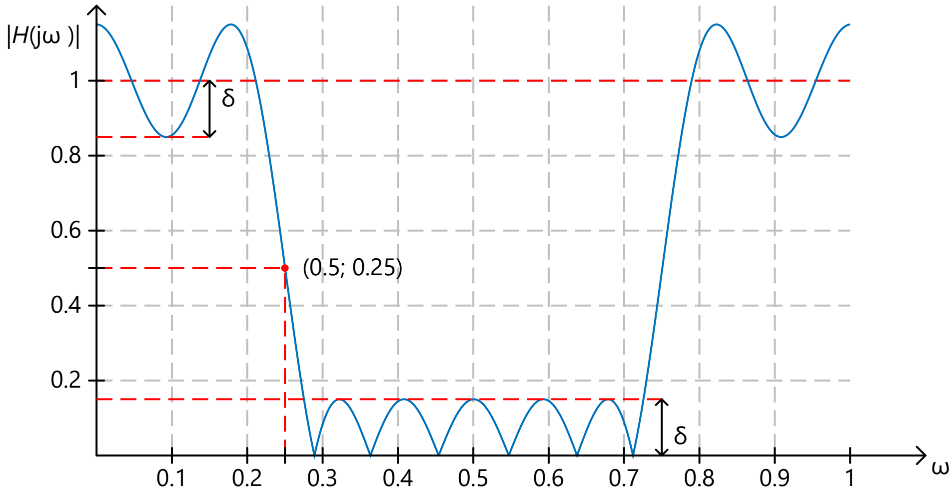



Figure 6.10 – Magnitude response of 10 order Half-band filter

The number of multipliers *M* in the folded form of a half-band filter can be found as

$$M = \frac{s+1}{2} + 1,$$

where *s* – the number of branches. If we derive the number of branches from the filter order *N*, it will be

$$s = N + 1.$$

So *M* can be calculated through filter order as

$$M = \frac{N}{2} + 2.$$

Additionally, structure can be folded like in Figure 6.10 for further multiplier reduction.

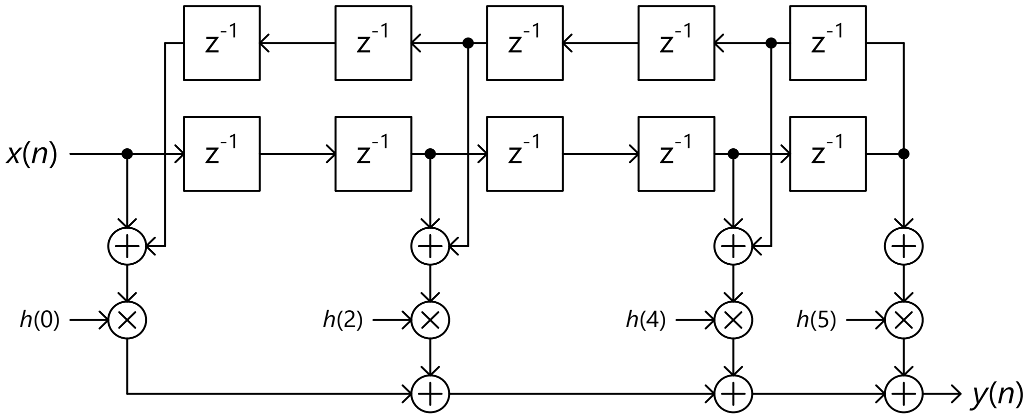



Figure 6.11 – Folded form of the 10-th order half-band filter

# Chapter 7 Infinite impulse response filters

## §7.1 Introduction

In the previous chapter, we discussed filters that have a finite length of an impulse response. Now, we go forward and take into consideration a case when the impulse has an unlimited length, i.e. infinite impulse response. To achieve this feature, it is required to introduce feedback into a filter. Let's do this with ordinary FIR structure. The transformations of FIR filter structure are presented in Figure 7.1. The last structure is an infinite impulse response filter (IIR filter).

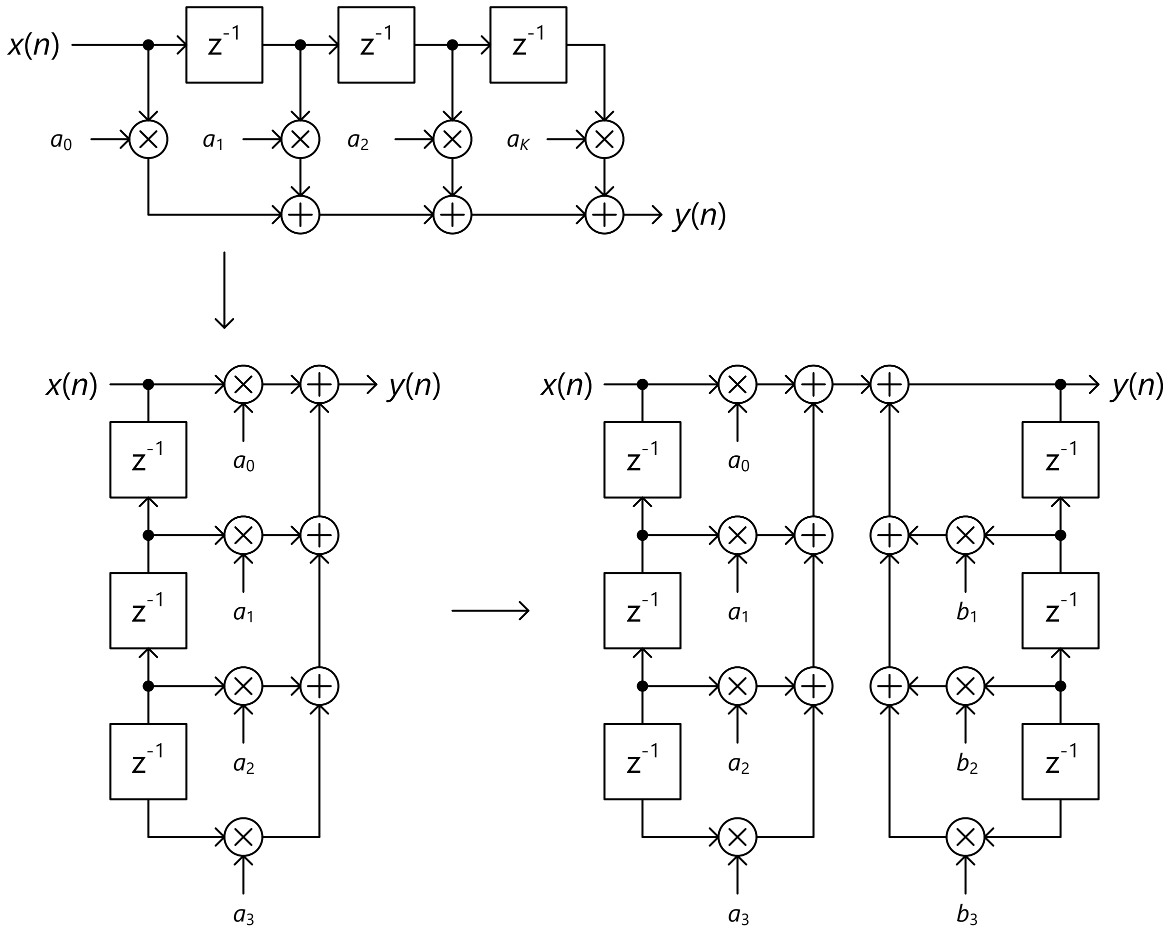



Figure 7.1 – An introduction of a feedback into filter structure

Now, we try to write an expression for such a filter output

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) + a_3x(n-3) + b_1y(n-1) + b_2y(n-2) + b_3y(n-3)$$
$$= \sum_{k=0}^{3} a_kx(n-k) + \sum_{k=1}^{3} b_ky(n-k)$$

And in a general case, we have for IIR filter

$$y(n) = \sum_{k=0}^{N} a_kx(n-k) + \sum_{k=1}^{M} b_ky(n-k)$$

From this expression, we can conclude that IIR filter is also a linear-time-invariant system. So the output of IIR filter can be expressed with its impulse response and convolution operation.

$$y(n) = \sum_{k=0}^{+\infty} h(k)x(n-k)$$

However, impulse response is not just a sequence of coefficients like in FIR filter and reaches zero value at infinity, i.e.

$$\lim_{n\to+\infty} h(n) = 0$$

So unlike FIR filter the upper limit of convolution sum cannot be replaced with a finite number.

Why are IIR filters of interest? Such filters have significantly higher slope and attenuation for the same filter order. Figure 7.2 shows magnitude response of FIR and IIR filter having the 5th order. As a result IIR filters demand less hardware costs than FIR filter for the certain response requirements.

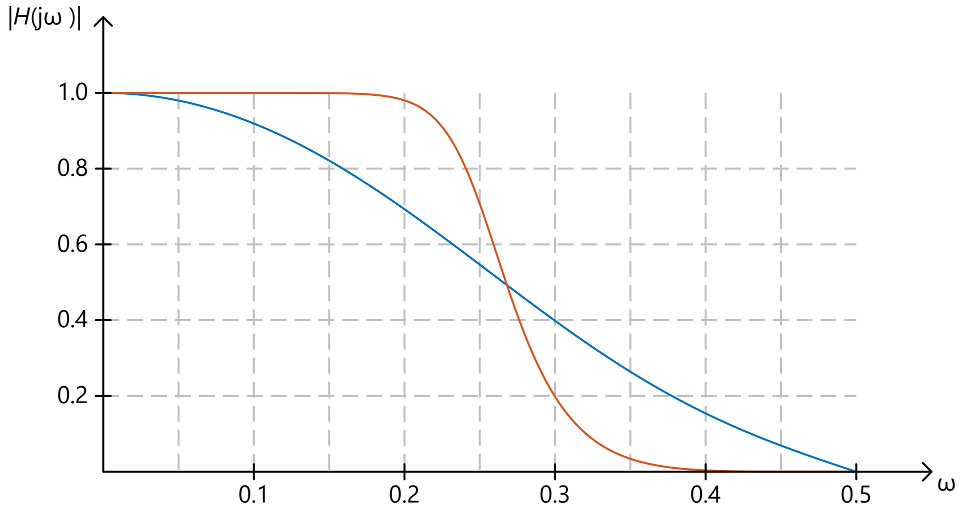



Figure 7.2 – Magnitude response of the 5th order FIR (blue) and IIR (red) filters

## §7.2 Filter analysis

As impulse response of IIR is not directly expressed with filter coefficients, to get the transfer function of IIR we need to make Z-transform for the output expression. In general case, the output expression is

$$y(n) = \sum_{k=0}^{N} a_k x(n-k) + \sum_{k=1}^{M} b_k y(n-k)$$

So its Z-transform is

$$Y(z) = \mathcal{Z}\{y(n)\} = \sum_{k=0}^{N} a_k X(z) z^{-k} + \sum_{k=1}^{M} b_k Y(z) z^{-k}.$$

We know that transfer function is defined as

$$H(z) = \frac{Y(z)}{X(z)}$$

So we can rewrite the expression for the output as

$$Y(z)\left(1 - \sum_{k=1}^{M} b_k z^{-k}\right) = X(z) \sum_{k=0}^{N} a_k z^{-k}$$

and derive the transfer function as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} a_k z^{-k}}{1 - \sum_{k=1}^{M} b_k z^{-k}}.$$

The order for an IIR filter is defined also as in §6.2, i.e. the maximum value of $M$ and $N$.

From the transfer function, magnitude and phase responses are obtained in the same way as for FIR filters. In terms of phase response linearity, IIR filter cannot provide strict linear dependence in the pass-band under no circumstances. It results from the filter and transfer function structures: denominator always has leading coefficient that equals 1, which prevents conversion of the fraction to a single trigonometric function.

## §7.3 Stability

Unlike FIR, IIR filters has denominator that can turn to 0. In other words, transfer function has poles and system may become unstable. As we have discussed in 1.9.2, the left half-plane of $p$-plane is transformed into a unit circle in $z$-plane. Stable IIR filter has all poles inside this unit circle. A presence of at least one pole outside the unit circle means that IIR filter is unstable and can become a generator. An illustration of poles location and systems stability is presented in Figure 7.3. Stability of IIR filter should be guaranteed by a filter designer with an appropriate choice of transfer function and its conversion during hardware implementation.

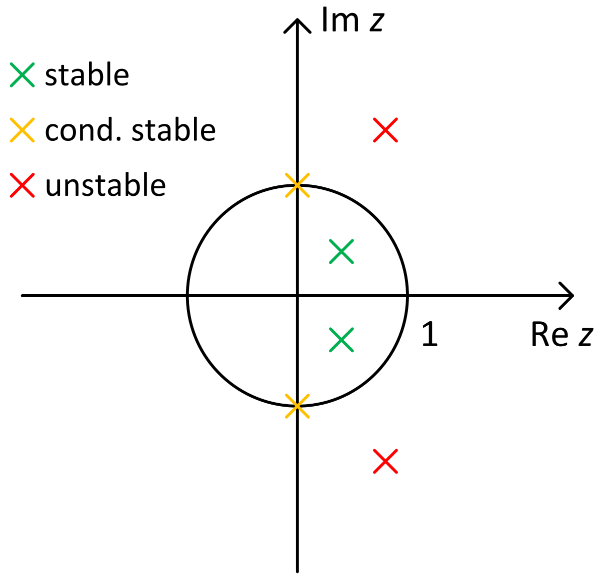



Figure 7.3 – Magnitude response of the 5th order FIR (blue) and IIR (red) filters

## §7.4 Structures

### *7.4.1 General considerations*

A structure obtained in §7.1 is a Direct Form I of a IIR filter. Beside this form, a complementary one exists. Knowing that IIR filter is a LTI system, we can employ commutativity property and swap its parts. The final structure will be equivalent to the original structure and is called Direct Form II. Both forms are depicted in Figure 7.4.

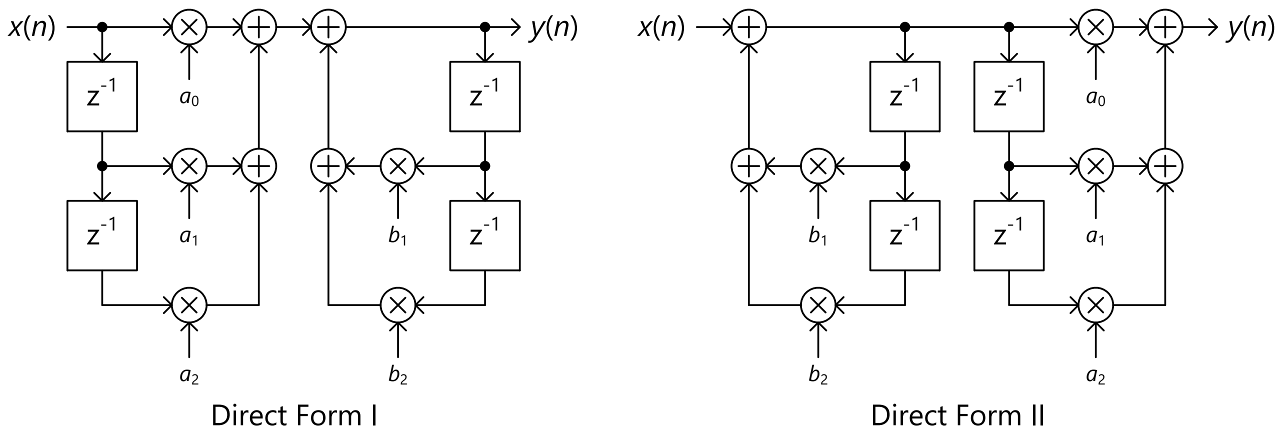
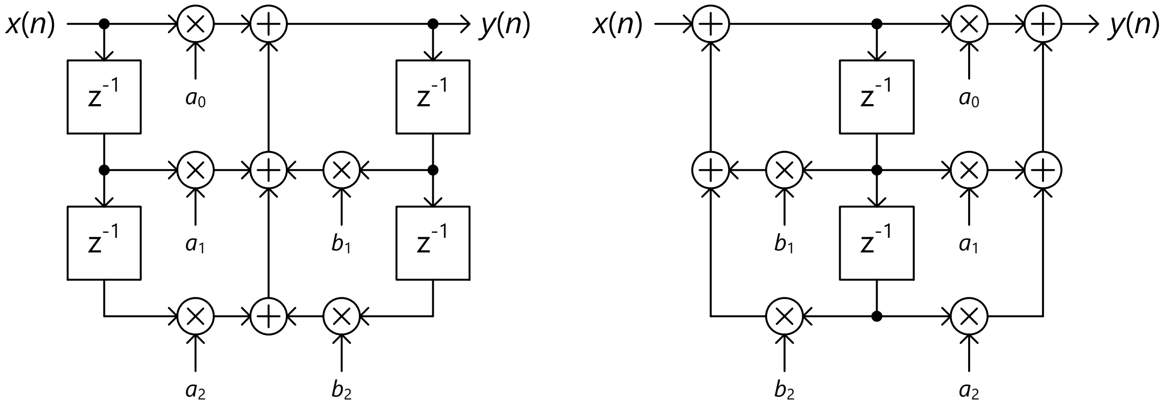



Figure 7.4 – Direct Forms of a IIR filter

Both forms can be simplified by combining the similar blocks into one. Simplified structures are shown in Figure 7.5. In Direct Form I, adders can be combined together. However, this simplification affect only the structure presentation as real adders have only two inputs and the total number of real adders will be the same. In terms of Direct Form II, simplification matters as it reduces number of delay units almost by 2 times. The Direct Form II is called the canonical form because it uses the minimal number of delay units, adders and multipliers.



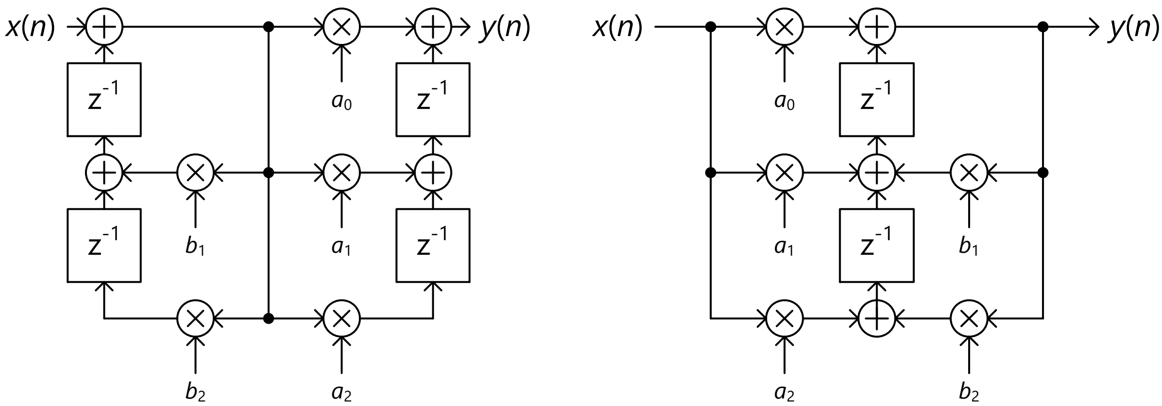

Simplified Direct Form I Simplified Direct Form II

Figure 7.5 – Simplified Direct Forms of a IIR filter

In addition to the Direct Forms, there are Transposed Forms. Transposition can be done by the following algorithm:

1. Replace nodes with adders;
2. Replace adder with nodes;
3. Revert arrows direction.

Transposed forms are illustrated in Figure 7.6.




Transposed Direct Form I Transposed Direct Form II

Figure 7.6 – Transposed Direct Forms of a IIR filter

*7.4.2 Implementation issues*

Consider all mentioned forms from an implementation point of view. Take into account length of critical path and necessity of a quantization (rounding) block $Q(z)$. Both issues are highlighted in Figure 7.7. Red line represents possible critical path. A quantization block is obligatory required at the output and in the beginning of the feedback path. The output quantization block provides the required resolution of the output samples since intermediate resolution is larger due to precision loss prevention measures. The quantization block in the beginning of the feedback path limits growth of the resolution in the feedback loop.

If some operations are done between the mentioned quantization blocks and delay unit, an additional quantization block can be inserted at delay unit input (dashed blocks). Such a quantization is optional as it can be interchanged with a resolution increase of the following delay unit. Profit of this interchange should be assessed in accordance with a system application.

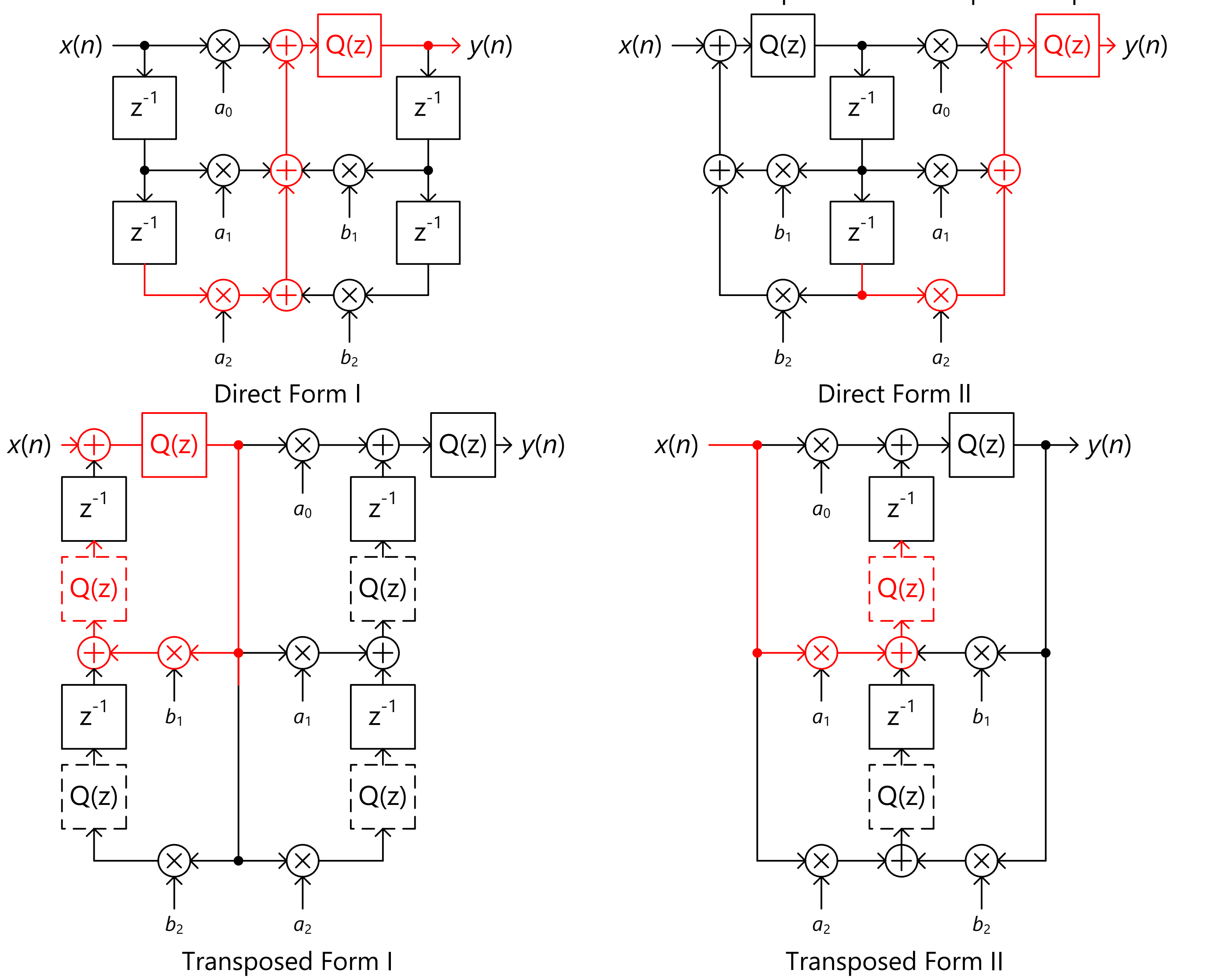



Figure 7.7 – Critical path and quantization blocks in different structures

Let's summarize pros and cons of each form in terms of implementation

| | Form I | Form II |
|---|---|---|
| Direct | • Large number of delay units;<br>• Critical path grows with order;<br>• The minimal number of roundings;<br>• Delay units with the minimal resolution; | • Small number of delay units;<br>• Critical path grows with order;<br>• 2 roundings;<br>• Delay units with a moderate resolution; |
| Transposed | • Large number of delay units;<br>• Critical path does not depend on order;<br>• 2 roundings;<br>• Delay units with a growing resolution (may be interchanged with rounding); | • Small number of delay units;<br>• Critical path does not depend on order;<br>• The minimal number of roundings;<br>• Delay units with a growing resolution (may be interchanged with rounding); |

Fixed point DSP usually prefers the non-transposed forms. The main limitation in such processing is a precision loss, so number of rounding should be minimized. Direct Forms has no more than 2 rounding, and they do not provoke an increase in resolution in delay units. These factors allow to obtain very resource efficient implementations. However, in high-speed applications systems incline to use transposed forms due to their minimal critical path. Floating point DSP usually prefers the transposed forms (especially, canonical form). In floating point calculations, precision loss is not a critical issue. So benefits of transposed forms, like shorter critical path, can be adopted without noticeable losses.

## §7.5 Pitfalls in IIR filter realization

Let's imagine that you design some filter. It perfectly fits the requirements. However, due to incorrect implementation it may have other parameters or even become a generator. Why may it happen? You have not taken into account finite resolution of operation blocks. Which type of error should be taken into account?

1. Coefficient quantization;
2. Overflow and underflow;
3. Rounding.

Each filter coefficient can be stored with finite resolution. Lack of resolution leads to an error in a coefficient presentation and changes in transfer function. So filter parameters deviate from the nominal values. Such deviation may translate into a shift of poles outside unit circle making filter unstable.

Overflow and underflow typically relates to adder and multipliers, which cannot present operation result with sufficient precision. Overflow means that resulting number is larger than possible for given resolution. Underflow means that resulting number is smaller than possible for presentation.

Rounding may be a part of adder or multiplier or be a standalone block. Rounding limits sample resolution and explicitly introduces a calculation error. In correctly designed systems, the introduced error does not affect the system stability and the output accuracy. Does FIR filter have the same problems? Yes. However, due to an absence of the feedback, they are not so critical.

## §7.6 Cascaded design

One of possible solution to mitigate the above issues of finite resolution is a cascaded design. At first, let's remember how systems can be combined. Figure 7.8 illustrates two possible connections: parallel and serial. These options are expressed as

$$H_{parallel}(z) = H_1(z) + H_2(z) \text{ and } H_{serial}(z) = H_1(z) \times H_2(z).$$

In this paragraph, we are interested in the last one – serial connection.

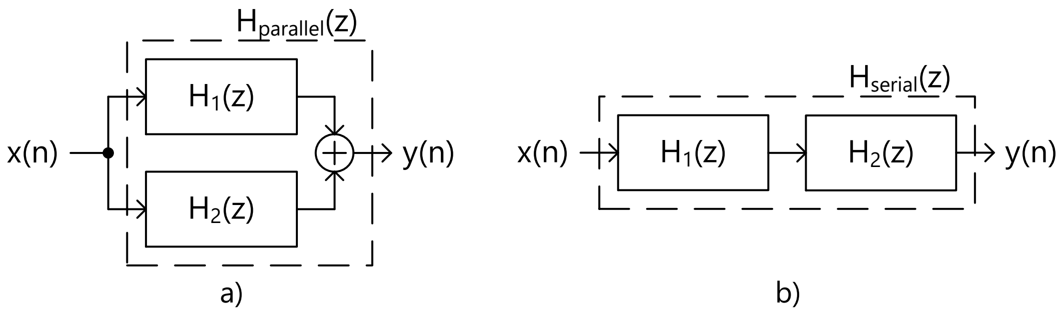



Figure 7.8 – Parallel (a) and serial (b) connection of systems

The main point of the cascaded design is to split a high-order system into a serial connection of low-order systems. This motivation origins from the fact that high-order systems impose high requirements on the accuracy of coefficients and calculations. So a transition to low-order systems allows to relax requirements and simplify design process. The key issue in this transition is a factorization of the target transfer function. Such a factorization employs second–order sections (SOS). The second order sections are preferred to the first order because the latter one can lead factorization to complex-valued coefficients that are less convenient for implementation. The second order always has real-valued coefficients. As a result, the target transfer function *H*(*z*) is presented as

$$H(z) = \prod_{i=1}^{N} H_i(z) = \prod_{i=1}^{N} \frac{a_{i0} + a_{i1}z^{-1} + a_{i2}z^{-2}}{1 + b_{i1}z^{-1} + b_{i2}z^{-2}}$$

where $H_i(z)$ – a transfer function of *i*-th SOS, $a_{ij}$ and $b_{ij}$ – coefficients of SOS. Structure of cascaded design is depicted in Figure 7.9.

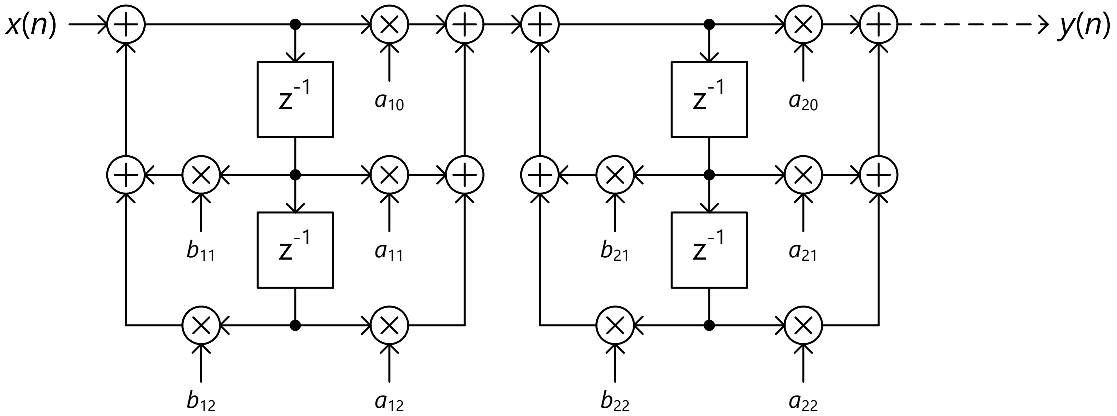



Figure 7.9 – Cascaded filter based on SOS

However, the following issue should be taken into account. As transfer functions are multiplied, ripples in the pass-band increase. Let's illustrate this and take transfer function in the pass-band as

$$H(z) = 1 + R.$$

If we serially put blocks with such a transfer function, then

$$H(z) \times H(z) = (1 + R)(1 + R) = 1 + 2R + R^2 \approx 1 + 2R + o(R^2).$$

for $R << 1$. From there, it can be seen that ripples increase up to 2 times. So each SOS has stricter requirements for ripples than the target transfer function.

## §7.7 Matrix form

In this paragraph, we discuss how to organize calculations of the output samples for IIR filter using matrix forms and operations. Have a look at an example. Consider filter structure depicted in Figure 7.10.

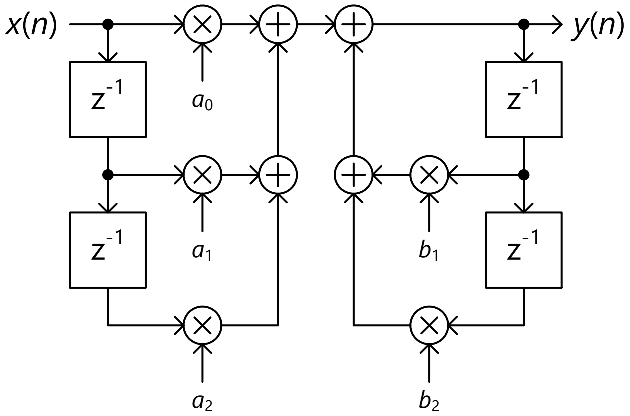



Figure 7.10 –

It is described by the following expression:

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + b_1 y_{n-1} + b_2 y_{n-2}$$

Let $b_0 = 0$, then

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + b_0 y_n + b_1 y_{n-1} + b_2 y_{n-2}$$

Now, introduce the input and the output vectors

$$\bar{x} = \begin{bmatrix} x_n \\ x_{n-1} \\ x_{n-2} \end{bmatrix}, \bar{y} = \begin{bmatrix} y_n \\ y_{n-1} \\ y_{n-2} \end{bmatrix}$$

and coefficient vectors

$$A = [a_0 \quad a_1 \quad a_2], B = [b_0 \quad b_1 \quad b_2] = [0 \quad b_1 \quad b_2]$$

Then the filter expression can be rewritten as

$$y_n = A\bar{x} + B\bar{y}.$$

We remember that on every clock cycle

$$y_i \to y_{i-1}.$$

That is, in our example it will be as

$$y_n \to y_{n-1}, y_{n-1} \to y_{n-2}$$

If we introduce 2 output vectors: $\overline{y'}$ (new output vector) and $\bar{y}$ (current output vector), then we can take it into account by the following expanding of $A$ and $B$.

$$A = \begin{bmatrix} a_0 & a_1 & a_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & b_1 & b_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The output vector for $\overline{y'}$ is correspondingly

$$\overline{y'} = \begin{bmatrix} y'_n \\ y'_{n-1} \\ y'_{n-2} \end{bmatrix}.$$

So the final form for the calculations is

$$\overline{y'} = B\bar{x} + A\bar{y}$$

Let's check it

$$\begin{bmatrix} y'_n \\ y'_{n-1} \\ y'_{n-2} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \\ x_{n-2} \end{bmatrix} + \begin{bmatrix} 0 & b_1 & b_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_n \\ y_{n-1} \\ y_{n-2} \end{bmatrix} = \begin{bmatrix} a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 y_{n-1} + b_2 y_{n-2} \\ y_n \\ y_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + b_1 y_{n-1} + b_2 y_{n-2} \\ y_n \\ y_{n-1} \end{bmatrix}$$

And after each iteration of calculation

$$\overline{y'} \to \bar{y}$$

This approach can be generalized. General form of the output is

$$y(n) = \sum_{k=0}^{N} a_k x(n-k) + \sum_{k=1}^{M} b_k y(n-k)$$

If assume that $b_0 = 0$, then

$$y(n) = \sum_{k=0}^{N} a_k x(n-k) + \sum_{k=0}^{M} b_k y(n-k)$$

Input and output vectors are

$$\bar{x} = \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-N} \end{bmatrix}, \bar{y} = \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-M} \end{bmatrix}$$

And coefficient matrices

$$A = \begin{bmatrix} a_0 & a_1 & \cdots & a_N \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} b_0 & b_1 & \cdots & b_{M-1} & b_M \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

## §7.8 Comparison of FIR and IIR filters

Now we compare digital filter realizations. For the comparison we assume that filters have the same slope.

| Parameter | FIR filter | IIR filter |
|---|---|---|
| Computation difficulty/hardware costs | High | Low |
| Performance/speed | Lower (but ultra-fast with some special techniques) | Higher |
| Irregularity of a magnitude response | Depends on window | Depends on prototype |

| | | |
|---|---|---|
| Linearity of a phase response | Linear in the passband for a symmetric impulse response | Nonlinear |
| Stability | Guaranteed | Must be provided by design |

Computation difficulty and performance are strongly correlated with number of coefficients. A FIR filter has larger coefficients number than an IIR. As a consequence, you need more multiply and adder operations with more inputs. The first one affects hardware costs, the second one affects speed. Despite these issues, the FIR filters have major advantages: linear phase response and guaranteed stability.

In terms of direct forms, a FIR filter is slower than an IIR filter due to larger number of coefficients and, as a consequence, larger critical path. But a combination of the transposed form with pipelining can increase speed of FIR up to D flip-flop delay limitation. In contrast, IIR has a feedback loop that cannot be pipelined and limits speed. A parallelization (unlike pipelining) is applicable for both types and cannot highlight one type of filters.

# Chapter 8 Sample rate conversion

## §8.1 Decimation

Decimation means that we drop some samples from the input sequence, supposing them as redundant with no additional information. It is possible when sampling frequency $f_s$ is significantly higher than signal band $B$ (see Figure 8.1). Then we can occupy the empty band by reducing sampling frequency to $f'_s$ (Figure 8.2) The output sequence $y(n)$ in the decimation is expressed as

$$y(n) = x(Mn + n_0)$$

where $M$ – a decimation factor, $n_0$ – an initial shift. Then sampling frequency will be

$$f'_s = \frac{f_s}{M}$$

and spectrum will be as in Figure 8.2. The decimation is not time-invariant transformation. Its output sequence has strong dependence from the initial shift $n_0$.

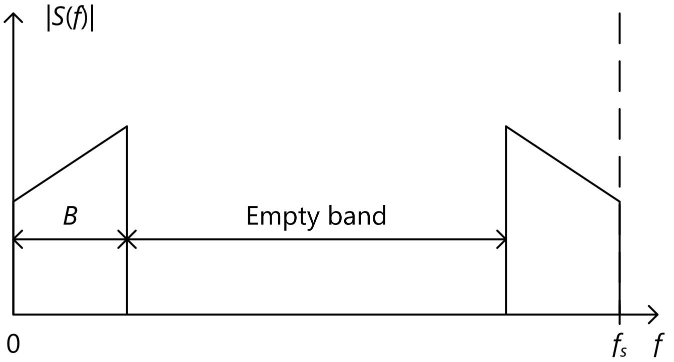



Figure 8.1 – Sampling frequency $f_s$ is significantly higher than signal band $B$

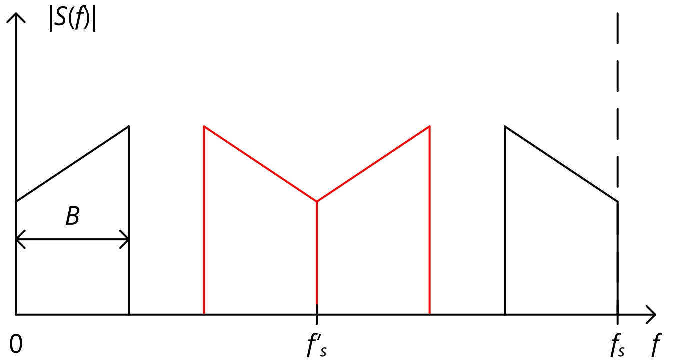



Figure 8.2 – Reduction of the sampling frequency by 2 times

Similar to the sampling, it is necessary to take into account the presence of unwanted signals and noise outside the signal band, as they can occur in the signal band after the decimation (an example is shown in Figure 8.3). To prevent it, a low-pass filter (LPF) is required before the decimation, which should attenuate the out-of-band signal to an acceptable level (Figure 8.4). The required magnitude response of such a filter is depicted as yellow slashed line in Figure 8.3.

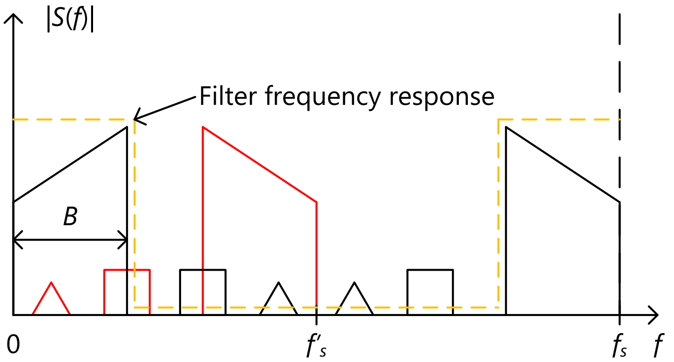


Figure 8.3 – Spectrum after decimation in presence of undesired signals and noise

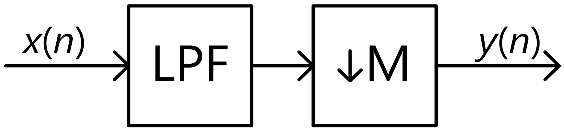



Figure 8.4 – Application of a filter in combination with decimation

## §8.2 Interpolation

Interpolation is reverse process when we reconstruct absent samples and increase sampling frequency $f_s$. Herewith, it is supposed during reconstruction that the signal band is known. To reconstruct samples and increase $f_s$ by $M$ times, we need to insert $M$-1 zeroes between the input samples (Figure 8.5).

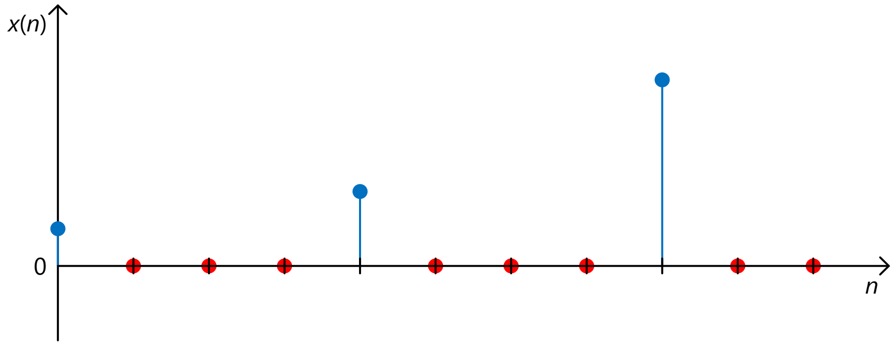



Figure 8.5 – Insertion of zeroes samples

After the insertion of $M$-1 zeroes, spectrum of the signal does not change, but sampling frequency increases by $M$ times. Let's show that zeroes insertion does not change the spectrum. After insertion

$$y(Mn) = x(n);\; f_s' = Mf_s.$$

Then

$$Y(m) = \sum_{n=0}^{MN-1} y(n) \cdot e^{-j\frac{2\pi nm}{NM}} = \sum_{n=0}^{N-1}\sum_{k=0}^{M-1} y(Mn+k) \cdot e^{-j\frac{2\pi(Mn+k)m}{NM}} =$$

$$= \sum_{n=0}^{N-1} \underbrace{y(Mn)}_{x(n)} \cdot e^{-j\frac{2\pi Mnm}{NM}} + \sum_{n=0}^{N-1}\sum_{k=1}^{M-1} \underbrace{y(Mn+k)}_{0} \cdot e^{-j\frac{2\pi(Mn+k)m}{NM}} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi nm}{N}} = X(m)$$

These changes of spectrum are illustrated in Figure 8.6.

The last step of interpolation is attenuation odd aliases of the signal that now are inside the new frequency band for $f_s'$. For this purpose, a digital filter with magnitude response shown in Figure 8.7 should be employed. Its stop-band should starts at least from the $f_s/2$ frequency.

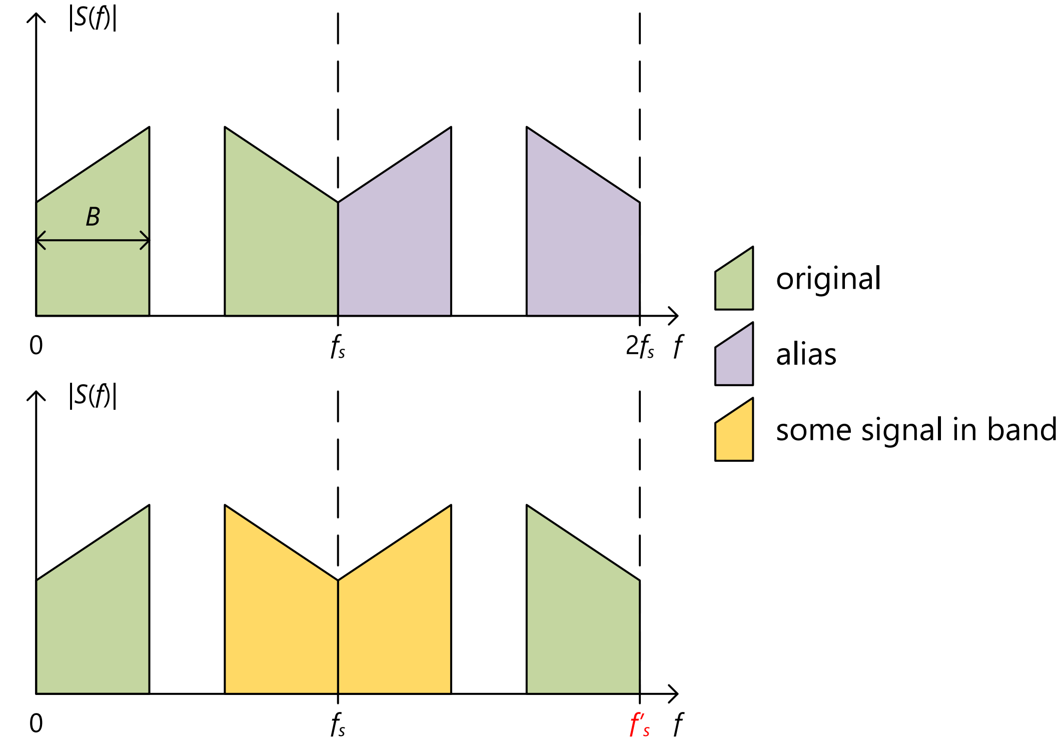



Figure 8.6 – Changes in the spectrum after zeroes insertion

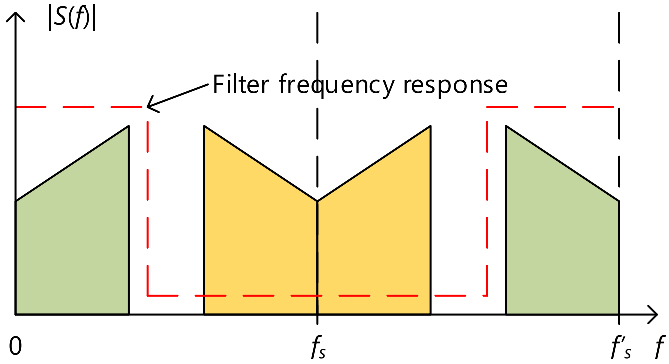



Figure 8.7 – Magnitude response of an interpolation filter.

## §8.3 Conversion with fractional coefficient

We have discussed decimation (lowering sample rate by integer number) and interpolation (increasing sample rate by integer number). Can we change sample rate by some fractional number? Yes. To do this, we need to combine interpolation and decimation in a way depicted in Figure 8.1. At first, we do an interpolation by $M$ times, then a decimation by $N$ times. As a result, we get a new sampling frequency $f'$ equals

$$f' = \frac{M}{N} \cdot f$$

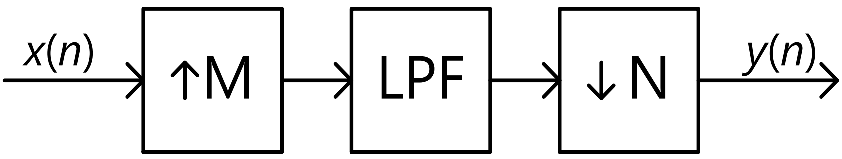



Figure 8.8 – A structure of sample rate conversion with a fractional coefficient

Between this two operations, a low-pass filter (LPF) is needed. The filter plays two roles. The first role is restoring the signal after putting zeroes during interpolation. The second role is attenuation of signals out of the passband to prevent distortion after decimation. May we change order of interpolation and decimation? Yes, but then we require two LFPs.

Now, let's have a look at changing of the signal magnitude. It is summarized in the following table

| Magnitude | Decimation | Interpolation |
|---|---|---|
| In time domain | NO change | ↓M |
| In frequency domain | ↓M | NO change |

How can we explain it? For decimation, we definitely know that the magnitude in the time domain doesn't change. For the frequency domain, we know that the magnitude is proportional to the number of samples *N*.

$$A \sim N.$$

After decimation, we get lower number of samples

$$N' = \frac{N}{M} \Rightarrow A' \sim N' = \frac{A}{M}$$

Thus, magnitude in frequency domain will be lower by *M* times.

Regarding interpolation, we have proven that the magnitude in the frequency domain doesn't change. For the time domain, we know that inverse DFT is proportional to inverse number of samples 1/*N*.

$$A \sim \frac{1}{N}$$

After interpolation, we get greater number of samples

$$N' = MN \Rightarrow A' \sim \frac{1}{N'} = \frac{1}{NM} = \frac{1}{N} \cdot \frac{1}{M} \sim A \cdot \frac{1}{M} = \frac{A}{M}$$

Thus, the magnitude in the time domain will be lower by *M* times.

# Chapter 9 Averaging

## §9.1 Introduction

Typically, received signal contains noise induced by environment. This noise can be expressed by multiplicative term μ(*t*) and additive term η(*t*), i.e.

$$s'(t) = \mu(t) \cdot s(t) + \eta(t)$$

where *s'*(*t*) – received signal, *s*(*t*) – transmitted signal. In this Chapter, we will discuss only influence of the additive term. So, the received signal is considered as

$$s'(t) = \alpha \cdot s(t) + \eta(t).$$

Additive term is commonly considered as white noise, that is we assume that η(*t*) has the normal distribution and the mean value (the mathematic expectation) equals 0. It is expressed by the following formulas

$$\bar{\eta} = \int_{-\infty}^{+\infty} \eta \cdot P(\eta) d\eta = 0; P(\eta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\eta-\bar{\eta})^2}{2\sigma^2}}.$$

In signal processing, we consider that processes have ergodicity property. Ergodicity means that averaging of system behavior over time axis is equivalent to averaging over all possible values. Thus, previous expression for the mean value of the noise can be rewritten as

$$\bar{\eta} = \int_{-\infty}^{+\infty} \eta(t) dt = 0;$$

Then it is possible to state that averaging of the received signal over time provide us the transmitted signal

$$s'_{av}(t) = \alpha \cdot s(t) + \bar{\eta} = \alpha \cdot s(t)$$

To make averaging of the received signal, we need to acquire several sample sets, where it is known that transmitted signal was the same. Averaging over time can be divided into two cases: coherent averaging and incoherent averaging (Figure 9.1). Coherent averaging is applied when we know the phase of each sample set; incoherent averaging, on the contrary, is applied when the phase of each sample set is unknown.

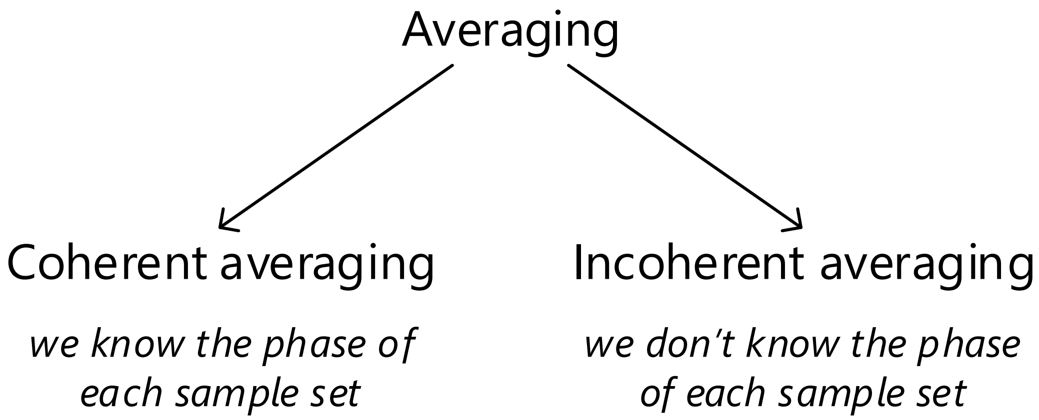



Figure 9.1 – Types of averaging

## §9.2 Coherent averaging

If we know the phase of each sample set, then we can overlap sample sets with each other (see Figure 9.2). There *k*-th sample set with duration *T* is designated as $s'_k(t)$. Each sample set starts with the same phase, and at the specific moment in time samples differ only by noise value. Then we can express averaged received signal in the following form

$$s'_{av}(k) = \frac{1}{K} \sum_{i=1}^{K} s'_k(k)$$

For the extreme case when infinite number of sets is taken into account

$$s'_{av}(k) = \lim_{K\to\infty} \frac{1}{K}\sum_{i=1}^{K} s'_k(k) = \alpha \cdot s(k) + \bar{\eta} = 0$$

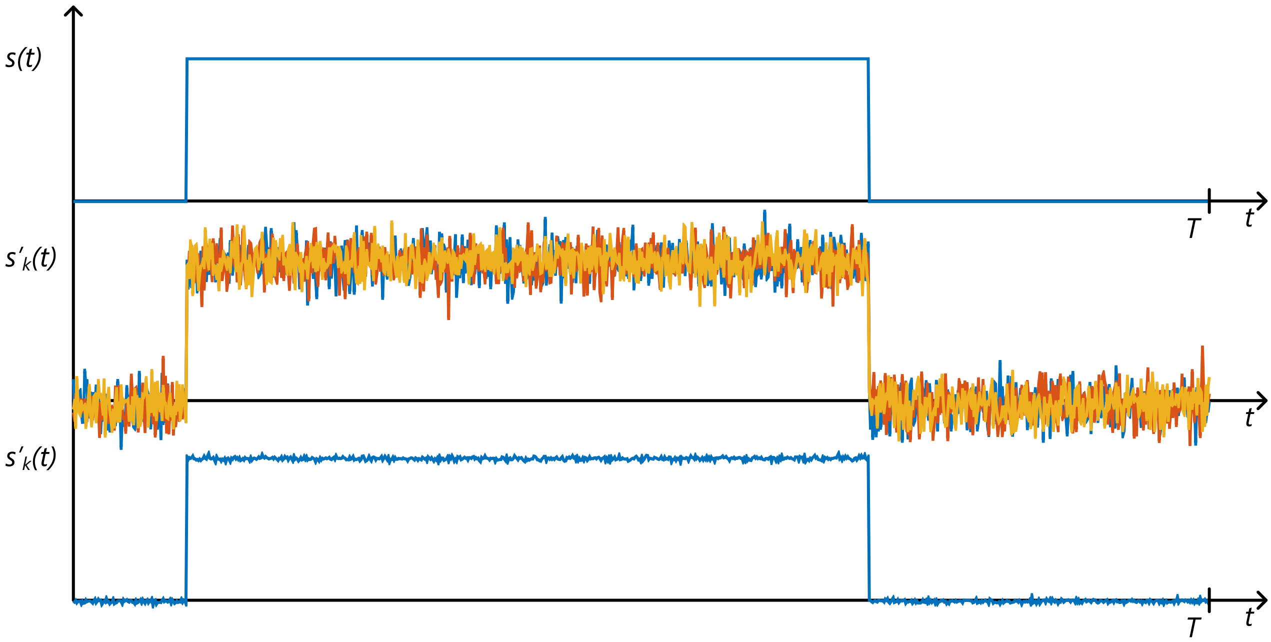



Figure 9.2 – Illustration of the coherent averaging

For ergodicity process, we may rewrite the above formula as

$$s'_{av}(k) = \frac{1}{K}\sum_{i=1}^{K} s'(N \times i + k)$$

where $N$ corresponds to the sample set duration $T$.

Due to linearity of the DFT, equation for the averaged signal can be transformed into

$$S'_{av}(m) = \frac{1}{K}\sum_{i=1}^{K} S'_k(m)$$

where $S'_{av}(m)$ – DFT of the averaged signal, $S'_k(m)$ – DFT of $k$-th sample set. This means that coherent averaging can be done both in time and frequency domains. Moreover, making averaging in frequency domain, we can restore averaged signal in time domain through performing inverse DFT for $S'_{av}(m)$. After averaging dispersion of noise is reduced by number of sample sets, i.e.

$$\sigma^2_{av} = \frac{\sigma^2_{in}}{K}$$

## §9.3 Incoherent averaging

On the contrary, incoherent averaging is used when we don't know a phase of each sample set. It is illustrated in Figure 9.3. There each sample set starts with different shift in time (phase). So, the approach for averaging proposed in the previous section cannot be applied here.

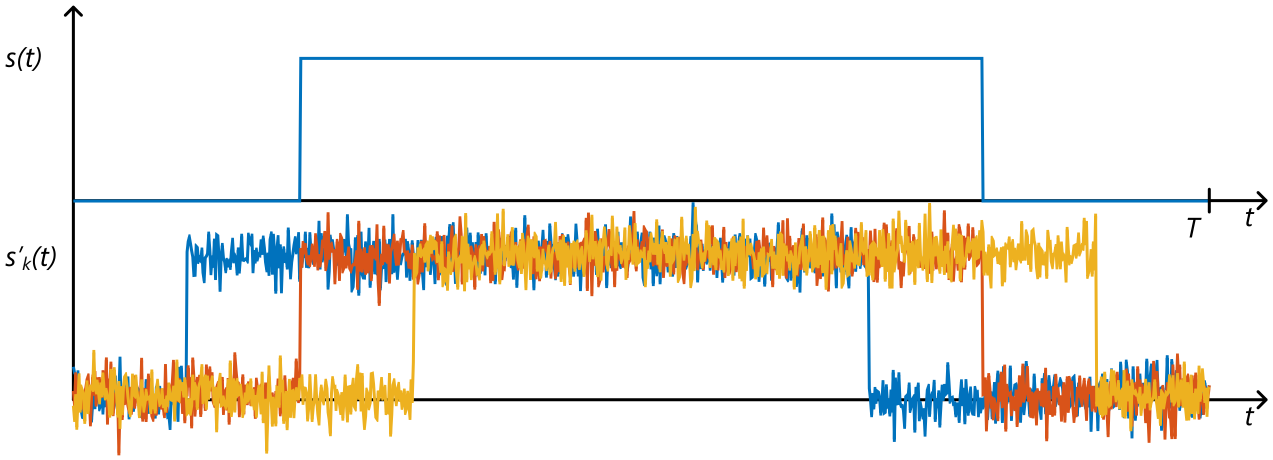



Figure 9.3 – Illustration of the incoherent averaging

Can we still reduce noise influence through averaging? We can still do averaging in frequency domain. But with a little change. We do it only for magnitude response. Due to different phase of sample sets, averaging of phase responses is not valid.

$$|S'_{av}(m)| = \frac{1}{K}\sum_{i=1}^{K} |S'_k(m)|$$

And, as a consequence, there we cannot restore averaged signal through inverse DFT due to absence of information about phase (phase response). Improvement for this type of averaging is the same as in the previous case, that is

$$\sigma_{av}^2 = \frac{\sigma_{in}^2}{K}$$

In the incoherent averaging, we cannot reconstruct the original signal (we don't know its phase component). As a result, the incoherent averaging does not actually decrease noise power and, consequently, does not improve SNR value. From this point of view, a decrease in the dispersion $\sigma_{av}$, indeed, only means a decrease in fluctuations of noise samples in the spectrum.

## §9.4 Realization of averaging

In Figure 9.4 you can see example of averaging realization. It contains several averaging FIR filters. Each filter provides one sample of the averaged signal. The input sequence *x*(*n*) is switched between these filters over the time. That is, samples with indices 0, *N*, 2*N* and so on go to Filter 0; with indices 1, *N*+1, 2*N*+1 and so on to Filter 1 and etc. Sequence *x*(*n*) can be either time domain or frequency domain samples.

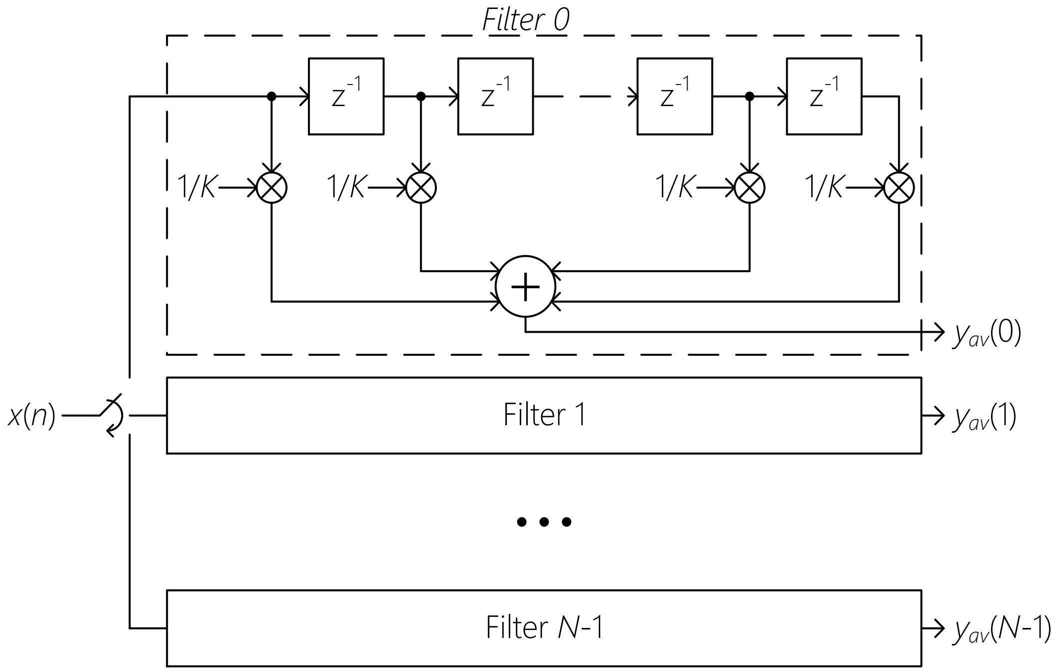



Figure 9.4 – Averaging filter

## §9.5 Exponential averaging

The realization presented in the previous section requires a large number of delays, multiplications and summations. If you need to reduce only high-frequency noise, there are more efficient ways. One of this efficient ways to do an averaging is an exponential averaging, which structure is shown in Figure 9.5. The output of this structure is defined by

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

From there, we can obtain its transfer function

$$T(z) = \frac{\alpha}{1 - (1 - \alpha) \cdot z^{-1}}$$

And impulse response

$$h(n) = \alpha \cdot (1 - \alpha)^n.$$

Impulse response of such a filter with different α is depicted in Figure 9.6.

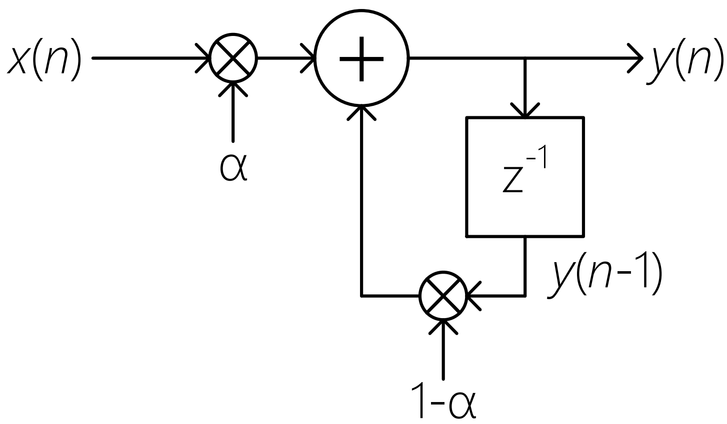



Figure 9.5 – A structure of the exponential averaging

Exponential averaging is a parametric low-pass IIR filter. Its parameter α defines noise reduction factor (i.e. cut-off frequency of the filter). Varying coefficient α, we can change the influence of the input sample to the output. With α → 0, the input sample doesn't affect the output and, therefore, noise is reduced. With α = 1, the output exactly equals the input, and noise reduction is absent. In Figure 9.7, you can see dependence between and α and SNR improvement. This improvement can be expressed by the following equations

$$\sigma_{av}^2 = \frac{\alpha}{2-\alpha}\sigma_{in}^2; S = -10\log_{10}\frac{\alpha}{2-\alpha}$$

For instance, $S$ = 0 dB with α = 0 and $S$ ≈ 13 dB with α = 0.1. This structure is significantly simpler than the filter from the section §9.4. However, it reduces only high-frequency noise components and cannot reduce in-band noise, unlike the fair averaging presented in the section §9.4.

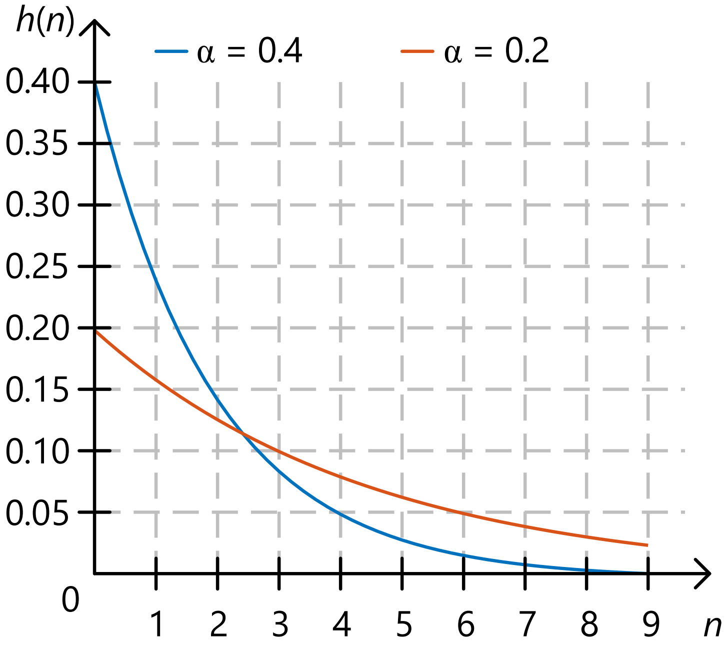



Figure 9.6 – An impulse response of the exponential averaging with different α

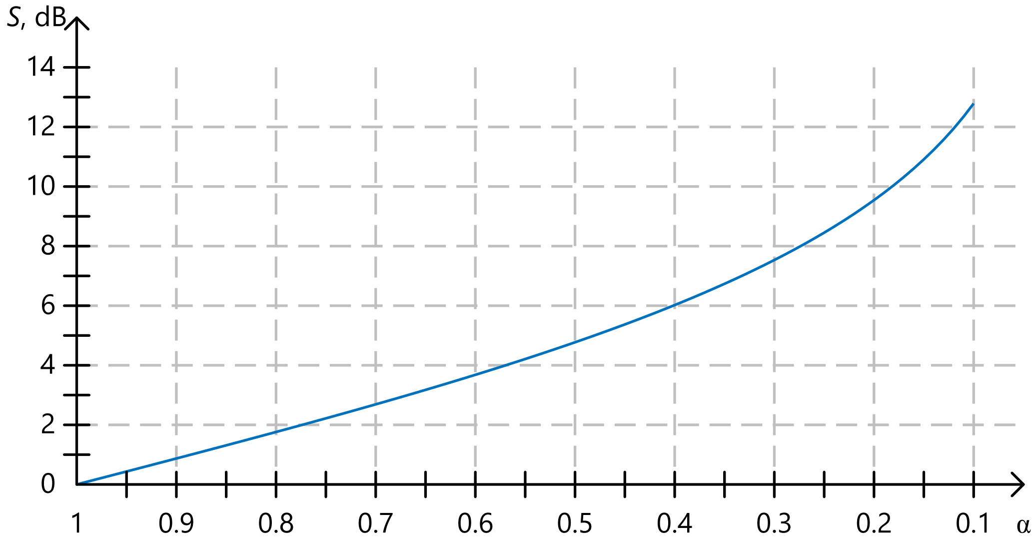



Figure 9.7 – SNR increase caused by the exponential averaging

# Chapter 10 Analytic signal

## §10.1 Introduction

We already know real signals. For example:

$$s(t) = A\cos(\omega t).$$

But there is an analytic signal (or a complex signal) $z(t)$ corresponding to this real signal. It is expressed by:

$$z(t) = s(t) + j\hat{s}(t),$$

where $\hat{s}(t)$ – an orthogonal complement to $s(t)$. The orthogonal complement is the Hilbert transform of $s(t)$. That is, it can be calculated by the following expression:

$$\hat{s}(t) = \mathcal{H}\{s(t)\} = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{s(\tau)}{t-\tau} d\tau.$$

where $\mathcal{H}$ – the Hilbert Transform. From circuit design point of view Hilbert transform may be interpreted as phase shifter for –π/2.

Now let's discuss the spectrum of the analytic signal. Let $S(\omega)$ – the spectrum of $s(t)$ and $\hat{S}(\omega)$ – the spectrum of $\hat{s}(t)$. It is known (will be discussed in Section §11.1) that $\hat{S}(\omega)$ is equal to

$$\hat{S}(\omega) = S(\omega) \cdot e^{-j\frac{\pi}{2}\cdot \operatorname{sign} \omega}.$$

Then the spectrum $Z(\omega)$ of the analytic signal is

$$Z(\omega) = S(\omega) + jS(\omega) = \begin{cases} S(\omega) + j\big(-jS(\omega)\big), \omega > 0 \\ S(\omega) + j\big(jS(\omega)\big), \omega < 0 \end{cases} = \begin{cases} 2S(\omega), \omega > 0 \\ 0, \omega < 0 \end{cases}$$

Both spectrums are depicted in Figure 10.1.

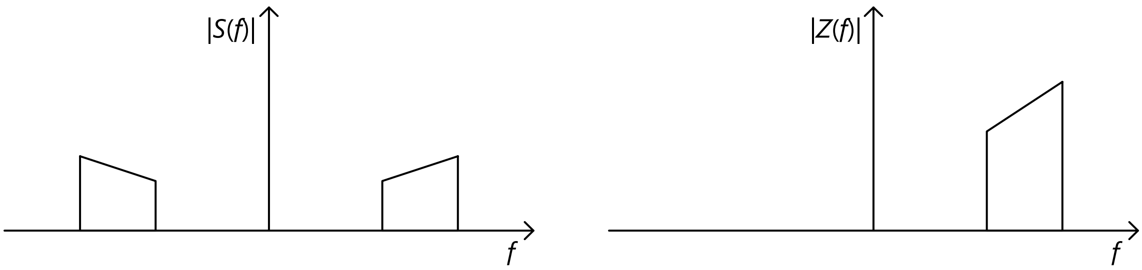



Figure 10.1 – Spectrums of a real and an analytic signal

## §10.2 Complex envelope

Let's assume that the original input signal has some modulation, then it can be written in the following form

$$s(t) = A(t) \cdot \cos\big(\omega_0 t + \varphi(t)\big).$$

where $A(t)$ – modulation of magnitude, $\varphi(t)$ – modulation of phase, $\omega_0$ – carrier frequency. For signals with a relatively narrow band ($B << f_0$) orthogonal complement equals

$$\hat{s}(t) = A(t) \cdot \sin\big(\omega_0 t + \varphi(t)\big).$$

As a result analytic signal z(t) becomes

$$z(t) = A(t) \cdot \cos\big(\omega_0 t + \varphi(t)\big) + j \cdot A(t) \cdot \sin\big(\omega_0 t + \varphi(t)\big) = A(t) \cdot e^{j(\omega_0 t + \varphi(t))}$$

Rewrite this expression

$$z(t) = A(t) \cdot e^{j\omega_0 t + \varphi(t)} = \underbrace{A(t) \cdot e^{j\varphi(t)}}_{F(t)} \cdot e^{j\omega_0 t} = F(t) \cdot e^{j\omega_0 t}$$

Now all information is concentrated in function $F(t)$ that is called "complex envelope". We can get complex envelope by the next multiplication

$$F(t) = z(t) \cdot e^{-j\omega_0 t}$$

This is one of the ways to get the complex envelope. The spectrum of the complex envelope is

$$F(\omega)=\int_{-\infty}^{+\infty}F(t)e^{-j\omega t}dt=\int_{-\infty}^{+\infty}z(t)\cdot e^{-j\omega_0 t}\cdot e^{-j\omega t}dt=\int_{-\infty}^{+\infty}z(t)\cdot e^{-j(\omega+\omega_0)t}dt=\begin{cases}2S(\omega+\omega_0),\omega+\omega_0\geq 0\\ 0,\omega+\omega_0<0\end{cases}$$

Spectrums of a real signal, analytic signal and complex envelope are depicted in Figure 10.2.

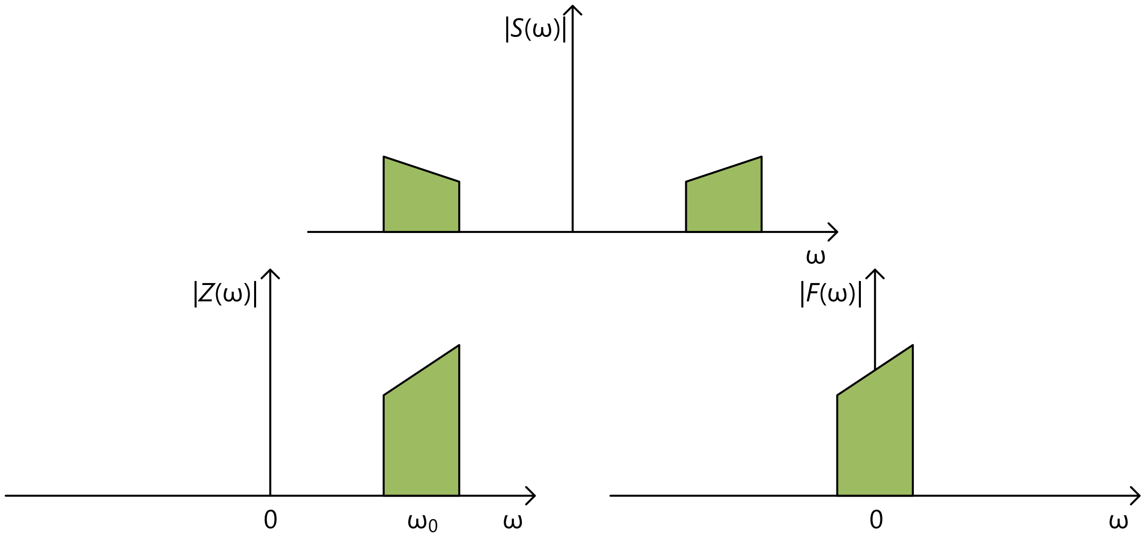



Figure 10.2 – Spectrums of the real, complex and envelope signals

Now, we take a closer look at the complex envelope. Write it down again

$$F(t)=A(t)\cdot e^{j\varphi(t)}$$

Take the absolute value and phase of this complex function

$$|F(t)|=\left|A(t)\cdot e^{j\varphi(t)}\right|=|A(t)|=A(t)$$

$$\arg F(t)=\arg\left(A(t)\cdot e^{j\varphi(t)}\right)=\arg\left(A(t)\right)+\varphi(t)=\varphi(t)$$

Thus, amplitude and phase modulations can be detected by means of operations with complex envelope. Moreover, absolute value of the complex envelope can be extracted even without addition or multiplication. Indeed,

$$|z(t)|=\sqrt{\left(A(t)\cdot\cos\left(\omega_0 t+\varphi(t)\right)\right)^2+\left(A(t)\cdot\sin\left(\omega_0 t+\varphi(t)\right)\right)^2}$$

$$=A(t)\cdot\underbrace{\sqrt{\cos^2\left(\omega_0 t+\varphi(t)\right)+\sin^2\left(\omega_0 t+\varphi(t)\right)}}_{1}=A(t)$$

### §10.3 Quadrature components

Parts of analytic signal are often called quadrature components and designated as *I* (in-phase) and Q (quadrature). Using this terms, analytic signal is written as

$$F(t)=I(t)+jQ(t);$$

$$I(t)=A(t)\cdot\cos\varphi(t)\,;Q(t)=A(t)\cdot\sin\varphi(t).$$

How to transmit quadrature components? This process is shown in Figure 10.2 and can be explained like that

$$s(t)=A(t)\cdot\cos\left(\omega_0 t+\varphi(t)\right)=A(t)(\cos\varphi(t)\cdot\cos\omega_0 t-\sin\varphi(t)\cdot\sin\omega_0 t)=I(t)\cdot\cos\omega_0 t-Q(t)\cdot\sin\omega_0 t$$

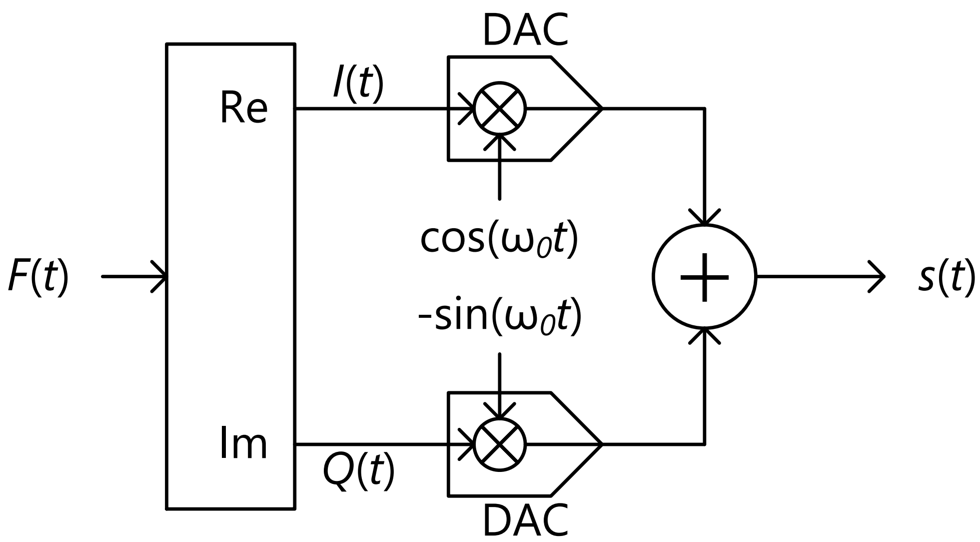



Figure 10.3 – A generation of a real signal *s*(*t*) from complex envelope *F*(*t*)

There, a structure with Digital-to-RF DAC is presented. However, the DAC can be placed both before mixer – Digital-to-IF (Intermediate Frequency) – and after adder – Direct Digital Synthesis (DDS).

In terms of receiver, we have already mentioned in previous section one way of quadrature components obtaining – Direct Digital Conversion (DDC) with multiplying the complex signal by the carrier (i.e. mixer in the digital domain). Another way is a structure presented in Figure 10.4, where the mixer is analog and its output signal has Zero-IF or Low-IF. The low-pass filter in Figure 10.4 play two roles:

- Anti-aliasing filter for the ADC;
- Filtering high-frequency image after frequency conversion.

Let's talk about the last point. After conversion the received signal *s*(*t*) will be multiplied by the carrier with frequency $\omega_0$, i.e.

*In I channel*

$$s(t)\cdot\cos\omega_0 t = A(t)\cdot\cos\big(\omega_0 t+\varphi(t)\big)\cdot\cos\omega_0 t = A(t)\cdot\frac{\cos(\omega_0 t+\varphi(t)-\omega_0 t)+\cos(\omega_0 t+\varphi(t)+\omega_0 t)}{2}$$

$$= A(t)\cdot\frac{\cos\big(\varphi(t)\big)+\cos\big(2\omega_0 t+\varphi(t)\big)}{2} = \frac{1}{2}I(t)+\frac{A(t)}{2}\cdot\cos\big(2\omega_0 t+\varphi(t)\big).$$

*In Q channel*

$$s(t)\cdot(-\sin\omega_0 t) = A(t)\cdot\cos\big(\omega_0 t+\varphi(t)\big)\cdot\sin(-\omega_0 t)$$

$$= A(t)\cdot\frac{\sin(\omega_0 t+\varphi(t)-\omega_0 t)+\sin(\omega_0 t+\varphi(t)+\omega_0 t)}{2} = A(t)\cdot\frac{\sin\big(\varphi(t)\big)+\sin\big(2\omega_0 t+\varphi(t)\big)}{2}$$

$$= \frac{1}{2}Q(t)+\frac{A(t)}{2}\cdot\sin\big(2\omega_0 t+\varphi(t)\big).$$

The component with frequency $2\omega_0$ is unnecessary and should be filtered out.

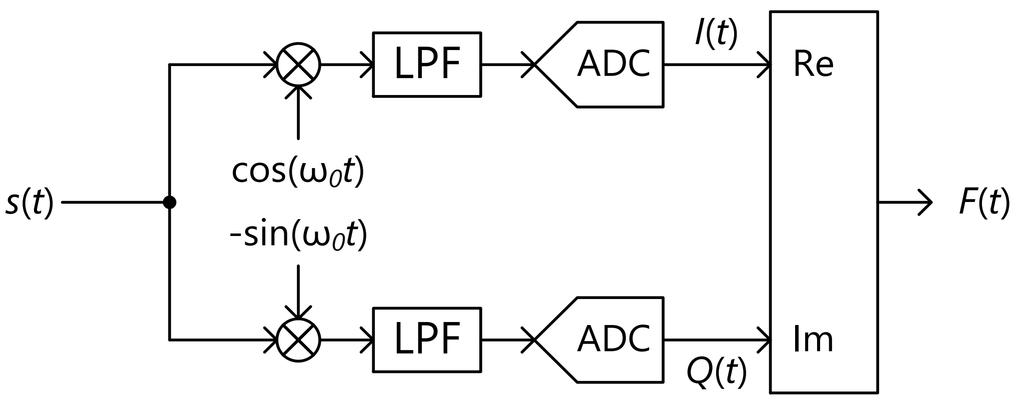



Figure 10.4 – Zero IF receiver

## §10.4 Why do we need it?

Why do we need so strange entity as analytic signal? Let's imagine that we get 3 samples of a harmonic signal (Figure 10.5). Can we determine amplitude, frequency and phase of this signal? Yes, for this purpose, we need to solve the following system

$$\begin{cases} A \cdot \cos(\omega \cdot 0t_s + \varphi) = x(0); \\ A \cdot \cos(\omega \cdot 1t_s + \varphi) = x(1); \\ A \cdot \cos(\omega \cdot 2t_s + \varphi) = x(2). \end{cases}$$

There are 3 unknown variables: $A$, ω and φ. It is not so easy to solve this, but it is possible. What if we have analytic signal that corresponds to this real signal? It will be easier to determine this parameters, that is

$$A = |z(n)|; \varphi = \arg z(n)$$

And this is with just one sample. The second sample can give us the frequency

$$\omega = \arg z(n) - \arg z(n-1)$$

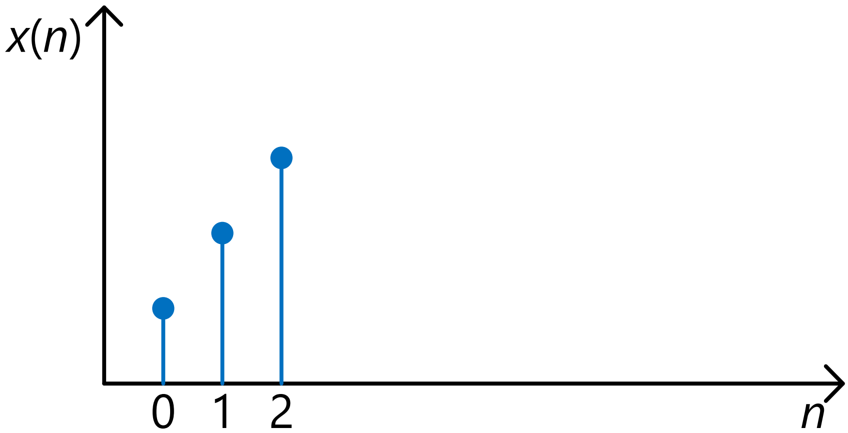



Figure 10.5 – Three samples of some harmonic signal

# Chapter 11 Hilbert transform

## §11.1 Transfer function and impulse response of Hilbert transform

### *11.1.1 Continuous time*

In §10.1 we have discussed that orthogonal complement can be obtained with Hilbert transform

$$\hat{s}(t) = \mathcal{H}\{s(t)\} = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{s(\tau)}{t-\tau}d\tau.$$

Here, we discuss Hilbert transform more detailed. From the expression above, it is seen that Hilbert transform is a convolution of function *s*(*t*) and function *h*(*t*)

$$h(t) = \frac{1}{\pi t}$$

In other words, we can say that Hilbert transform is a LTI system with impulse response *h*(*t*). Let's try to get its transfer function.

$$H(\omega) = \int_{-\infty}^{+\infty} h(t)e^{-j\omega t}dt = \int_{-\infty}^{+\infty}\frac{1}{\pi t}e^{-j\omega t}dt = \int_{-\infty}^{0}\frac{1}{\pi t}e^{-j\omega t}dt + \int_{0}^{+\infty}\frac{1}{\pi t}e^{-j\omega t}dt =$$

$$= \int_{+\infty}^{0}\frac{1}{\pi(-t)}e^{-j\omega(-t)}d(-t) + \int_{0}^{+\infty}\frac{1}{\pi t}e^{-j\omega t}dt = \int_{+\infty}^{0}\frac{1}{\pi t}e^{j\omega t}dt + \int_{0}^{+\infty}\frac{1}{\pi t}e^{-j\omega t}dt = -\int_{0}^{+\infty}\frac{1}{\pi t}e^{j\omega t}dt + \int_{0}^{+\infty}\frac{1}{\pi t}e^{-j\omega t}dt$$

$$= \int_{0}^{+\infty}\frac{e^{-j\omega t} - e^{j\omega t}}{\pi t}dt = \int_{0}^{+\infty}\frac{-2j\sin\omega t}{\pi t}dt = \frac{-2j}{\pi}\cdot\frac{\pi}{2}\cdot\operatorname{sign}\omega = -j\cdot\operatorname{sign}\omega$$

(Reminder of a table integral)

$$\int_{0}^{+\infty}\frac{\sin kx}{x}dx = \frac{\pi}{2}\cdot\operatorname{sign} k$$

Finally, we have that

| Impulse response | Transfer function |
|---|---|
| $h(t) = \frac{1}{\pi t}$ | $H(\omega) = -j\cdot\operatorname{sign}\omega = \begin{cases} j, \omega < 0 \\ 0, \omega = 0 \\ -j, \omega > 0 \end{cases}$ |

Impulse response and transfer function of Hilbert transform are illustrated in Figure 11.1.

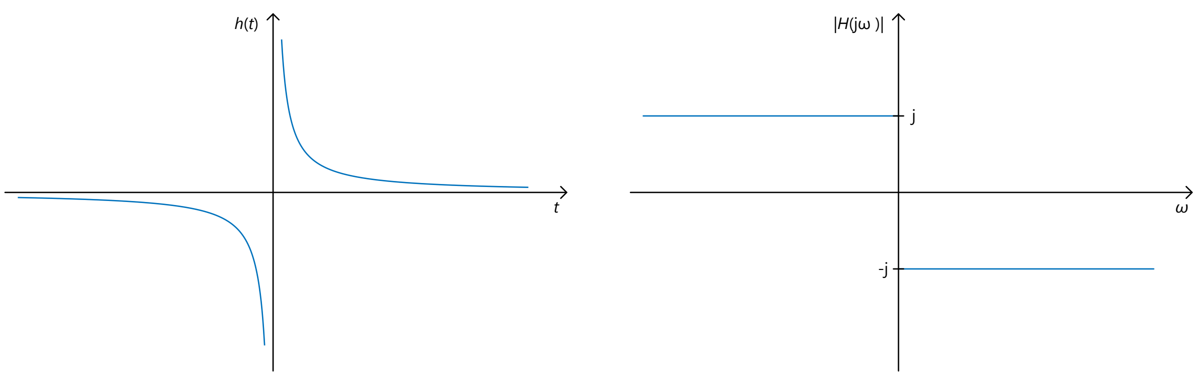
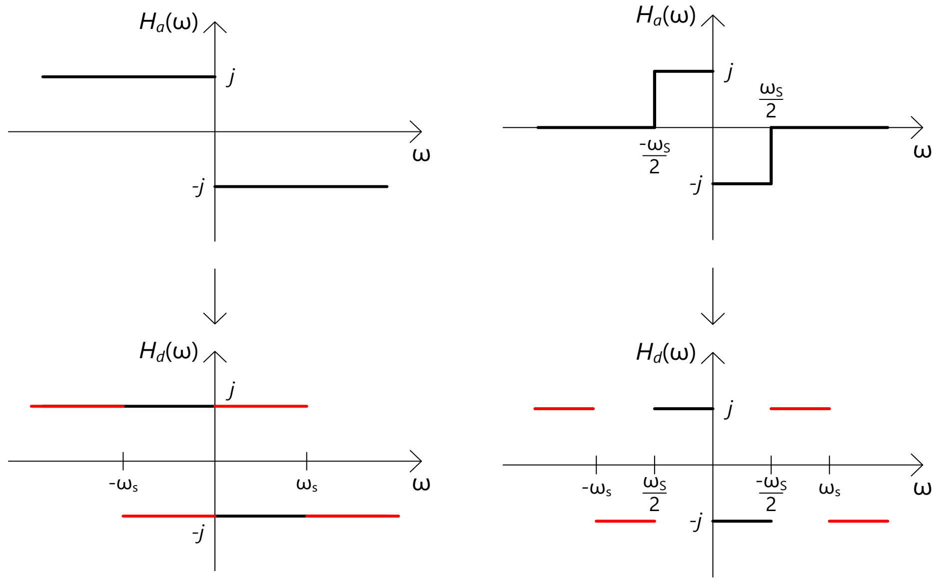



Figure 11.1 – Impulse response and transfer function of Hilbert transform

### *11.1.2 Discrete time*

As we see, the transfer function of the Hilbert transform is not limited by frequency. It results in impossibility to just discretize the impulse response in time to get its discrete version. This issue is illustrated in Figure 11.1. After discretization transfer function will repeat and overlap each other (Figure 11.1a), and the transformation will loss its properties. To prevent distortion caused by spectrum overlapping, we need to limit transfer function by frequency. Let transfer function be equal to 0 outside the baseband ($\omega>\omega_s/2$), then spectrum repetition does not change properties of our transformation (Figure 11.1b).




$$H_a(\omega) = -j \cdot \operatorname{sign} \omega$$

a)

$$H_a(\omega) = \begin{cases} -j \cdot \operatorname{sign} \omega, |\omega| \le {}^{\omega_s}/_2 \\ 0, |\omega| > {}^{\omega_s}/_2 \end{cases}$$

b)

Figure 11.2 – Transfer function of the Hilbert transform after discretization without limitation of frequency range (a) and with limitation of frequency range (b)

Now we need to get corresponding impulse response with the help of inverse IFT.

$$h_d(t) = \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2} H_d(\omega)e^{j\omega t}d\omega = \frac{1}{2\pi}\int_{-\omega_s/2}^{0} je^{j\omega t}d\omega + \frac{1}{2\pi}\int_{0}^{\omega_s/2} -je^{j\omega t}d\omega = \frac{j}{j2\pi t}e^{j\omega t}\bigg|_{\frac{-\omega_s}{2}}^{0} - \frac{j}{j2\pi t}e^{j\omega t}\bigg|_{0}^{\frac{\omega_s}{2}}$$

$$= \frac{1}{2\pi t}\left(1 - e^{-j\frac{\omega_s}{2}t} + 1 - e^{j\frac{\omega_s}{2}t}\right) = \frac{1}{2\pi t}\left(2 - 2\cos\frac{\omega_s t}{2}\right) = \frac{1}{\pi t}\left(1 - \cos\frac{\omega_s t}{2}\right) = \frac{2\sin^2\frac{\omega_s t}{4}}{\pi t}$$
$$= \frac{2\sin^2\frac{\pi f_s t}{2}}{\pi t}$$

That is

$$h_d(t) = \frac{1}{\pi t}\left(1 - \cos\frac{\omega_s t}{2}\right) = \frac{2\sin^2\frac{\omega_s t}{4}}{\pi t}.$$

To get the value of impulse response, we need to calculate a limit

$$\lim_{t\to 0} h_d(t) = \lim_{t\to 0}\frac{2\sin^2\frac{2\pi f_s t}{4}}{\pi t} = \lim_{t\to 0}\frac{2\left(\frac{\pi f_s t}{2}\right)^2}{\pi t} = \lim_{t\to 0}\frac{\pi f^2{}_s t}{2} = 0$$

Finally, the discrete version of the impulse response can be obtained with safe time variable replacement

$$t \to nt_s$$

$$\frac{2\sin^2\frac{\omega_s t}{4}}{\pi nt_s} = \frac{2\sin^2\frac{\pi f_s t}{2}}{\pi nt_s} = \frac{2\sin^2\frac{\pi f_s nt_s}{2}}{\pi nt_s} = \frac{2\sin^2\frac{\pi n}{2}}{\pi nt_s}$$

This impulse response is depicted in Figure 11.3. It represents discretization of oscillation function with frequency $f_s/2$.

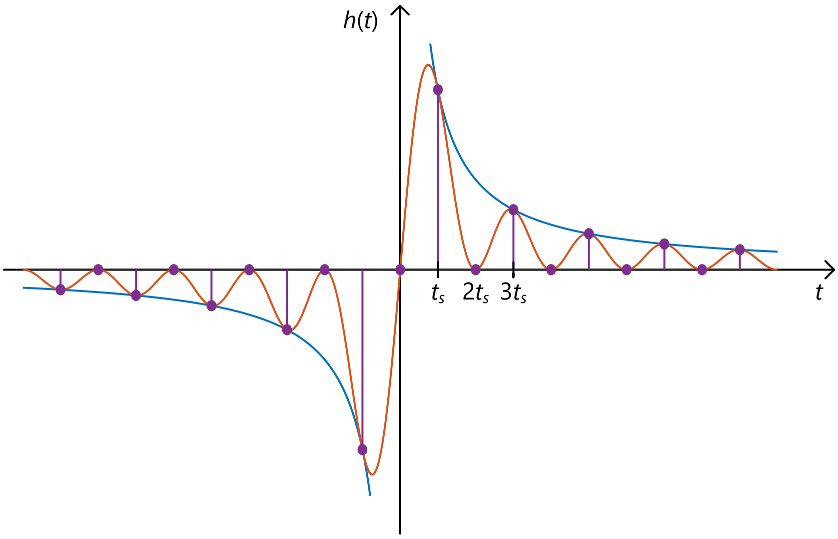



Figure 11.3 – Impulse response of a discrete Hilbert transform

### §11.2 Hilbert converter

Now, we know how impulse response of Hilbert transform looks like and can discuss how to implement such a device. The straightforward approach to realize Hilbert converter is a FIR design. Impulse response samples are just filter coefficients in this case. And there we have 2 options: with odd or even number of coefficients. Why does it matter?

Impulse response of Hilbert transform is antisymmetric. From §6.3 we know that different number of samples leads us to different types of magnitude responses (Figure 11.4). In case of even number of samples, magnitude response equals to 0 only at 0 and $f_s$ frequencies. This is acceptable behavior as constant level cannot have phase and orthogonal complement. In case of odd number of samples, magnitude response additionally equals 0 at $f_s/2$ frequency. This behavior is undesirable as it limits pass-band of the Hilbert converter.

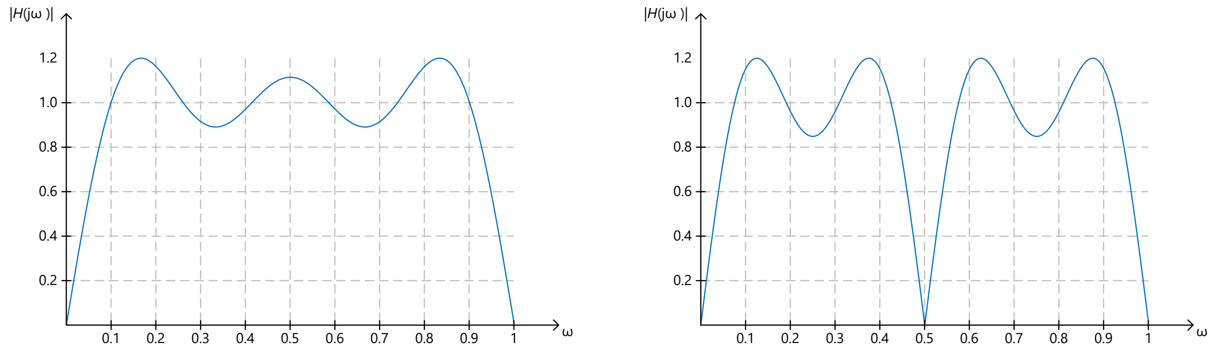



Figure 11.4 – Magnitude responses of Hilbert converter with even (left) and odd (right) number of samples

Now, let's look at this issue from another point of view. Examples of a converter structure for even and odd number of coefficients are shown in Figures 11.5 and 11.6. We know that the delay of a device is defined by its group delay, which is expressed for a FIR filter as

$$G = -\frac{d\varphi}{d\omega} = t_s\frac{N-1}{2}.$$

In case of an even number of coefficients, the group delay becomes non-integer and makes it difficult to synchronize $y(n)$ and $\hat{y}(n)$ (delay $z^{-1/2}$ cannot be implement in single rate systems). In case of an odd number of coefficients, the group delay is integer and $y(n)$ and $\hat{y}(n)$ can be easily synchronized by taking $y(n)$ from the middle of converter.

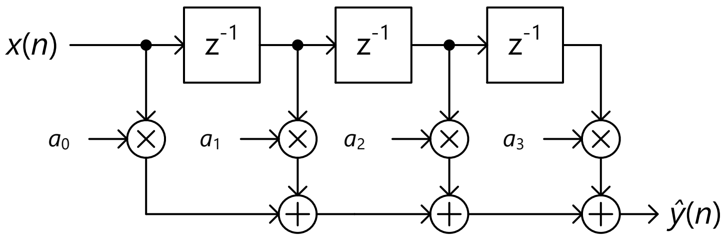



Figure 11.5 – Structure of converter with even number of coefficients

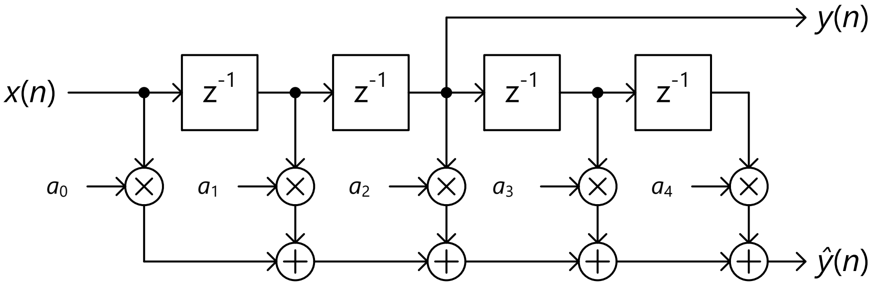



Figure 11.6 – Structure of converter with odd number of coefficients

There is a feature of impulse response for even number of samples. As impulse must be antisymmetric it must not contain a zero sample at the center. It results in skipping all even samples in the impulse response and impulse response becomes as in Figure 11.7. In frequency domain, it is equivalent to the following. We get an impulse response for doubled $f_s$ and decimate it with indices $2n+1$. The decimation leads to an overlap of range $[f_s, 2f_s]$ to range $[0, f_s]$. However, as they are identical, nothing changes.

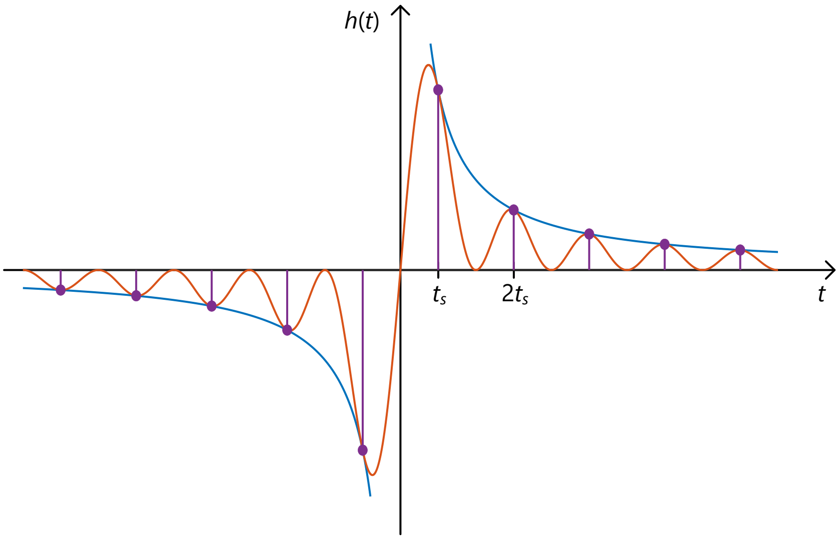



Figure 11.7 – Impulse response for even number of samples

## §11.3 Hilbert transform in frequency domain

Another approach to implement Hilbert transform and get analytic signal is a conversion in frequency domain. One of possible solutions is to use fast convolution scheme. This implies:

1. Do DFT from the input sequence;
2. Multiply by the transfer function of Hilbert transform;
3. Restore signal with help of inverse DFT;
4. Add delay to the input sequence to synchronize components of an analytic signal.

Such a scheme is depicted in Figure 11.6.

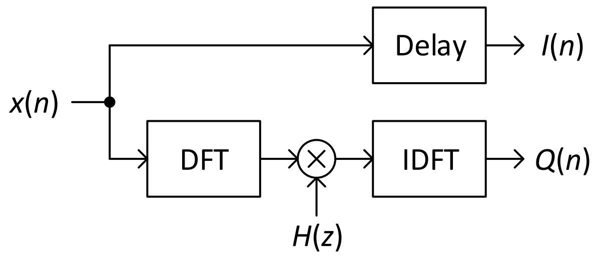



Figure 11.8 – Hilbert transform with fast convolution

Other possible solution is to transform the spectrum of a real-valued signal to the spectrum of a complex-valued signal by the following procedure:

1. Do DFT from the input sequence;
2. Drop all samples for negative frequency;
3. Multiply samples for positive frequencies by 2 times (except 0 and fs/2);
4. Do inverse DFT and get analytic signal;

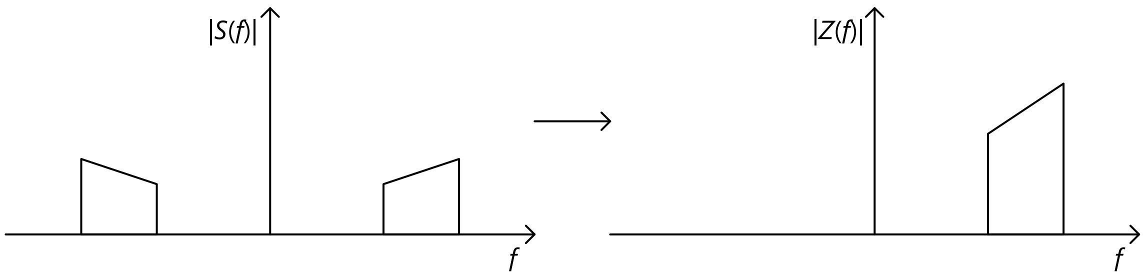



Figure 11.9 – Conversion of a real-value signal to a complex-valued signal

There are some issues that should be taken into account. Both solutions rely on DFT and susceptible for leakage. As a result, any change of frequency components may produce harmonic distortion and inaccurate result will be obtained. Other issue is also because of leakage: both solutions cannot be directly used in real-time processing. Only processing of the complete samples set makes sense. However, solution can be modified to be applicable in real-time processing.

# References


1. Сергиенко А. Б. Цифровая обработка сигналов. Санкт-Петербург: БХВ-Петербург, 2013.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов: СПб.: БХВ-Петербург, 2005.
3. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. М.: Техносфера, 2009.
4. Lyons R. G. Understanding Digital Signal Processing, 3rd ed. Pearson Education, Inc., 2011.
5. Цикин И. А. Сигналы в информационных радиосистемах [Электронный ресурс] : учебное пособие / И. А. Цикин ; Санкт-Петербургский политехнический университет Петра Великого. Санкт-Петербург : Изд-во Политехн. ун-та, 2017. 10.18720/SPBPU/2/i17-364.

*Енученко Михаил Сергеевич*
*Иванов Никита Валерьевич*

# ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Конспект лекций

Налоговая льгота – Общероссийский классификатор продукции
ОК 005-93, т. 2; 95 3005 – учебная литература

---

Подписано в печать 02.10.2024. Формат 60×84/16. Печать цифровая.
Усл. печ. л. 6,5. Тираж 52. Заказ 4619.

---

Отпечатано с готового оригинал-макета, предоставленного авторами,
в Издательско-полиграфическом центре Политехнического университета.
195251, Санкт-Петербург, Политехническая ул., 29.
Тел.: (812) 552-77-17; 550-40-14.