

Information Technology: Transmission, Processing, and Storage

---

*Digital Signal  
Processing for  
Measurement  
Systems*  
*Theory and Applications*

---

Gabriele D'Antona  
Alessandro Ferrero

Copyrighted Material

# **Digital Signal Processing for Measurement Systems**

Theory and Applications

# Information Technology: Transmission, Processing, and Storage

Series Editors: Robert Gallager  
*Massachusetts Institute of Technology  
Cambridge, Massachusetts*

Jack Keil Wolf  
*University of California at San Diego  
La Jolla, California*

---

The Multimedia Internet  
Stephen Weinstein

Coded Modulation Systems  
John B. Anderson and Arne Svensson

Communication System Design Using DSP Algorithms:  
With Laboratory Experiments for the TMS320C6701 and TMS320C6711  
Steven A. Tretter

Interference Avoidance Methods for Wireless Systems  
Dimitrie C. Popescu and Christopher Rose

MIMO Signals and Systems  
Horst J. Bessai

Multi-Carrier Digital Communications: Theory and Applications of OFDM  
Ahmad R.S. Bahai, Burton R. Saltzberg and Mustafa Ergan

Performance Analysis and Modeling of Digital Transmission Systems  
William Turin

Stochastic Image Processing  
Chee Sun Won and Robert M. Gray

Wireless Communications Systems and Networks  
Mohsen Guizani

A First Course in Information Theory  
Raymond W. Yeung

Nonuniform Sampling: Theory and Practice  
Edited by Farokh Marvasti

Principles of Digital Transmission: with Wireless Applications  
Sergio Benedetto and Ezio Biglieri

Simulation of Communication Systems, Second Edition: Methodology,  
Modeling, and Techniques  
Micheal C. Jeruchim, Phillip Balaban and K. Sam Shanmugan

Wireless Networks  
Multiuser Detection in Cross-Layer Design  
Christina Comaniciu, Narayan B. Mandayam and H. Vincent Poor

# Digital Signal Processing for Measurement Systems

Theory and Applications

Gabriele D'Antona

*Dipartimento di Elettrotecnica – Politecnico di Milano  
Milan, Italy*

Alessandro Ferrero

*Dipartimento di Elettrotecnica – Politecnico di Milano  
Milan, Italy*

 Springer

Gabriele D'Antona  
Alessandro Ferrero  
Department of Electrical Engineering  
Politechnic University of Milan  
Milano 20133  
Italy  
gabriele.dantona@polimi.it  
alessandro.ferrero@polimi.it

Library of Congress Control Number: 2005927075

ISBN-10: 0-387-24966-4      e-ISBN 0-387-28666-7  
ISBN-13: 978-0387-24966-7

Printed on acid-free paper.

© 2006 Springer Science+Business Media, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaption, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.      (SPI/EB)

9 8 7 6 5 4 3 2 1

springeronline.com

To our families

# CONTENTS

Contributors .....	xiii
Preface .....	xv
CHAPTER 1 – THE DIGITAL SIGNAL PROCESSING.....	1
CHAPTER 2 – DISCRETE-TIME SIGNALS AND SYSTEMS .....	5
Introduction.....	5
The sequences.....	6
DEFINITION.....	6
PARTICULAR SEQUENCES.....	7
PERIODIC SEQUENCES.....	9
OPERATIONS BETWEEN SEQUENCES.....	11
The discrete-time systems.....	12
DEFINITION.....	12
THE LINEAR SYSTEMS.....	13
THE SHIFT-INVARIANT LINEAR SYSTEMS.....	13
THE CONVOLUTION COMPUTATION.....	16
STABILITY AND CAUSALITY.....	22

Frequency-domain representation of the discrete-time signals and systems.....	24
THE FREQUENCY RESPONSE OF THE LINEAR, SHIFT-INVARIANT SYSTEMS.....	24
THE FOURIER TRANSFORM OF A SEQUENCE.....	29
SYMMETRY PROPERTIES OF THE FOURIER TRANSFORM.....	31
 CHAPTER 3 – SIGNAL TRANSFORMATION FROM THE CONTINUOUS TIME TO THE DISCRETE TIME DOMAIN.....	 33
Introduction.....	33
The sampling theorem.....	35
THE GENERAL FORMULATION.....	35
Shannon interpolation formula.....	41
THE CASE OF THE PERIODIC SIGNALS.....	43
Fourier-series approach.....	43
Convolution approach.....	47
The interpolation formula.....	51
The leakage errors.....	52
 CHAPTER 4 – METHODS FOR THE REDUCTION OF THE LEAKAGE ERRORS..	 57
Introduction.....	57
The weighing windows.....	58
THE RECTANGULAR WINDOW.....	58
THE COSINE WINDOWS.....	60
Cosine windows with maximum asymptotic decay.....	63
Cosine windows with minimum amplitude for the second lobe.....	67
Flat-top cosine windows.....	70
THE SELECTION OF THE OBSERVATION INTERVAL.....	72
The interpolation methods.....	74
TIME-DOMAIN INTERPOLATION METHOD.....	74
FREQUENCY-DOMAIN INTERPOLATION METHOD.....	76
Hardware synchronization.....	80
 CHAPTER 5 – ARCHITECTURE AND PERFORMANCE OF DSP-BASED INSTRUMENTS.....	 83
Introduction.....	83
The analog-to-digital conversion.....	84
DEFINITIONS.....	84



THE IDEAL ADC CHARACTERISTIC .....	85
The dithering technique .....	88
THE REAL ADC CHARACTERISTIC .....	91
The offset error .....	91
The gain error .....	92
The non-linearity error .....	94
ADC BEHAVIOUR IN THE PRESENCE OF A VARIABLE INPUT SIGNAL .....	96
The Sample & Hold devices .....	97
ADC BEHAVIOUR IN THE PRESENCE OF NOISE.....	101
ADC ARCHITECTURES .....	104
The integrating converters.....	104
The successive approximation converters .....	104
The flash converters .....	105
The $\Delta\Sigma$ converters .....	105
The processing unit .....	106
THE PROCESSOR.....	107
THE MEMORY.....	109
Bandwidth estimation .....	110
SINGLE-CHANNEL STRUCTURES .....	110
MULTI-CHANNEL STRUCTURES .....	111
The non-multiplexed structure.....	111
The multiplexed structure.....	112
CHAPTER 6 – ANTI-ALIASING METHODS AND SAMPLING STRATEGIES.....	115
Introduction.....	115
The anti-aliasing filter.....	115
The Heterodyne conversion .....	119
The equivalent-time sampling technique .....	122
Multi converter structures .....	124

CHAPTER 7 – DYNAMICAL DISCRETE SYSTEMS.....	127
Introduction.....	127
Dynamical discrete systems.....	127
$n$ -domain representation of dynamical discrete systems: difference equations.....	129
Stability of linear shift invariant discrete systems.....	135
The $z$ -transform.....	141
EXAMPLES OF Z-TRANSFORMS.....	152
The inverse $z$ -transform.....	155
EXAMPLES OF INVERSE Z-TRANSFORMS.....	156
$z$ -domain representation of causal discrete linear shift invariant systems.....	162
Linear shift invariant discrete filters.....	166
Relationships between sequences and sampled functions in the $z$ and $s$ domains.....	169
CHAPTER 8 – FINITE IMPULSE RESPONSE (FIR) FILTERS.....	179
$n$ -domain representation of FIR filters.....	179
$z$ -domain representation of FIR filters.....	180
Linear phase FIR filters.....	186
FIR filter synthesis by Fourier series expansion.....	192
FIR filter synthesis by the frequency sampling method.....	200
Effects of filter coefficients quantization.....	206
FIR filter architectures.....	209

CHAPTER 9 – INFINITE IMPULSE RESPONSE (IIR) FILTERS .....221

*z*-domain representation of IIR filters. ....221

*n*-domain representation of IIR filters.....222

Purely recursive IIR filters.....224

IIR filters design from analog specifications. ....230

    BUTTERWORTH FILTERS.....231

    CHEBYSHEV FILTERS .....232

    ELLIPTIC FILTERS.....233

    ANALOG TO DISCRETE DOMAIN MAPPING .....234

        Input response invariance.....235

        Frequency domain design.....240

        Frequency transformation from low-pass filters to band-pass and  
        band-stop filters .....250

IIR design by numerical optimization.....253

Effects of filter coefficients quantization.....255

IIR filter architectures.....259

## CONTRIBUTORS

Prof. Gabriele D'Antona, Ph.D., is Associate Professor of Electrical and Electronic Measurements at the Electrical Engineering Department of the "Politecnico di Milano" University, Milano, Italy. Previously, since 1991 to 1996, he joined the JET-Joint European Torus Laboratory in Abingdon, UK. His current research interests are principally concerned with the study of advanced variational and probabilistic signal processing techniques for the state estimation and control of distributed complex systems. Applications have been developed in several national and international programs for thermonuclear fusion research and environmental monitoring.

He is co-chair of the Environmental Measurements Technical Committee of the IEEE Instrumentation and Measurement Society and Scientific Collaborator of the Italian Institute of Plasma Physics.

Prof. Alessandro Ferrero was born in Milano, Italy, in 1954. He received the M.S. degree in Electrical Engineering from the "Politecnico di Milano", Milano, in 1978. In 1983 he joined the Dipartimento di Elettrotecnica of the Politecnico di Milano as an Assistant Professor on Electrical Measurements. From 1987 to 1991 he was Associate Professor of "Measurements on Electrical Machines and Systems" at the University of Catania, Catania, Italy. From 1991 to 1994 he was Associate Professor of Electrical Measurements at the Dipartimento di Elettrotecnica of the "Politecnico di Milano" University, Milano, Italy. He is presently Full Professor of Electrical and Electronic Measurements at the same Department. His current research interests are concerned with the application of Digital Signal Processing to Electrical Measurements, measurements on electric power systems and the qualification of the measurement process. He is a Fellow member of IEEE, member of AEIT (the Italian Association of Electrical and Electronic Engineers), member of ANIPLA (the Italian Association on Industrial Automation). He's presently the Chairman of the Italian Association on Electrical and Electronic Measurements, member of the North Italy Chapter of the IEEE IM Society and member of the AdCom of the IEEE Instrumentation and Measurement Society. He is the recipient of the 2006 IEEE Joseph F. Heithley Award in Instrumentation and Management.

# PREFACE

This book deals with the fundamentals of the Digital Signal Processing Theory, paying particular attention to its application to measurement issues. Therefore, the sampling theorem is thoroughly analyzed under different points of view, in order to emphasise its impact on the measurement accuracy. Different methods are also explained for the mitigation of the measurement errors that may arise from a not totally correct sampling strategy.

The discrete systems are also considered and analyzed, in order to frame the digital filters into a strict mathematical approach. The FIR and IIR filters are then introduced, and the basic guidelines for their specification, design and use are given.

This book is therefore mainly devoted to graduated and post-graduated students of technical faculties, and to professionals that want to have a closer look to the Digital Signal Processing theory in order to apply it to the practical situations in the correct and most effective way.

## THE DIGITAL SIGNAL PROCESSING

### *An introductory overview*

The digital signal processing finds its origins, as a well defined subject, during the seventeenth and eighteenth centuries, thanks also to the work of the two great mathematicians Newton and Gauss. At that time its applications were typically oriented to the numerical calculus, since this subject was mainly concerned with the representation of the mathematical functions by means of sequences of numbers, or symbols, and the elaboration of such sequences. In this respect, the classical numerical techniques, such as those used for the interpolation, integration and derivation of functions can be fully considered as Digital Signal Processing (DSP) techniques.

Only recently, approximately starting from the second half of the 70s, in the 20<sup>th</sup> century, the evolution of the computing devices and the Analog-to-Digital (AD) converters has greatly extended the application field of this subject.

One of the fields which most benefit of this evolution is that of measurement, and in particular the electric and electronic measurement. It is well known that the measurement activity typically consist in observing a physical phenomenon, both spontaneous (as is the case, for instance, of measurements of natural phenomena, such as those performed in astronomy or seismology), or artificially originated in order to perform the measurements (as is the case, for instance, of the test measurements on devices or machines, where the device under test is properly stimulated).

It is known as well that a physical phenomenon can be observed through the variation of physical quantities to which the information describing that phenomenon is associated. For instance, the presence of an electric field can be detected by analysing the distribution of the electric potential in the region of space where the electric field is present. Once a proper reference potential is taken, the observation of the electric field can be attained by means of measurements of potential differences. It is then possible to associate the information related to the observed physical phenomenon (the electric field) to the values taken by these potential differences and their variation in time.

In this example, the potential differences play the role of *measurement signals*, whose processing allows to extract the required information. The possible ways the measurement signals can be processed may be, in this case, for instance, the evaluation of the peak value, from which the maximum intensity of the field can be obtained, the evaluation of the rms value, to which the capability of the field to do an electric work can be associated, or the analysis of the spectral components, to which the field behaviour in the frequency domain can be associated.

The key point in a measurement process is therefore processing one or more measurement signals in order to extract the required information from the signals

themselves. This processing is generally described by a more or less complex mathematical relationship. Still considering the above example, the rms value of a potential difference  $u(t)$ , periodic with period  $T$  over time  $t$ , is expressed by the following relationship:

$$U = \sqrt{\frac{1}{T} \int_{t_0-T}^{t_0} u^2(t) dt}, \quad \forall t_0 \quad (1.1)$$

This result is not easy to obtain with the traditional analog techniques. In the past, electromechanical devices have been adopted, such as the electrodynamic instruments, where, due to the mechanical action caused by the interaction between currents, a driving torque, proportional to the square of the applied voltage, is exerted on a pivoted moving coil. The mechanical inertia of the moving coil acts as a low-pass filter, so that, in the presence of a spring restraining torque, the displacement angle of the moving coil is proportional to the mean value of the squared voltage. If a pointer is fixed on the moving coil rotation axes, its displacement on a square-law scale provides the desired rms value.

It can be easily understood that, if a good accuracy is required, the mechanical structure becomes critical and very expensive. When the analog electronic components became available, instruments based on electronic multipliers and low-pass filters could be realized, with a far better accuracy and wider bandwidth than the electromechanical instruments. However, the number of components required to process the input signal is still large, and therefore there is a large number of uncertainty sources to control in order to feature a good accuracy, which results in a high cost for the whole system. The problem becomes even more difficult to solve with the analog techniques when the measurement of quantities defined by more complex mathematical relationships are considered, such as, for instance, the spectral components of a signal.

On the other hand, the numerical calculus provides a relatively simple and direct solution to (1.1). In fact, if a device is available, able to sample  $u(t)$  with a constant sampling period  $T_c$ , convert the obtained samples into a digital code, store the obtained sequence of codes into a digital memory and process them digitally, it is possible to prove that, under a number of given conditions that will be discussed in the following chapters, (1.1) can be written as:

$$U = \sqrt{\frac{1}{T} \sum_k u^2(kT_c) T_c} \quad (1.2)$$

The block diagram of an instrument able to perform the above-listed operations is schematically drawn in Figure 1.1.

In such an instrument the result of the measurement is obtained by computing (1.2), with a proper program stored in the memory of the processor; this means that it is possible to perform different measurements, always starting from the same samples of  $u(t)$ , by simply changing the program, thus attaining also a dramatic cost reduction.

The above examples show that the operating principle of a digital instrument that employs DSP techniques is based on the instrument capability of sampling the input signals over time, converting the samples into digital values and processing them with a suitable algorithm in order to attain the result of the desired measurement.

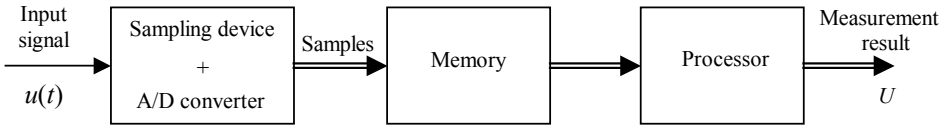


Figure 1.1. Block diagram for a measuring instrument based on the digital signal processing techniques.

When an analog signal is sampled and the samples are converted into digital values, the mathematical object *sequence* is obtained, defined in the *discrete-time* domain. This means that any sequence  $s(n)$  is defined only for *integer* values of the independent variable  $n$ , whilst a function  $f(t)$  of time  $t$  is defined for *real* values of the independent variable  $t$ , and is therefore defined in the *continuous-time* domain.

Since the obtained values are coded into a finite number of digits, generally representing a binary code, the input signal is allowed to assume only a finite number of values also in amplitude, thus giving rise to the *quantization* phenomenon.

A digital instrument, based on DSP techniques is hence characterized by a double transformation of the input signal from the continuous to the discrete domain, both in time and amplitude. This double transformation is the characterizing element of this kind of instruments, and differentiates them from another class of so-called digital instruments that process the input signals in the continuous time and amplitude domains and provide only the measurement result in a digital format.

The double domain transformation represents also the most critical part of the theoretical analysis of the DSP-based instruments, since the very first question to be answered is whether this double transformation modifies the information associated to the input signals or not.

This book is therefore aimed to analyse the mathematical theory that describes the two domain transformations, and in particular the transformation from the continuous to the discrete time, since this transformation is responsible for the most dangerous changes in the information associated with the input samples when nowadays sampling and converting devices are employed.

The devices in the block diagram in Fig. 1.1 are then considered, in order to evaluate their metrological characteristics and the way they contribute to the metrological performance of the whole DSP-based instrument.

At last the discrete systems are considered and analyzed in order to study a very important family of systems in the DSP technique: the digital filters.



Some of the considered items can be analysed only by means of advanced mathematical tools. Since this book is concerned with the digital processing of *measurement* signals rather than the mathematics of the discrete-time signals, the mathematical derivations will be limited to the amount that is strictly required to the correct comprehension of the presented theory. The readers are invited to refer to mathematics books if they wish to investigate the mathematical approach more in depth.

## DISCRETE-TIME SIGNALS AND SYSTEMS

### INTRODUCTION.

As shown in the previous chapter, the signals are the sources of the measurement information. Generally, the systems that are dealt with in practical situations evolve in time with continuity. Due to this property, they can be mathematically represented by functions of the independent variable *time* that belongs to the set of the *real numbers*. For this reason, these functions are generally referred to as belonging to the *continuous time* domain, and the signals are called *continuous-time signals*.

On the other hand, situations exist where the signals do not evolve with continuity in their appertaining domain. This is the typical case of the signals that represent quantities in the quantum mechanics, and it is also the more simple case of the periodic signals when they are represented in the frequency domain. The Fourier theory shows that, in this case, the signal is defined, in the frequency domain, only for discrete values of the independent variable frequency.

When the independent variable is time and the signal is defined only for discrete values of the independent variable, the signal is defined as belonging to the *discrete time* domain and is synthetically called a *discrete-time signal*. When the independent variable takes only discrete values, it sweeps over its axis by *quanta*; therefore it can be represented only by *integer numbers*, that represent actually the serial number of the quantum. For this reason, the independent variable of the mathematical object that represents a discrete-time signal belongs to the set of the integer numbers; the mathematical object is called a *sequence*.

An example of discrete-time signal is provided by a signal obtained by sampling a continuous-time signal with a constant sampling period. Usually, most discrete-time signals are obtained by sampling continuous-time signals. Anyway, for the sake of generality, a discrete-time signal can be seen as generated by a process defined in the discrete time domain.

For this reason, the discrete-time signals will be analysed by their own, without referring to their possible origin in a sampling operation. In this way the properties of the discrete-time signals can be fully perceived, and the mathematical tools can be defined that are required to provide an answer to the fundamental question of the digital signal processing theory: how and with which changes the information associated to a continuous-time signal is transferred to a discrete-time signal by sampling the continuous-time signal?

THE SEQUENCES.

**Definition.**

A discrete-time signal is mathematically represented by a *sequence* of values  $x$ :

$$x = \{x(n)\}, \quad -\infty \leq n \leq \infty \tag{2.1}$$

The  $n$ -th value  $x(n)$  in the sequence is also called the  $n$ -th *sample* of the sequence.

As already stated, the independent variable  $n$  belongs to the set of the integer numbers; this means that  $\{x(n)\}$  is mathematically defined only for integer values of  $n$ , and is not defined for non-integer<sup>1</sup> values of  $n$ .

Fig. 2.1 shows an example of graphical representation of a sequence.

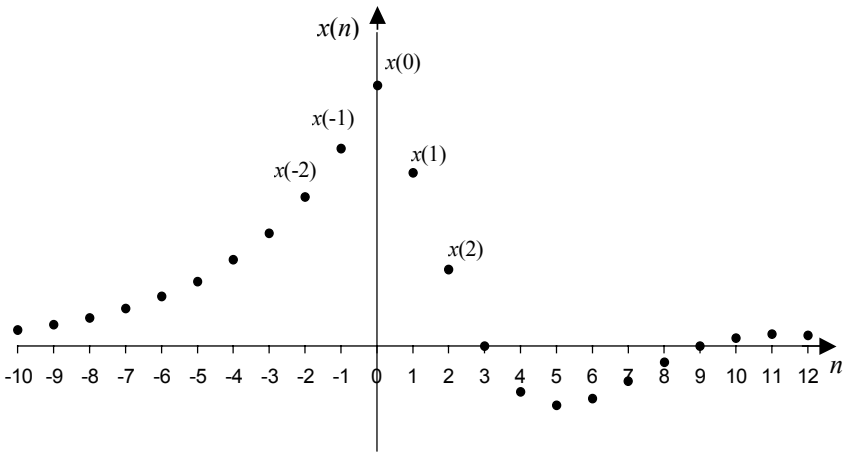


Figure 2.1. Graphical representation of a sequence

Sometimes, when the whole sequence cannot be confused with its own samples, the notation in (2.1) is uselessly complex, and therefore the whole sequence is referred to as  $x(n)$ .

---

<sup>1</sup> It is a common mistake stating that  $x(n) = 0$  for non-integer values of  $n$ . This is absolutely incorrect, since a sequence cannot be mathematically defined for non-integer values of its independent variable  $n$ .

## Particular sequences.

Some sequences play a very important role in the analysis of the discrete-time signals. The most important is the *unit sample* sequence, which is mathematically defined as:

$$\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases} \quad (2.2)$$

and is graphically represented in Fig. 2.2.

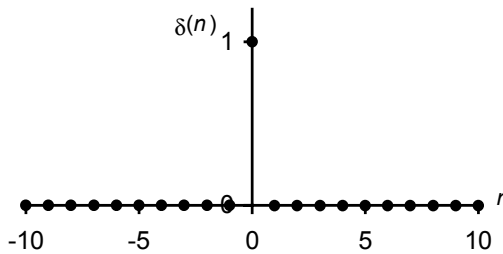


Figure 2.2. Unit sample sequence

This sequence is important because it plays, in the analysis of the discrete-time signals and systems, a similar role as that played by the Dirac impulse in the analysis of the continuous-time signals and systems. For this reason, the unit sample sequence is often, though not properly, called *discrete-time impulse*, or *impulse sequence*. It is worth while noting that its mathematical definition is very simple, if compared with the definition of the Dirac impulse.

A second important sequence is the *unit step* sequence. It is graphically shown in Fig. 2.3, and is mathematically defined as:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (2.3)$$

Similarly to the unit sample, the unit step too plays, in the discrete-time domain, the same role played by the step function in the continuous-time domain.

According to (2.1) and (2.3), the unit step sequence can be written in terms of the unit sample sequence as:

$$u(n) = \sum_{k=-\infty}^n \delta(k) \quad (2.4)$$

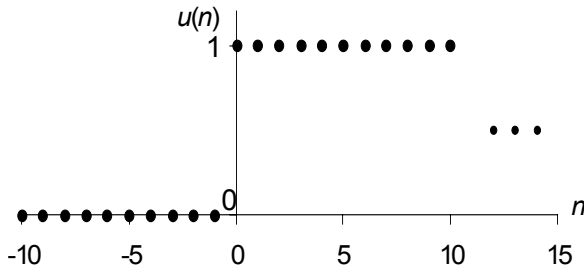


Figure 2.3. Unit step sequence

This can be proved by recalling that, due to (2.2), the elements in the sum (2.4) are non-zero (and equal to 1) only if  $k = 0$ . For negative values of  $n$ , all elements in the sum (2.4) are zero, and therefore  $u(n) = 0$  for  $n < 0$ . For  $n \geq 0$  only one non-zero term exists, for each value of  $n$ , and it is equal to 1 for  $k = 0$ . Therefore  $u(n) = 1$  for  $n \geq 0$ , and 2.4 is proved.

The inverse relationship, linking the unit sample sequence to the unit step sequence, can be readily written as:

$$\delta(n) = u(n) - u(n-1) \tag{2.5}$$

Fig. 2.4 shows the *real exponential* sequence, whose mathematical expression is:

$$x(n) = a^n \tag{2.6}$$

with  $a$  a real number.

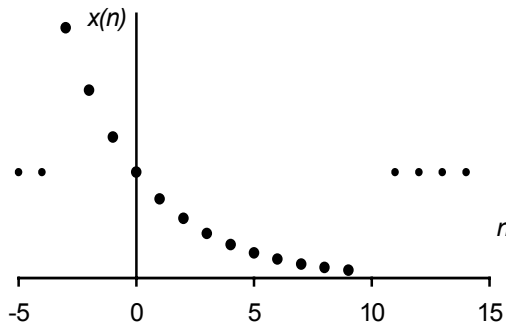


Figure 2.4. Real exponential sequence  $x(n) = a^n$ , with  $a < 1$

More generally, the *complex exponential* sequence can be defined as:

$$x(n) = e^{(\sigma + j\omega_0)n} \quad (2.7)$$

where  $\sigma$  and  $\omega_0$  are real numbers.

The *sinusoid* sequence can be obtained from the complex exponential sequence for  $\sigma = 0$ . This sequence is graphically represented in Fig. 2.5 and can be mathematically written also as:

$$x(n) = A \cos(\omega_0 n + \varphi) \quad (2.8)$$

where  $A$ ,  $\omega_0$  and  $\varphi$  are still real numbers.

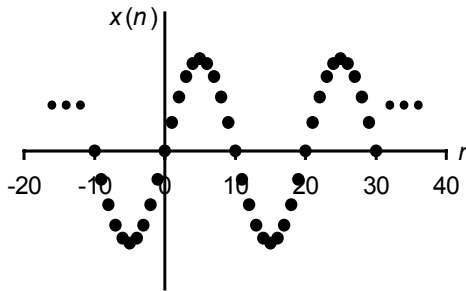


Figure 2.5. Sinusoid sequence

## Periodic sequences.

When the continuous-time signals are considered, the sinusoid function and the complex exponential function with  $\sigma = 0$  are periodic in time. Similarly, when the discrete-time signals are considered, the sinusoid sequence and the complex exponential sequence with  $\sigma = 0$  could be expected to be periodic too. However, since in this last case the independent variable  $n$  belongs to the set of the integer numbers, these sequences could not be periodic.

The definition of a periodic sequence  $x(n)$ , over the period  $N$ , is given by the following relationship:

$$x(n) = x(n + N), \quad \forall n \quad (2.9)$$

Since the independent variable of a sequence must belong to the set of the integer numbers, period  $N$  must be an integer too. Therefore, the sinusoid sequence and complex exponential sequence with  $\sigma = 0$  are periodic with period  $2\pi/\omega_0$  only if this quantity is an integer number. If  $2\pi/\omega_0$  is not an integer, but is a rational number, these sequences

are still periodic, but their period will be an integer multiple of  $2\pi/\omega_0$ . If  $2\pi/\omega_0$  is neither a rational number, these sequences are not periodic.

This property of the periodic sequences is very important, because it explains how the information associated to a continuous-time periodic signal can be modified after the signal itself has been sampled.

Let  $s(t)$  be a sinusoidal, continuous-time signal, with period  $T$ . This signal is described by the following equation:

$$s(t) = A \sin\left(\frac{2\pi}{T}t + \varphi\right) \quad (2.10)$$

Let  $s(t)$  be sampled with  $T_s$  sampling period, so that:

$$T/T_s = N, \quad N \text{ an integer} \quad (2.11)$$

Eq. (2.10), evaluated at each sampling time  $kT_s$ , provides:

$$s(kT_s) = A \sin\left(2\pi \frac{kT_s}{T} + \varphi\right),$$

that, taking into account (2.11), leads to the following sequence:

$$\hat{s}(k) = A \sin\left(\frac{2\pi}{N}k + \varphi\right) \quad (2.12)$$

It can be readily checked that, in (2.12),  $\omega_0 = \frac{2\pi}{N}$  and therefore, having supposed  $N$  an integer, the sequence in (2.12) is periodic with period  $N$ . It can be hence concluded that the sequence obtained by sampling the periodic signal  $s(t)$  is still periodic and the relationship between the period of the continuous-time signal and that of the discrete-time signal is given by:  $T = NT_s$ .

On the other hand, signal (2.10) could have been sampled with a sampling period  $T'_s$  so that:  $T = mT'_s$  with  $m$  a real number. In this case (2.12) would have become:

$$\hat{s}(k) = A \sin\left(\frac{2\pi}{m}k + \varphi\right)$$

and, not being  $m$  an integer or a rational number, this sequence is not periodic. It can be concluded that the sampling operation, in this case, has changed the information associated with the continuous-time signal, because a non-periodic sequence has been obtained from a periodic signal. As it will be shown in the next chapter, condition (2.11)

is one of the conditions that must be satisfied in order to sample a periodic signal correctly.

Coming back to the sinusoid and complex exponential sequences, the  $\omega_0$  parameter is called the *angular frequency* (and  $f_0 = \frac{\omega_0}{2\pi}$  the *frequency*) of the sequences, no matter whether they are periodic or not.

## Operations between sequences.

The sum and product operations can be defined for the sequences. In particular, the sum of two sequences,  $x$  and  $y$ , is a sequence whose samples are obtained as the sum of the corresponding samples of the two sequences. From the mathematical point of view, it is:

$$x + y = \{x(n) + y(n)\} \quad (2.13)$$

Similarly, the product of two sequences is defined as a sequence whose samples are obtained as the product of the corresponding samples of the two sequences. From the mathematical point of view, it is:

$$x \cdot y = \{x(n) \cdot y(n)\} \quad (2.14)$$

It can be easily proven that the sum and product operations between sequences satisfy the commutative and associative properties, and the product satisfies also the distributive property with respect to the sum.

The product of a sequence  $x$  by a number  $a$  is defined as the sequence whose samples are those of sequence  $x$ , each multiplied by  $a$ . It is:

$$x \cdot a = \{a \cdot x(n)\} \quad (2.15)$$

At last, the shift operation is defined as the operation that, starting from a sequence  $x$ , provides its replica, shifted along the  $n$  axis by a given integer quantity  $n_0$ ; from the mathematical point of view, this can be written as:

$$y(n) = x(n - n_0) \quad (2.16)$$

When  $n_0 > 0$  the original sequence is shifted on the right, over the  $n$  axis, whilst when  $n_0 < 0$  it is shifted on the left.

If shifted versions of the unit sample sequence are employed, any sequence can be expressed in terms of the unit sample sequence. Given a sequence  $x(n)$ , it is indeed possible to write:



$$x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k) \quad (2.17)$$

The above relationship can be readily proven by considering that the shifted unit sample sequence  $\delta(n-k)$  is non zero (and equal to 1) only for  $n - k = 0$ , and hence for  $n = k$ . Therefore, the sum at the right-hand side of (2.17) has only one non-zero term, for  $n = k$ , whose value is equal to sample  $x(n)$ ,  $\forall n$ . Eq. (2.17) is then proven.

## THE DISCRETE-TIME SYSTEMS.

### Definition.

A system, whichever is the domain it belongs to, is mathematically defined as a unique transformation that maps an input quantity  $x$  into an output quantity  $y$ . In the discrete time domain, the system's input and output quantities are sequences, which means that a discrete-time system transforms the input sequence  $x(n)$  into the output sequence  $y(n)$  univocally. As a matter of fact, a discrete-time transformation can be seen as an algorithm that processes the samples of the input sequence in order to provide the samples of the output sequence.

From the mathematical point of view, a transformation is expressed as:

$$y(n) = T[x(n)]$$

and is graphically represented as shown in Fig. 2.6.

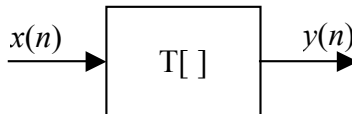


Figure 2.6. Graphical representation of a system and related transformation

The output sequence  $y(n)$  is also called the *response* of system  $T$  to the input sequence  $x(n)$ .

The analysis of a generic system and its related transformation is generally quite complex, unless suitable constraints are introduced that limit the freedom degrees of the system itself, by defining suitable classes of transformations. The most important class is that of the linear systems, to which the overwhelming majority of the algorithms employed in the measurement field belongs.

## The linear systems.

The linear systems can be mathematically defined by means of the superposition principle. Given the two input sequences  $x_1(n)$  and  $x_2(n)$ , let  $y_1(n)$  and  $y_2(n)$  be the responses of a system T to the given input sequences respectively, so that:

$$y_1(n) = T[x_1(n)] \quad \text{and} \quad y_2(n) = T[x_2(n)]$$

According to the superposition principle, system T is linear if and only if:

$$T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)] = ay_1(n) + by_2(n) \quad (2.18)$$

where  $a$  and  $b$  are arbitrary constants.

Since, according to (2.17), any sequence can be written in terms of the unit sample sequence, the response of a system can be written as:

$$y(n) = T \left[ \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k) \right] \quad (2.19)$$

If system T is linear, (2.18) applies and (2.19) can be written as:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)T[\delta(n-k)] \quad (2.20)$$

Let  $h_k(n)$  be the system response to the shifted unit sample sequence  $\delta(n-k)$ ; (2.20) can be written as:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h_k(n) \quad (2.21)$$

This last equation shows that the response of a linear system to an input sequence depends on response  $h_k(n)$  to the shifted unit sample sequence. However, this response depends on both  $n$  and  $k$ , thus limiting the practical utility of (2.21).

## The shift-invariant linear systems.

An important sub-set of the linear systems is that of the *shift-invariant* linear systems. These systems represent the counterpart, in the discrete time domain, of the time-invariant systems in the continuous time domain, and have the property that, being  $y(n)$

the response to the input sequence  $x(n)$ , the response to the shifted sequence  $x(n - k)$  is  $y(n - k)$ , that is the same response as that to sequence  $x(n)$ , shifted by the same quantity  $k$ .

Due to this property, it can be immediately proved that, if  $h(n)$  is the system response to the unit sample sequence  $\delta(n)$ , then the response to the shifted unit sample sequence  $\delta(n - k)$  is  $h(n - k)$ . For the shift-invariant linear systems, eq. (2.21) becomes:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n - k) \quad (2.22)$$

Therefore, it can be stated that a shift-invariant linear system is fully characterized by its response  $h(n)$  to the unit sample sequence. Eq. (2.22) shows that, once  $h(n)$  is known, it is possible to determine the response to any other input sequence  $x(n)$ .

Eq. (2.22) is called the *convolution sum*, and the resulting sequence  $y(n)$  is called the *convolution* of the two sequences  $x(n)$  and  $h(n)$ . The usual notation for the convolution operation is:

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

It can be readily proved that the convolution sum satisfies the commutative property, that is:

$$y(n) = x(n) * h(n) = h(n) * x(n) \quad (2.23)$$

To prove this property, let's substitute variable  $k$  with  $k = n - m$  in (2.22), that becomes:

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

For a given value of  $n$ , having quantity  $n - m$  varying from  $-\infty$  to  $+\infty$  is the same as having quantity  $m$  varying in the same interval. Therefore, we get:

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

which proves (2.23).

Having proved that the convolution sum satisfies the commutative property leads to the conclusion that the response of a shift-invariant linear system does not change if the input sequence and the response to the unit sample sequence are interchanged. In other words, the response of a shift-invariant linear system, featuring  $h(n)$  as the response to the

unit sample sequence, to sequence  $x(n)$  is the same as that of a system, featuring  $x(n)$  as the response to the unit sample sequence, to sequence  $h(n)$ .

The convolution sum satisfies also the associative property. Considering three generic sequences  $x(n)$ ,  $y(n)$  and  $w(n)$ , it can be proven that:

$$x(n) * [y(n) * w(n)] = [x(n) * y(n)] * w(n) \quad (2.24)$$

In order to prove (2.24), let  $\vartheta(n) = y(n) * w(n)$ , so that:

$$\begin{aligned} x(n) * \vartheta(n) &= \sum_{k=-\infty}^{+\infty} x(k) \vartheta(n-k) = \sum_{k=-\infty}^{+\infty} x(k) \sum_{j=-\infty}^{+\infty} y(j) w(n-k-j) = \\ &= \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} x(k) y(j) w(n-k-j) \end{aligned} \quad (2.25)$$

Similarly, let  $\xi(n) = x(n) * y(n)$ , so that, taking into account the commutative property of the convolution sum:

$$\begin{aligned} \xi(n) * w(n) &= \sum_{k=-\infty}^{+\infty} w(k) \xi(n-k) = \sum_{k=-\infty}^{+\infty} w(k) \sum_{j=-\infty}^{+\infty} y(j) x(n-k-j) = \\ &= \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} w(k) y(j) x(n-k-j) \end{aligned}$$

If the variable  $m = n - k - j$  is introduced, so that  $k = n - m - j$ , the above equation can be rewritten as:

$$\begin{aligned} \xi(n) * w(n) &= \sum_{n-m-j=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} w(n-m-j) y(j) x(m) = \\ &= \sum_{m=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} x(m) y(j) w(n-m-j) \end{aligned} \quad (2.26)$$

Since the right-hand terms of (2.25) and (2.26) are equal, the associative property is proved.

According to this property, it is possible to prove that the response to the unit sample sequence of a system realized by cascading two systems, as shown in Fig. 2.7, is given by the convolution of the responses of the two single systems to the unit sample sequence.

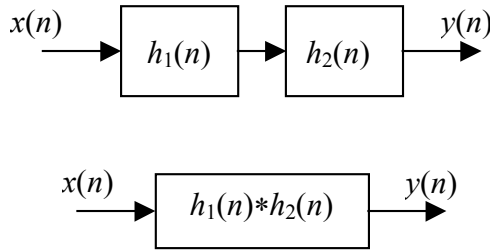


Figure 2.7. Cascaded shift-invariant linear systems

Let  $y_1(n)$  be the output of the first sub-system and  $h_1(n)$  its response to the unit sample sequence; it is:

$$y_1(n) = x(n) * h_1(n)$$

that leads to write the output  $y(n)$  of the whole system as:

$$y(n) = y_1(n) * h_2(n) = [x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)] \quad (2.27)$$

proving that the system response to the unit sample sequence is given by the convolution of the single responses.

Taking into account also the commutative property of the convolution sum, it can be stated that the system output is independent on the order with which the two sub-systems are cascaded. Since a discrete-time system is, as a matter of fact, an algorithm, this means that, if a signal processing procedure consists of two or more algorithms executed in series, the order with which the algorithms are executed does not affect the final result.

At last, it can be immediately checked that the convolution sum satisfies also the distributive property with respect to the addition. For this reason, if two shift-invariant linear systems are connected in parallel, as shown in Fig. 2.8, the response of the whole system to the unit sample sequence is given by the sum of the responses of the single systems. The proof of this last property is very simple and is left to the reader.

### The convolution computation.

The convolution sum plays, for the linear systems in the discrete time domain, a similar role as that played by the convolution integral for the linear systems in the continuous time domain. They are both extremely important in the theoretical analysis of this kind of systems. However, the convolution integrals are often difficult to solve, so that their importance is generally confined to the theoretical analysis, being the practical application limited to a few particular situations.

On the other hand, the convolution sum can be computed in a far less difficult way and its computation is quite immediate when the two sequences have a finite length, as in

many practical cases. This extends the importance of the convolution sum from the theoretical analysis to the practical applications, assigning it an even greater importance than that assigned to the convolution integral; moreover this is one of the reasons for the digital techniques are nowadays preferred to the analog ones.

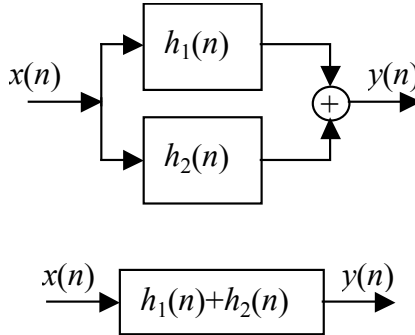


Figure 2.8. Shift-invariant linear systems connected in parallel

The convolution sum represents also a very important tool to understand some fundamental concepts about signal sampling and the way the sampling operation may generate sequences carrying a different information than the one provided by the continuous-time signal. For this reason, the way the convolution sum of two sequences is computed is now described into details.

Let us consider the two sequences in Fig. 2.9:

$$x(n) = \begin{cases} 1, & 0 \leq n \leq 4 \\ 0, & \text{otherwise} \end{cases} \quad (2.28)$$

and

$$y(n) = \begin{cases} 0.5, & 0 \leq n \leq 4 \\ 0, & \text{otherwise} \end{cases} \quad (2.29)$$

According to definition (2.22), the convolution of these two sequences is given by:

$$w(n) = \sum_{k=-\infty}^{+\infty} x(k)y(n-k) \quad (2.30)$$

The first step in the computation of the convolution is, according to (2.30), the generation of sequence  $y(-k)$ . It can be immediately seen that this sequence is obtained by

reversing  $y(k)$  with respect to the vertical axes, as shown in Fig. 2.10 where both sequences  $x(k)$  and  $y(-k)$  are drawn.

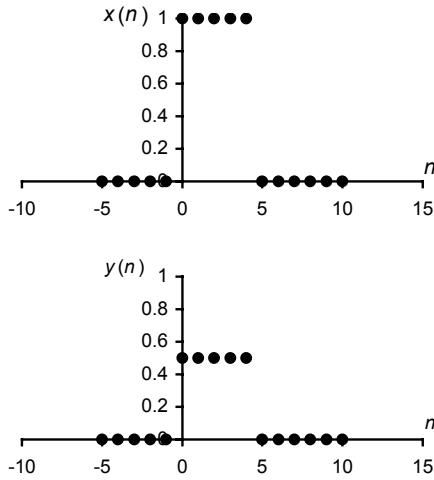


Figure 2.9.  $x(n)$  and  $y(n)$  sequences

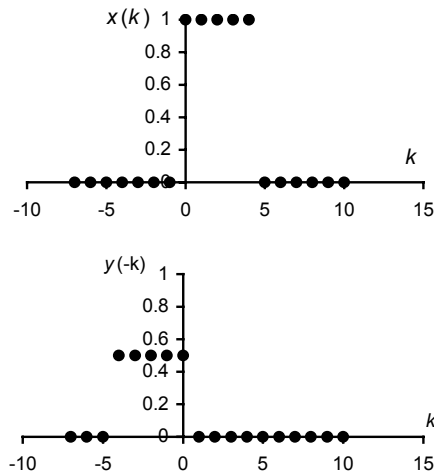


Figure 2.10.  $x(k)$  and  $y(-k)$  sequences

The next step is the generation of sequence  $y(n - k)$ , obtained, for every value of  $n$ , by shifting sequence  $y(-k)$  by  $n$ . It can be immediately checked that, for the considered sequences (2.28) and (2.29), their convolution  $w(n)$  in (2.30) has non-zero values only for  $0 \leq n \leq 9$ , since the shifted sequence  $y(n - k)$  has non-zero values where  $x(n)$  has non-zero values only in this range of  $n$  values.

To better understand this point, let us consider the case of  $n = -1$ . Fig. 2.11 shows the resulting sequences  $x(k)$  and  $y(-1 - k)$ .

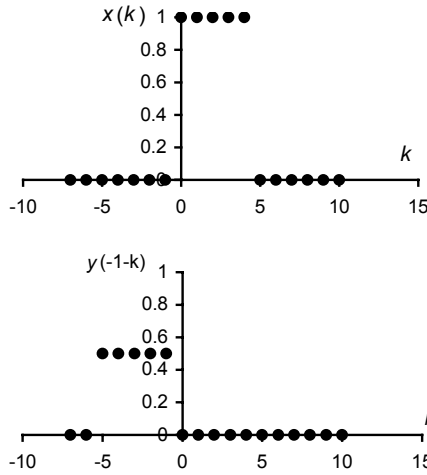


Figure 2.11. Convolution step for  $n = -1$

It can be immediately seen that all products  $x(k)y(-1 - k)$  are zero, for every value of  $k$ , so that sample  $w(-1)$  in the convolution will be zero too. The above applies to every value  $n \leq -1$ .

Let us now consider the case of  $n = 0$ . The two resulting sequences  $x(k)$  and  $y(-k)$  are represented in Fig. 2.10, and it can be seen that the only non-zero product in (2.30) is  $w(0) = x(0)y(0) = 0.5$ , which represents the convolution value for  $n = 0$ .

When  $n = 1$ , the two resulting sequences  $x(k)$  and  $y(1 - k)$  are represented in Fig. 2.12. It can be seen that only two products in (2.30) have non-zero values:

$$x(0)y(0) = 0.5$$

and

$$x(1)y(1) = 0.5$$

Therefore, the convolution sample for  $n = 1$  is given by:



$$w(1) = 0.5 + 0.5 = 1$$

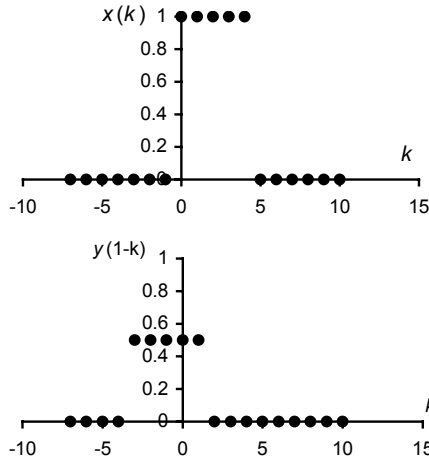


Figure 2.12. Convolution step for  $n = 1$

It is then possible to verify, in a quite similar way, that:

$$w(2) = 0.5 + 0.5 + 0.5 = 1.5$$

$$w(3) = 0.5 + 0.5 + 0.5 + 0.5 = 2$$

When  $n = 4$ , the non-zero samples of the resulting sequences  $x(k)$  and  $y(4 - k)$  take the same position, as shown by Fig. 2.13. Therefore:

$$w(4) = 0.5 + 0.5 + 0.5 + 0.5 + 0.5 = 2.5$$

and the convolution reaches its maximum value.

This can be readily understood looking at Fig. 2.14, which shows the resulting sequences  $x(k)$  and  $y(5 - k)$  for the case of  $n = 5$ . In this case, only 4 non-zero samples of sequence  $x(k)$  are in the same position as the non-zero samples of sequence  $y(5 - k)$ , so that the corresponding convolution sample will be:

$$w(5) = 0.5 + 0.5 + 0.5 + 0.5 = 2.$$

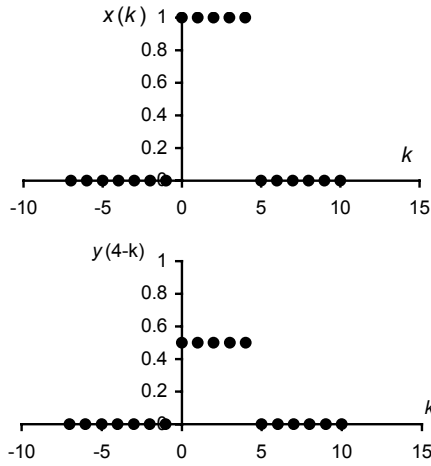


Figure 2.13. Convolution step for  $n = 4$

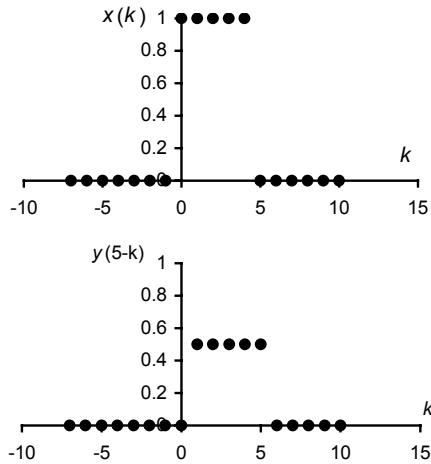


Figure 2.14. Convolution step for  $n = 5$

The remaining samples of  $w(n)$  can be computed in a quite similar way, obtaining:

$$w(6) = 0.5 + 0.5 + 0.5 = 1.5,$$

$$w(7) = 0.5 + 0.5 = 1,$$

$$w(8) = 0.5.$$

At last, it can be easily checked that, for  $n = 9$ , none of the non-zero samples of the resulting sequence  $y(9 - k)$  is in the same position as the non-zero samples of sequence  $x(k)$ , so that  $w(9) = 0$ . The same applies for  $n > 9$ .

The convolution  $w(n) = x(n) * y(n)$  is hence the sequence shown in Fig. 2.15.

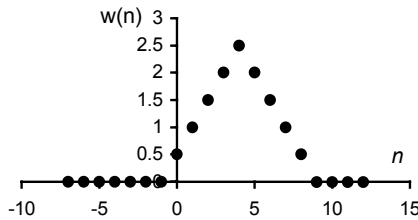


Figure 2.15. Convolution result

This figure shows that the obtained sequence has a longer length than that of the two starting sequences. This example can be generalized, showing that the convolution of two sequences, with finite length of  $N$  and  $M$  samples respectively, will be again a finite length sequence, with length equal to  $N + M - 1$  samples.

### Stability and causality.

The previous paragraphs have defined an important class of discrete-time systems by introducing the linearity and shift invariance constraints. Two new constraints are now introduced, the *stability* and *causality*, that lead to the definition of two new sets of linear, shift-invariant systems, very important in the study of the digital filters, as it will be shown in the next chapters.

A system is *stable* when, for any input sequence with *bounded amplitude*, it provides an output sequence that is bounded in amplitude too. The necessary and sufficient condition to ensure that a linear, shift invariant system is stable is that its response  $h(n)$  to the unit sample sequence meets the following condition:

$$\sum_{k=-\infty}^{+\infty} |h(k)| = S < \infty \tag{2.31}$$

It is possible to prove that this condition is sufficient to ensure the system stability by considering a bounded input sequence  $x(n)$ , so that  $|x(n)| < M, \forall n$ . The system output

sequence can be obtained by applying (2.22); supposing that condition (2.31) is also satisfied, it can be written:

$$|y(n)| = \left| \sum_{k=-\infty}^{+\infty} h(k)x(n-k) \right| \leq M \sum_{k=-\infty}^{+\infty} |h(k)| < \infty$$

thus proving that the output sequence is also bounded in amplitude.

It can be now proved that condition (2.31) is necessary to ensure stability by proving that, if  $S = \infty$ , an input sequence can be found with bounded amplitude that gives rise to an output sequence with unbounded amplitude. Let us consider a linear, shift-invariant system characterized by a response  $h(n)$  to the unit sample sequence, and let us consider an input sequence:

$$x(n) = \begin{cases} \frac{h^*(-n)}{|h(-n)|}, & h(n) \neq 0 \\ 0, & h(n) = 0 \end{cases}$$

where  $h^*(n)$  is the complex conjugate of  $h(n)$ .  $x(n)$  is bounded because of the way it has been defined. If the output sequence  $y(n)$  is now evaluated by applying (2.22), its value for  $n = 0$  is given by:

$$y(0) = \sum_{k=-\infty}^{+\infty} x(-k)h(k) = \sum_{k=-\infty}^{+\infty} \frac{h^*(k)h(k)}{|h(k)|} = \sum_{k=-\infty}^{+\infty} \frac{|h(k)|^2}{|h(k)|} = \sum_{k=-\infty}^{+\infty} |h(k)| = S$$

Therefore, if  $S = \infty$ , the output sequence is unbounded even in the presence of a bounded input sequence. This proves that condition (2.31) is a necessary condition too.

A system is *causal* when its output  $y(n)$ , for every  $n = n_0$ , depends on the input samples for  $n \leq n_0$  only. For this reason, if  $x_1(n)$  and  $x_2(n)$  are two input sequences to a causal system, taken in such a way that  $x_1(n) = x_2(n)$  for  $n < n_0$ , then the system's output sequences  $y_1(n)$  and  $y_2(n)$  will feature  $y_1(n) = y_2(n)$  for  $n < n_0$ .

A linear, shift-invariant, causal system has the property that its response to the unit sample sequence is zero for  $n < 0$ . This can be proved by considering the system output, evaluated in  $n = n_0$ , in terms of (2.22) where the commutative property of the convolution has been considered too:

$$y(n_0) = \sum_{k=-\infty}^{+\infty} h(k)x(n_0 - k) \quad (2.32)$$

In (2.32) the samples of  $x(n)$  for  $n > n_0$  are obtained when  $k < 0$ . In order to prevent these samples to contribute to the sum in (2.32), and hence to the determination of  $y(n_0)$ , it must be  $h(k) = 0$  for  $k < 0$ , as stated.

For this reason a sequence with zero samples for  $n < 0$  is called causal. This is somehow justified by the fact that this sequence could be the response of a linear, shift-invariant, causal system to the unit sample sequence.

An example of linear, shift-invariant, causal and stable system is a system whose response to the unit sample sequence is:

$$h(n) = a^n u(n)$$

Due to the presence of the unit step sequence  $u(n)$ ,  $h(n) = 0$  for  $n < 0$ , so that the system is causal. In order to ensure the system stability, (2.31) must be verified, which means that the following quantity:

$$S = \sum_{k=-\infty}^{+\infty} |h(k)| = \sum_{k=0}^{+\infty} |a|^k \quad (2.33)$$

must be finite.

It can be noted that the quantity in the right-hand side of (2.33) is a geometrical series that converges to the finite quantity:

$$S = \frac{1}{1 - |a|}$$

for  $|a| < 1$ , otherwise the series diverges to infinite. Therefore, the system is stable only if  $|a| < 1$ .

## FREQUENCY-DOMAIN REPRESENTATION OF THE DISCRETE-TIME SIGNALS AND SYSTEMS.

### The frequency response of the linear, shift-invariant systems.

In the previous sections it has been shown how a linear, shift-invariant system can be fully characterized by its response to the unit sample sequence. This property of the discrete-time systems is the counterpart, in the discrete time, of the property of the continuous-time linear, time-invariant systems to be fully characterized by their impulse response.

Another fundamental properties of these continuous-time systems is that their steady-state response to a sinusoidal input is still a sinewave, with the same frequency as the input sinewave, with amplitude and phase determined by the system. It is because of this

property that the linear, time-invariant systems are usefully represented as a suitable combination of sinewaves or complex exponential signals (Fourier representation).

This property can be extended to the discrete-time systems in a straightforward way. Let us consider the following sequence as the input sequence of a discrete-time, linear, shift-invariant system:

$$x(n) = e^{j\omega n}, \quad -\infty < n < +\infty \quad (2.34)$$

It can be immediately recognized that this sequence is a complex exponential sequence with  $\sigma = 0$  and  $\omega$  angular frequency. According to (2.22), the output sequence of a linear, shift-invariant system with  $h(n)$  response to the unit sample sequence is:

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k) e^{j\omega(n-k)} = e^{j\omega n} \sum_{k=-\infty}^{+\infty} h(k) e^{-j\omega k} \quad (2.35)$$

Supposing that the series at the right-hand side of (2.35) is convergent, the following function is obtained:

$$H(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} h(k) e^{-j\omega k} \quad (2.36)$$

and (2.35) can be rewritten as:

$$y(n) = H(e^{j\omega}) e^{j\omega n} \quad (2.37)$$

or, taking into account (2.34):

$$y(n) = H(e^{j\omega}) x(n) \quad (2.38)$$

Equations (2.37) and (2.38) show that the complex exponential input sequence  $x(n)$ , due to the transformation T introduced by the linear, shift-invariant system, goes through a variation in its amplitude and phase only, variation that is described by the complex function  $H(e^{j\omega})$ .  $H(e^{j\omega})$  is a continuous function of the angular frequency  $\omega$  and is called the *frequency response* of the system whose response to the unit sample sequence is  $h(n)$ .

The above conclusion is valid, as mentioned in deriving (2.36), only if the response to the unit sample sequence  $h(n)$  allows (2.36) to converge to  $H(e^{j\omega})$ . If this is verified,  $H(e^{j\omega})$  is not only a continuous function of the angular frequency  $\omega$ , but it is also periodic in  $\omega$  with a  $2\pi$  period. This last property comes directly from (2.36), since:

$$e^{j(\omega+2\pi)k} = e^{j\omega k}$$

The fact that  $H(e^{j\omega})$  takes the same values for any  $\omega = \omega_0$  and  $\omega = \omega_0 + 2\pi$  means that the discrete-time system provides the same response to complex exponential sequences with these two angular frequency values. This is totally justified by the fact that the two complex exponential sequences do not differ.

As an example of the above conclusions, let us consider a linear, shift-invariant system, whose response to the unit sample sequence is given by:

$$h(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.39}$$

as shown in Fig. 2.16 for the particular case of  $N = 8$ .

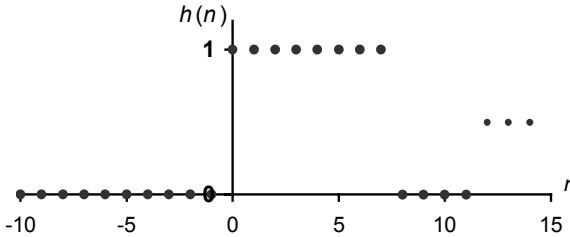


Figure 2.16. Response (2.39) to the unit sample sequence for  $N=8$

When (2.36) is applied, the following frequency response is obtained:

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} e^{-j\omega k} = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} = \frac{\sin(\omega N/2)}{\sin(\omega/2)} e^{-j(N-1)\omega/2} \tag{2.40}$$

Amplitude and phase of (2.40) are shown in Fig. 2.17 a and b respectively, for  $N = 8$ .

Since the frequency response (2.36) is a continuous, periodic function of  $\omega$ , it can be developed in terms of Fourier series. Actually, (2.36) expresses  $H(e^{j\omega})$  in terms of its Fourier series coefficients, which are equal to the samples of the unit sample sequence response  $h(n)$ . It is therefore possible to express this double relationship between the frequency response and the response to the unit sample sequence as:

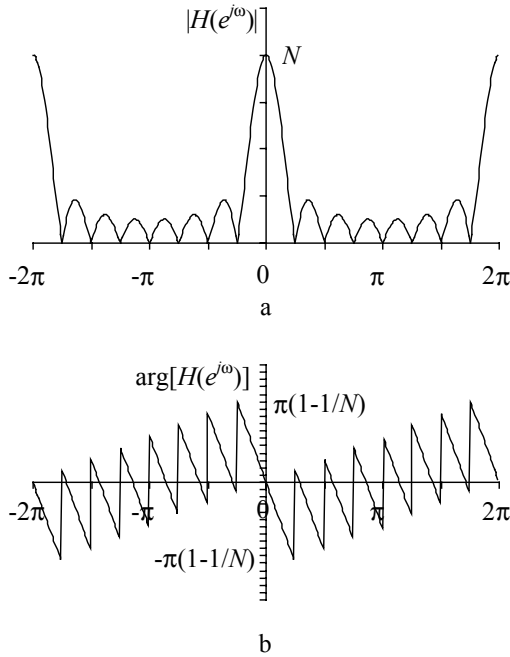


Figure 2.17. Amplitude (a) and phase (b) of the frequency response (2.40) for  $N=8$

$$\begin{aligned}
 h(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \\
 H(e^{j\omega}) &= \sum_{n=-\infty}^{+\infty} h(n) e^{-j\omega n}
 \end{aligned}
 \tag{2.41}$$

where the first equation represents the direct Fourier series analysis for  $H(e^{j\omega})$ , and the second equation represents the inverse Fourier synthesis equation.

These equations can be also interpreted in a rather different, though more interesting way. The first equation in (2.41) describes sequence  $h(n)$  in terms of elementary exponential components, whose complex amplitudes are determined by the second equation in (2.41). In this respect, the equations in (2.41) can be seen as, respectively, the inverse and direct Fourier transform pair for sequence  $h(n)$ .

This representation provides also a rule for establishing whether the series at the right-hand side of the second equation in (2.41) converges or not, and hence  $H(e^{j\omega})$  exists or not. According to the Fourier theory this convergence is ensured if:



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

that is if  $h(n)$  is *absolutely summable* and the series is *absolutely convergent*. In this situation, (2.36) or the second equation in (2.41), converges to a continuous function of  $\omega$ . Taking into account (2.31), this leads to the conclusion that it is always possible to define the frequency response of stable systems.

It can be also proved that, since:

$$\sum_n |h(n)|^2 \leq \left[ \sum_n |h(n)| \right]^2,$$

if  $h(n)$  is absolutely summable, it will be also:

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

which means that  $h(n)$  has *finite energy*. The contrary is not a priori verified, that is a finite energy sequence may not be absolutely summable. In this case, the Fourier theory proves that the Fourier transform of such a sequence still exists, but it will be a *generally continuous* function of  $\omega$ , instead of being a continuous function.

This is the case of the following sequence:

$$h(n) = \frac{\sin(\omega_0 n)}{\pi n} \tag{2.42}$$

which is shown in Fig. 2.18 for the particular case of  $\omega_0 = \pi/3$ .

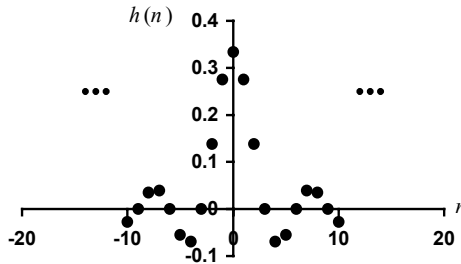


Figure 2.18. Sequence (2.42) for the particular case of  $\omega_0 = \pi/3$

It can be easily checked that this sequence cannot be absolutely summed, but has finite energy. Therefore, its Fourier transform exists, and is a generally continuous function of  $\omega$ , with a  $2\pi$  period:

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_0 \\ 0, & \omega_0 < |\omega| \leq \pi \end{cases} \quad (2.43)$$

This equation is therefore the frequency response of a system whose response to the unit sample sequence is given by (2.42). It can be noticed that (2.43) is a real function, which means that the phase of the frequency response is nil for every value of  $\omega$ . By looking at the plot of (2.43), shown in Fig. 2.19 in the case of  $\omega_0 = \pi/3$ , it can be noted that it is the same as the frequency response of an ideal low-pass filter. It is worth noting that, since its response to the unit sample sequence (2.42) is not absolutely summable, a digital system acting as an ideal low-pass filter is not stable.

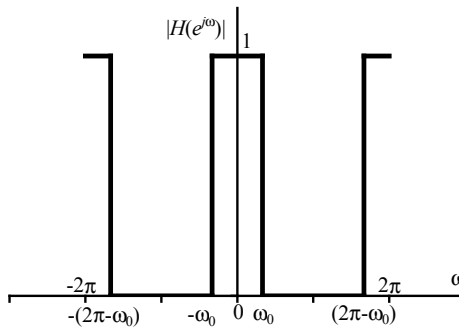


Figure 2.19. Frequency response (2.43) for  $\omega_0 = \pi/3$

## The Fourier transform of a sequence.

In the last paragraph, the Fourier transform of the sequence obtained as the response to the unit sample sequence of a linear, shift-invariant system has been derived by analysing the response of such a system to the generalized sinusoid sequence. The result of this analysis can be extended to any sequence, so that the Fourier transform of a sequence  $x(n)$  can be defined as:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n}, \quad (2.44)$$

provided that (2.44) is convergent. The rules to establish whether (2.44) converges to  $X(e^{j\omega})$  or not are, of course, the same as those seen for (2.36).

If (2.44) converges, then, similarly to (2.36), it converges to a continuous, or generally continuous function of  $\omega$ , periodic in  $\omega$  with a  $2\pi$  period. This conclusion is very important, because it shows that any sequence, provided it has a Fourier transform, has a periodic Fourier transform, with a  $2\pi$  period in  $\omega$ . This outlines, as it will be shown into details in the following chapter, a first important source of modification of the information associated to a signal, when moving from the continuous time domain to the discrete time domain after a sampling operation. Every sequence obtained by sampling a continuous-time signal will show a periodic spectrum (Fourier transform) with a  $2\pi$  period in  $\omega$ , no matter on how the spectrum of the continuous-time signal is. If we consider that the continuous-time signals that are generally dealt with in practical situations have a non-periodic spectrum, the significance of the modifications to which the information associated to a signal is subjected because of the sampling operation is quite evident.

Similarly to what was done for the Fourier transform of  $h(n)$ , it is possible to define the inverse Fourier transform of  $X(e^{j\omega})$  as:

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \quad (2.45)$$

Having defined the Fourier transform of a sequence, it is possible to determine the Fourier transform of the output  $y(n)$  of a linear, shift-invariant system, with input sequence  $x(n)$  and  $h(n)$  response to the unit sample sequence.

Applying (2.44) to (2.22) we get:

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} \left[ \sum_{k=-\infty}^{+\infty} h(n-k)x(k) \right] e^{-j\omega n} \quad (2.46)$$

If  $m = n - k$  is taken, (2.46) becomes:

$$\begin{aligned} Y(e^{j\omega}) &= \sum_{m=-\infty}^{+\infty} \left[ \sum_{k=-\infty}^{+\infty} h(m)x(k) \right] e^{-j\omega(m+k)} = \\ &= \sum_{m=-\infty}^{+\infty} h(m) e^{-j\omega m} \sum_{k=-\infty}^{+\infty} x(k) e^{-j\omega k} \end{aligned}$$

which, taking into account (2.44), leads to write:

$$Y(e^{j\omega}) = H(e^{j\omega}) \cdot X(e^{j\omega}) \quad (2.47)$$

Eq. (2.47) shows that, similarly to the continuous-time, linear, time-invariant systems, the Fourier transform of the output sequence of a discrete-time, linear, shift-invariant system is given by the product of the Fourier transform of the input sequence by the frequency response, given by the Fourier transform of the response to the unit sample sequence.

If (2.22) is considered, (2.47) shows that the convolution between two sequences in the discrete time domain is changed into the product between the Fourier transforms of the two sequences in the  $\omega$  domain. It is also possible to prove that the product between two sequences in the discrete time domain is changed into the convolution between the Fourier transforms of the two sequences in the  $\omega$  domain.

### Symmetry properties of the Fourier transform.

Similarly to the Fourier transform of the continuous-time functions, the Fourier transform of the sequences satisfies some very useful properties of symmetry that come mainly from the property of any sequence to be decomposed into the sum of a conjugate symmetric sequence and a conjugate antisymmetric sequence. In this section, the most significant properties will be shortly recalled, without entering into the mathematical details of the proofs, that are well known from the Fourier theory.

A sequence of complex samples  $x_{cs}(n)$  is called *conjugate symmetric* when the following relationship applies:

$$x_{cs}(n) = x_{cs}^*(-n)$$

where the superscript \* stands for complex conjugate. Similarly, a sequence of complex samples  $x_{ca}(n)$  is called *conjugate antisymmetric* when the following relationship applies:

$$x_{ca}(n) = -x_{ca}^*(-n)$$

It can be proved that any sequence  $x(n)$  can be written as:

$$x(n) = x_{cs}(n) + x_{ca}(n) \quad (2.48)$$

where :

$$x_{cs}(n) = \frac{1}{2} [x(n) + x^*(-n)] \quad (2.49)$$

and :

$$x_{ca}(n) = \frac{1}{2} [x(n) - x^*(-n)] \quad (2.50)$$

A *real* conjugate symmetric sequence, that is a sequence for which  $x_{cs}(n) = x_{cs}(-n)$  applies, is called an *even* sequence. Similarly, a *real* conjugate antisymmetric sequence, that is a sequence for which  $x_{ca}(n) = -x_{ca}(-n)$  applies, is called an *odd* sequence.

As for the Fourier transform, the same properties as those valid for the continuous-time functions still apply, so that it can be decomposed as:

$$X(e^{j\omega}) = X_{cs}(e^{j\omega}) + X_{ca}(e^{j\omega}) \quad (2.51)$$

where:

$$X_{cs}(e^{j\omega}) = \frac{1}{2} [X(e^{j\omega}) + X^*(e^{-j\omega})] \quad (2.52)$$

and:

$$X_{ca}(e^{j\omega}) = \frac{1}{2} [X(e^{j\omega}) - X^*(e^{-j\omega})] \quad (2.53)$$

If  $x(n)$  is a real sequence, then its Fourier transform is a conjugate symmetric function, that is:

$$X(e^{j\omega}) = X^*(e^{-j\omega}) \quad (2.54)$$

If  $x(n)$  is an even sequence, then its Fourier transform is a real function.

## SIGNAL TRANSFORMATION FROM THE CONTINUOUS TIME TO THE DISCRETE TIME DOMAIN

### *The sampling theorem and its consequences*

#### INTRODUCTION.

It has been already stated in the previous chapters that the modern digital measuring instruments are more and more based on the DSP techniques. This means that the input signals are converted into a sequence of sampled values by means of a sampling operation performed at given time instants, with a constant sampling period.

According to the analysis done in the last chapter 2, this sampling operation results in a domain transformation, from the continuous time domain where the original analog signal  $s(t)$  is defined, to the discrete time domain where the resulting sequence of sampled data  $s(n)$  is defined.

The main problem of the Digital Signal Processing theory is therefore assessing whether the information associated with the original signal  $s(t)$  is correctly transferred into the resulting sequence  $s(n)$  after the domain transformation operated by the sampling operation, or, if not, how far it is modified and which kind of errors the sampling operation introduces in the measuring process.

From a very intuitive point of view, the sampling operation does not leave the information associated with the original signal  $s(t)$  unaltered. However, still from a very intuitive point of view, if a *sufficient* number of samples is taken, the information associated with the original signal can be still retrieved from the sequence of sampled data  $s(n)$ . This is well shown by Fig. 3.1 and Fig. 3.2, which show a continuous-time signal  $s(t)$  and the sequence of sampled data  $s(n)$  respectively, where the samples have been obtained sampling  $s(t)$  with a sample rate of 40 samples/period.

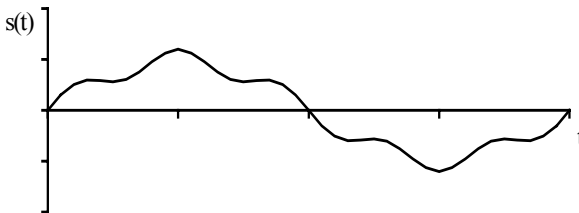


Figure 3.1. Example of continuous-time signal

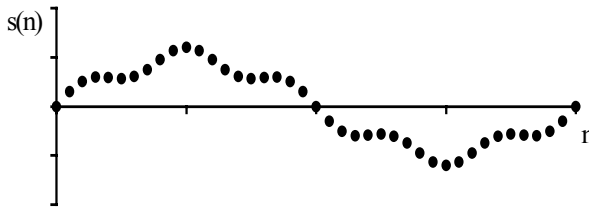


Figure 3.2. Sequence obtained by means of a correct sampling of the signal in Fig. 3.1

The original signal  $s(t)$  can be easily recognized in the sequence of sampled data represented in Fig. 3.2. If now the same signal  $s(t)$  is sampled with a much lower sample rate of only 10 samples/period, as shown in Fig. 3.3, the original signal can be no longer recognized in the sequence of sampled signals.

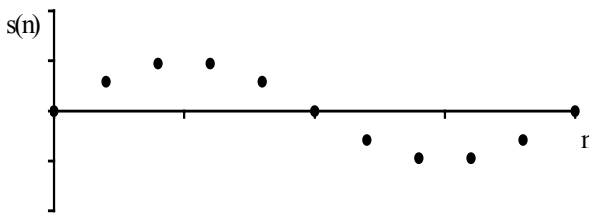


Figure 3.3. Sequence obtained by means of an incorrect sampling of the signal in Fig. 3.1

This intuitive approach is confirmed by the considerations, exposed in the previous chapter 2, concerning the Fourier transform of a sequence. Eq. (2.44) shows that the Fourier transform of a sequence is always periodic, with a  $2\pi$  period in  $\omega$ . This means that any sequence obtained by sampling a time-continuous signal has a periodic spectrum in the frequency domain, no matter on the spectrum of the sampled signal itself. Since most of the signals of practical interest show aperiodic spectra, the above consideration leads to the conclusion that the transformation from the continuous time domain to the discrete time domain operated by the sampling operation *always* modifies the information associated with the sampled signal.

The main question to which the DSP theory must give an answer is therefore if, how and under which conditions the original information can be still retrieved from the sequence of sample data.

Again from an intuitive point of view, the original information is still available in the sequence of sampled data if the modifications introduced by the sampling operation are somehow *additive* and *redundant* with respect to the original information and do not interfere with the original information in a *destructive* way. This consideration is compatible with the fact that the Fourier transform of a sequence is periodic if, as it will be shown later, the different periods of the spectrum are separated replicas of the

spectrum of the sampled signal. In this case the information associated with the sampled signal can be thought to be associated, unmodified, with a single period of the spectrum of the sequence of the sampled data.

The strict proof of this statement is given, in a mathematically sound way, by the *sampling theorem*. The following sections of this chapter will deal with this theorem in a thorough way, from the general, theoretically important approach, to the particular application to the case of periodic signals, which is very important in the practical applications.

## THE SAMPLING THEOREM.

### The general formulation.

In order to establish how the information associated with a signal in the continuous time domain is transferred into the discrete time domain after having sampled the continuous-time signal, let us consider a signal  $s_{ct}(t)$ , defined in the continuous time domain.

Let us suppose that the Fourier transform can be defined for this signal, so that its direct and inverse Fourier transform pair can be written as:

$$S_{ct}(j\Omega) = \int_{-\infty}^{+\infty} s_{ct}(t) e^{-j\Omega t} dt \quad (3.1)$$

$$s_{ct}(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{ct}(j\Omega) e^{j\Omega t} d\Omega \quad (3.2)$$

where  $\Omega$  is the angular frequency.

Let us also suppose that the Fourier transform of  $s_{ct}(t)$  is limited in  $\Omega$ , so that:

$$\begin{aligned} S_{ct}(j\Omega) &\neq 0 && \text{for } |\Omega| < \Omega_0, \\ S_{ct}(j\Omega) &= 0 && \text{for } |\Omega| \geq \Omega_0 \end{aligned} \quad (3.3)$$

This situation is graphically represented in Fig. 3.4, where an example of Fourier transform of such a signal is shown.

Let us now suppose to sample signal  $s_{ct}(t)$ , with a constant sampling period  $T_s$ , to which a sampling frequency  $f_s = 1/T_s$  and an angular sampling frequency  $\Omega_s = 2\pi/T_s$  correspond. A sequence  $s(n)$  is obtained, whose samples are equal to the values taken by signal  $s_{ct}(t)$  at the sampling periods:



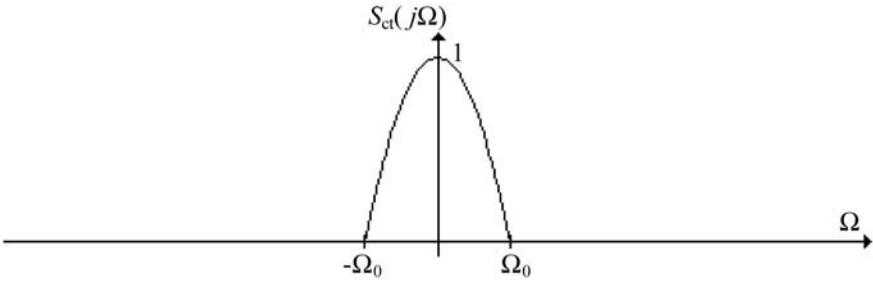


Figure 3.4. Example of Fourier transform of the considered continuous-time signal

$$s(n) = s_{ct}(nT_s) \tag{3.4}$$

Supposing that this sequence has a Fourier transform  $S(e^{j\omega})$  given by (2.44), each sample of this same sequence can be written, according to (2.45), in terms of  $S(e^{j\omega})$  as:

$$s(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} S(e^{j\omega}) e^{j\omega n} d\omega \tag{3.5}$$

Due to (3.4), the same samples can be obtained starting from (3.2) as:

$$s(n) = s_{ct}(nT_s) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{ct}(j\Omega) e^{j\Omega nT_s} d\Omega \tag{3.6}$$

Since the left-hand sides of (3.5) and (3.6) are the same, if a relationship can be obtained, starting from these two equations, between the Fourier transform  $S_{ct}(j\Omega)$  of the continuous-time signal  $s_{ct}(t)$  and the Fourier transform  $S(e^{j\omega})$  of the discrete-time sequence  $s(n)$  of samples of  $s_{ct}(t)$ , the way the information associated with  $s_{ct}(t)$  is transferred to  $s(n)$  is also found.

In order to find this relationship, let us express the integral in (3.6) as a sum of integrals over intervals with finite length  $2\pi/T_s$ . We get:

$$s(n) = \frac{1}{2\pi} \sum_{q=-\infty}^{+\infty} \int_{(2q-1)\pi/T_s}^{(2q+1)\pi/T_s} S_{ct}(j\Omega) e^{j\Omega nT_s} d\Omega \tag{3.7}$$

If now the new variable  $\Omega + 2\pi q/T_s$  is considered instead of  $\Omega$ , each integral in (3.7) can be written over the interval  $(-\pi T_s, +\pi T_s)$ , still with finite length  $2\pi/T_s$ . (3.7) becomes:

$$s(n) = \frac{1}{2\pi} \sum_{q=-\infty}^{+\infty} \int_{-\pi/T_s}^{\pi/T_s} S_{\text{ct}} \left( j\Omega + j \frac{2\pi q}{T_s} \right) e^{j\Omega n T_s} e^{j2\pi q n} d\Omega \quad (3.8)$$

Since both  $q$  and  $n$  are integer,  $e^{j2\pi q n} = 1$ . Due to the linearity of the Fourier transform, the order of integration and summation in (3.8) can be interchanged, so that (3.8) becomes:

$$s(n) = \frac{1}{2\pi} \int_{-\pi/T_s}^{+\pi/T_s} \left[ \sum_{q=-\infty}^{+\infty} S_{\text{ct}} \left( j\Omega + j \frac{2\pi q}{T_s} \right) \right] e^{j\Omega n T_s} d\Omega \quad (3.9)$$

If now  $\Omega = \omega/T_s$  is taken, (3.9) becomes:

$$s(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left[ \frac{1}{T_s} \sum_{q=-\infty}^{+\infty} S_{\text{ct}} \left( j \frac{\omega}{T_s} + j \frac{2\pi q}{T_s} \right) \right] e^{j\omega n} d\omega \quad (3.10)$$

which is formally the same as (3.5). Therefore, it is possible to write:

$$S(e^{j\omega}) = \frac{1}{T_s} \sum_{q=-\infty}^{+\infty} S_{\text{ct}} \left( \frac{j\omega}{T_s} + j \frac{2\pi q}{T_s} \right) \quad (3.11)$$

or, if the angular frequency  $\Omega$  is considered:

$$S(e^{j\Omega T_s}) = \frac{1}{T_s} \sum_{q=-\infty}^{+\infty} S_{\text{ct}} \left( j\Omega + j \frac{2\pi q}{T_s} \right) \quad (3.12)$$

Equations (3.11) and (3.12) represent the relationship between the Fourier transform of a continuous-time signal and the Fourier transform of the discrete-time signal obtained by sampling the continuous-time signal with a constant sampling period.

First of all it can be noted that the Fourier transform of the resulting discrete-time signal is periodic with a  $2\pi$  period in  $\omega$  (as expected, since it is the Fourier transform of a sequence), and periodic with a  $\Omega_s = 2\pi/T_s$  period (which is equal to the angular sampling frequency) in  $\Omega$ .

Looking at Fig. 3.5, and according to (3.11) and (3.12), it can be concluded that, if  $\Omega_s \geq 2\Omega_0$ , each period of (3.11) is an undistorted replica of the Fourier transform of  $s_{ct}(t)$ , and therefore the information associated with this signal can be still retrieved from the sequence of sampled values  $s(n)$ .

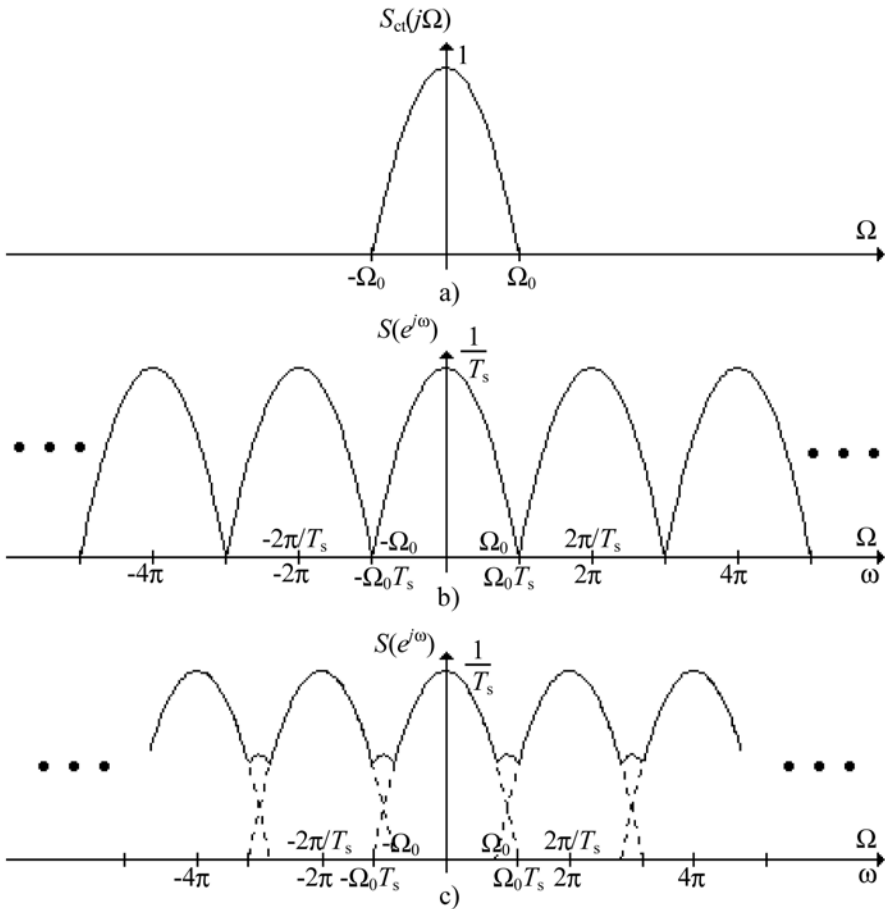


Figure 3.5. Relationship between the Fourier transforms of a continuous-time and a discrete-time signal. (a) Fourier transform of the continuous-time signal. (b) Fourier transform of the discrete-time signal obtained by a correct sampling. (c) Fourier transform of the discrete-time signal obtained by an incorrect sampling.

On the contrary, if  $\Omega_s < 2\Omega_0$ , the shifted replicas of  $S_{ct}(j\Omega)$  in (3.11) are overlapping over an interval whose length depends on how lower  $\Omega_s$  is than  $2\Omega_0$ , as shown in Fig. 3.6.

In this case, the information associated with  $s_{ct}(t)$  is modified irreparably, and cannot be retrieved from the sequence of sampled values any longer. This phenomenon is called *aliasing* and must be absolutely avoided in order to get correct results from the digital processing of the input signal.

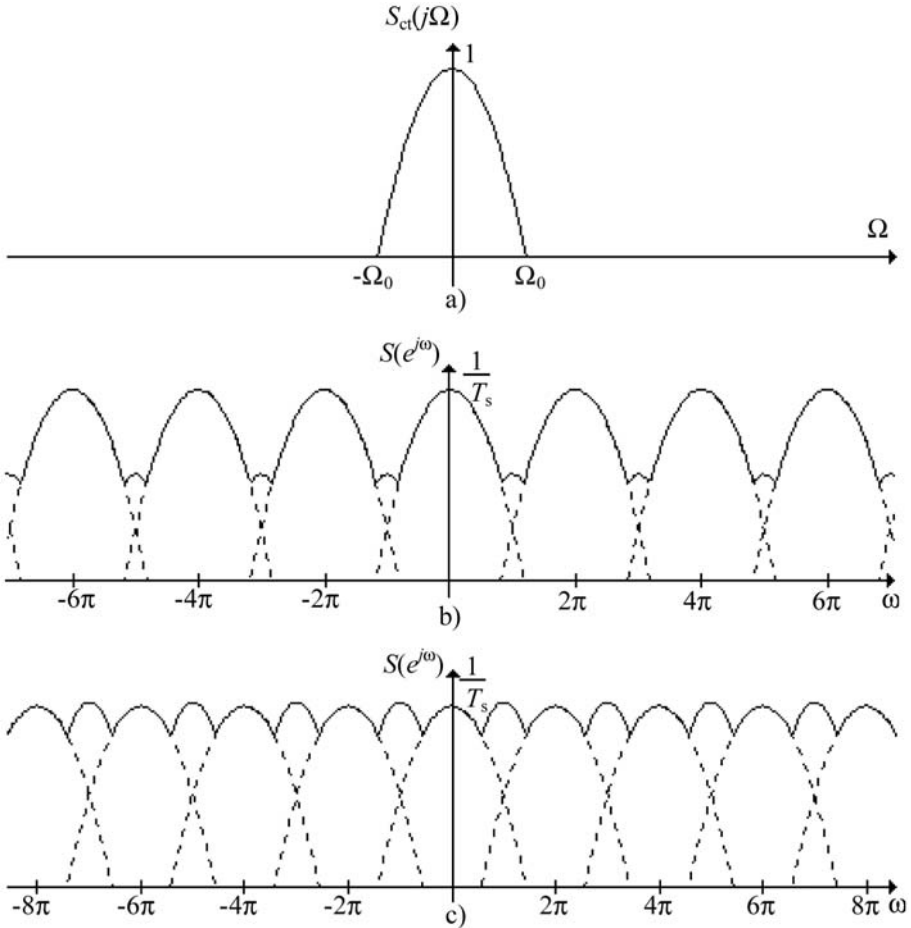


Figure 3.6. Aliasing effect on the Fourier transform of the discrete-time signal. (a) Fourier transform of the continuous time signal. (b) Fourier transform of the discrete-time signal affected by aliasing. (c) Same aliasing situation as in (b), but caused by a lower sampling frequency.

Fig. 3.6 shows that when the aliasing situation occurs, the high frequency components in the Fourier transform of the continuous-time signal are folded back over the lower

frequency components, thus modifying them. The lower is the sampling frequency with respect to the  $2\Omega_0$  limit, the more are the frequency-domain components of the signal affected by the aliasing, so that the information associated with the original signal is totally lost in the sequence of sampled data.

This phenomenon can be extremely dangerous, because the sequence of sampled data might apparently reproduce a signal with similar characteristics as those of the original sampled signal. For instance, Fig. 3.7 shows a sinewave with angular frequency  $\Omega_1 = 1$ , sampled with an angular sampling frequency  $\Omega_s = 0.9091$ . It can be readily recognized that the sampled values of this signal belong also to a sinewave with angular frequency  $\Omega_2 = 0.0909$ , and can be obtained by this one by sampling it with the same angular sampling frequency  $\Omega_s$ . Due to the aliasing, the original frequency-domain component at angular frequency  $\Omega_1$  has been therefore shifted to the lower angular frequency  $\Omega_2$ . If no information on the original signal is known, but the waveform, the sequence obtained from the incorrect sampling operation might appear as correct, thus leading to dramatic measurement errors.

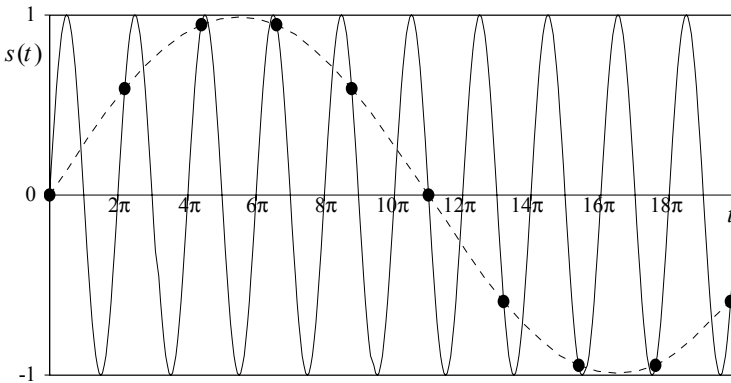


Figure 3.7. Aliasing effects: a high frequency sinewave appears to be a low frequency sinewave

The sampling theorem and the above considerations lead to conclude that a continuous-time signal is correctly sampled, that is aliasing is avoided and the information associated with this signal can be still retrieved from the sequence of sampled values, if the following conditions are met.

1. The Fourier transform (spectrum) of the continuous-time signal is upper-bounded in frequency at  $\Omega_0$ , as shown by (3.3).
2. The sampling period  $T_s$  is constant.
3. The angular sampling frequency  $\Omega_s$  is taken in such a way that  $\Omega_s \geq 2\Omega_0$ . The lowest limit for the sampling frequency  $\Omega_{sm} = 2\Omega_0$  is called the Nyquist frequency.
4. The signal is sampled over the whole time axis (that is from  $t_0 = -\infty$  to  $t_1 = +\infty$ ).

### Shannon interpolation formula.

The above paragraph proved that the information associated with a continuous-time signal in the *frequency domain* can be still retrieved from the sequence of sampled values, provided that the sampling operation meets the above listed conditions. The Fourier theory states that, if the Fourier transform exists, there is a perfect equivalence between the time and the frequency domains. This means that everything is proved as valid in one domain is also valid in the other domain. Coming to the sampling theory, the domain equivalence stated by the Fourier theory leads to conclude that, if the conditions of the sampling theorem are met, the information associated with a continuous-time signal must be retrieved also in the time-domain. This means that we must be able to find the value taken by the signal at any time  $t$ , starting by the sampled values at time  $kT_s$ .

To prove this statement, let us suppose that the continuous-time signal  $s_{ct}(t)$  has been sampled correctly, that is  $\Omega_s \geq 2\Omega_0$ . Over the  $\omega$  axis this means that  $\Omega_0 \leq \pi/T_s$ . Under this condition, (3.12) leads to:

$$S(e^{j\Omega T_s}) = \frac{1}{T_s} S_{ct}(j\Omega), \quad -\frac{\pi}{T_s} \leq \Omega \leq \frac{\pi}{T_s} \quad (3.13)$$

as graphically shown in Fig. 3.5b.

Taking into account the inverse Fourier transform (3.2) for the continuous-time signal, we get:

$$s_{ct}(t) = \frac{1}{2\pi} \int_{-\pi/T_s}^{+\pi/T_s} S(e^{j\Omega T_s}) e^{j\Omega t} d\Omega \quad (3.14)$$

Since, according to (2.44), it is:

$$S(e^{j\Omega T_s}) = \sum_{k=-\infty}^{+\infty} s_{ct}(kT_s) e^{-j\Omega T_s k},$$

(3.14) becomes:

$$s_{ct}(t) = \frac{T_s}{2\pi} \int_{-\pi/T_s}^{+\pi/T_s} \left[ \sum_{k=-\infty}^{+\infty} s_{ct}(kT_s) e^{-j\Omega T_s k} \right] e^{j\Omega t} d\Omega \quad (3.15)$$

and, by interchanging the order of integration and the summation:

$$s_{\text{ct}}(t) = \sum_{k=-\infty}^{+\infty} s_{\text{ct}}(kT_s) \left[ \frac{T_s}{2\pi} \int_{-\pi/T_s}^{+\pi/T_s} e^{j\Omega(t-kT_s)} d\Omega \right] \quad (3.16)$$

Keeping into account the solution of the integral in the square brackets, (3.16) becomes:

$$s_{\text{ct}}(t) = \sum_{k=-\infty}^{+\infty} s_{\text{ct}}(kT_s) \frac{\sin[(\pi/T_s)(t-kT_s)]}{(\pi/T_s)(t-kT_s)} \quad (3.17)$$

Equation (3.17) is due to Shannon and is generally called the *Shannon interpolation formula*. It shows that it is possible to evaluate the value taken by the continuous-time signal  $s_{\text{ct}}(t)$  at any time  $t$ , starting from the values taken at the sampling instants  $kT_s$ , provided that the sampling theorem conditions are met and no aliasing occurs. It has been hence proved that the information associated with the continuous-time signal can be retrieved from the sequence of sampled data both in the frequency and time domains.

Equation (3.17) has a great theoretical importance, but a limited practical utility, because it requires to interpolate an infinite number of samples, which means that the observation interval must be of infinite length.

This last statement is in perfect agreement with the conditions of the sampling theorem, listed at the end of the previous section, which are the direct consequence of having assumed no restrictions for the signal, except the limited bandwidth in the frequency-domain. Since the signal can be *a priori* aperiodic and its duration can be not limited in time, it is necessary to observe it over an infinite length interval to retrieve the whole amount of information associated with it.

In the practical situations, however, this kind of signals is not generally met. The phenomena that are generally dealt with are represented either by finite length signals, that are not required to be observed outside their duration, either by periodic signals, whose associated information can be retrieved by observing only one single period.

The finite length signals could not be sampled, from a strict theoretical point of view, since the Fourier transform of such signals does not show a finite bandwidth. From the practical point of view, however, the bandwidth of the Fourier transform can be forced to be limited by filtering the signal with a low-pass filter. This will introduce some decaying exponential terms in the original signal that extend its duration to infinity. However, since this components are generally rapidly decaying, they fall below the resolution of the Analog-to Digital converter employed, and the signal can be still considered with a finite duration, at least from the practical point of view.

The periodic signals need a separate analysis. It has been already shown in Chapter 2 that a sequence obtained by sampling a periodic signal might not be periodic any longer. Therefore it may happen that the sequence of the sampled values of a periodic, continuous-time signal does not allow to retrieve the information associated with the signal itself, even if the signal has been sampled with a sampling frequency greater than the Nyquist frequency, because it loses the original periodicity.

The sampling theorem must be hence reconsidered, in order to find if a condition exists that allows to sample a periodic signal correctly.

### The case of the periodic signals.

*Fourier-series approach.*

Let  $s_{ct}(t)$  be a continuous-time signal, represented by a generally continuous function of time  $t$ , periodic with period  $T$  in  $t$ , and absolutely integrable in  $t$ . Under these conditions,  $s_{ct}(t)$  can be represented by means of the Fourier series:

$$s_{ct}(t) = \frac{a_0}{2} + \sum_{k=-\infty}^{+\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=-\infty}^{+\infty} b_k \sin\left(\frac{2\pi kt}{T}\right) \quad (3.18)$$

where:

$$a_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s_{ct}(t) \cos\left(\frac{2\pi kt}{T}\right) dt \quad (3.19)$$

and:

$$b_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s_{ct}(t) \sin\left(\frac{2\pi kt}{T}\right) dt \quad (3.20)$$

It is also assumed that the following condition is satisfied:

$$\begin{aligned} a_k &= 0 & \text{for } |k| > N \\ b_k &= 0 & \text{for } |k| > N \end{aligned} \quad (3.21)$$

which means that  $s_{ct}(t)$  has a bounded spectrum.

Let us now suppose to sample  $s_{ct}(t)$  with constant sampling period  $T_s$  taken in such a way that exactly  $2N+1$  samples are taken over the signal period  $T$ , from  $-T/2$  to  $T/2$ . This means that the following relationship applies between  $T$  and  $T_s$ :

$$T = (2N + 1)T_s \quad (3.22)$$

showing that the signal period and the sampling period are *synchronous*.



Starting from the sampled values of  $s_{ct}(t)$  and taking into account (3.22), the Fourier series coefficient in (3.19) and (3.20) can be evaluated as, respectively:

$$\tilde{a}_k = \frac{2}{2N+1} \sum_{n=-N}^N s\left(\frac{nT}{2N+1}\right) \cos\left(k \frac{2\pi n}{2N+1}\right) \quad (3.23)$$

$$\tilde{b}_k = \frac{2}{2N+1} \sum_{n=-N}^N s\left(\frac{nT}{2N+1}\right) \sin\left(k \frac{2\pi n}{2N+1}\right) \quad (3.24)$$

Due to the relationship (3.22) between  $T$  and  $T_s$ , the integrals in (3.19) and (3.20) can be still obtained from (3.23) and (3.24) as:

$$a_k = \lim_{N \rightarrow \infty} \tilde{a}_k \quad (3.25)$$

$$b_k = \lim_{N \rightarrow \infty} \tilde{b}_k \quad (3.26)$$

According to Eulero, as far as (3.25) is concerned, and Gauss, as far as (3.26) is concerned, these relationships can be further on simplified, removing the limit operation. The following equations are obtained:

$$a_k = \tilde{a}_k - \sum_{j=1}^{\infty} [a_{j(2N+1)+k} + a_{j(2N+1)-k}] \quad (3.27)$$

$$b_k = \tilde{b}_k - \sum_{j=1}^{\infty} [b_{j(2N+1)+k} - b_{j(2N+1)-k}] \quad (3.28)$$

It can be immediately recognized that, if condition (3.21) is satisfied, then the summations in (3.27) and (3.28) are equal to zero, for any value of  $|k| \leq N$ . This means that:

$$\begin{aligned} \tilde{a}_k &= a_k \\ \tilde{b}_k &= b_k \end{aligned} \quad \text{for } |k| \leq N \quad (3.29)$$

It can also be recognized, by computing the summations in (3.27) and (3.28), that, for  $N < k \leq 2N$ , the Fourier series coefficients obtained from (3.25) and (3.26) provide:

$$\begin{aligned}\tilde{a}_k &= a_{(2N+1)-k} \\ \tilde{b}_k &= -b_{(2N+1)-k}\end{aligned}\quad (3.30)$$

which confirms that the Fourier components of a sequence are periodic in the frequency domain.

If condition (3.22) is removed, that is if:

$$T \neq (2N+1)T_s, \quad (3.31)$$

Eulero and Gauss relationships (3.27) and (3.28) are not applicable any longer, so that, if the signal period and the sampling period are not synchronous, it is always:

$$\begin{aligned}\tilde{a}_k &\neq a_k \\ \tilde{b}_k &\neq b_k\end{aligned}\quad \text{for any } k \quad (3.32)$$

Therefore, it can be concluded that a continuous-time, periodic signal is correctly sampled if and only if both the Nyquist condition  $\Omega_s \geq 2\Omega_0$  and the synchronous sampling condition (3.22) are satisfied. Under this assumption, the original information associated with the continuous-time signal can be retrieved from the discrete-time signal, as shown by (3.29); otherwise, the original information cannot be retrieved any longer from the discrete-time signal.

From a very intuitive point of view, this conclusion is explained, in a graphical way, by Fig. 3.8 a and b. Fig. 3.8 a) shows the sequence of samples obtained by sampling a sinuswave with 20 samples/period, in a synchronous way ( $T = 20T_s$ ). The two periods of the sampled signal shown in Fig. 3.8 a) have been drawn by duplicating the samples obtained from one single period of the original signal, shifting them by  $20T_s$ , and attaching them to the right of the first period plot. It can be immediately recognized that the original sinuswave appears unaltered.

Fig. 3.8 b) has been drawn in the same way as Fig. 3.8 a), again sampling a sinuswave with 20 samples/period, but not in a synchronous way ( $T \neq 20T_s$ ). It can be immediately recognized that a distortion appears in the sinuswave, at zero crossing, as evidenced on Fig. 3.8 b) by an ellipse.

The above mathematical derivation and the intuitive example reported in Fig. 3.8 prove that the synchronous sampling condition must be satisfied in order to preserve the original information associated with the continuous-time signal after the sampling operation.

Unfortunately, the synchronous condition (3.22) can be barely satisfied under practical conditions. In fact, it requires the knowledge of the signal period, which is generally unknown, and an infinite resolution of the sampling clock in order to generate the

sampling pulses that satisfy (3.22) for every possible signal period  $T$ .

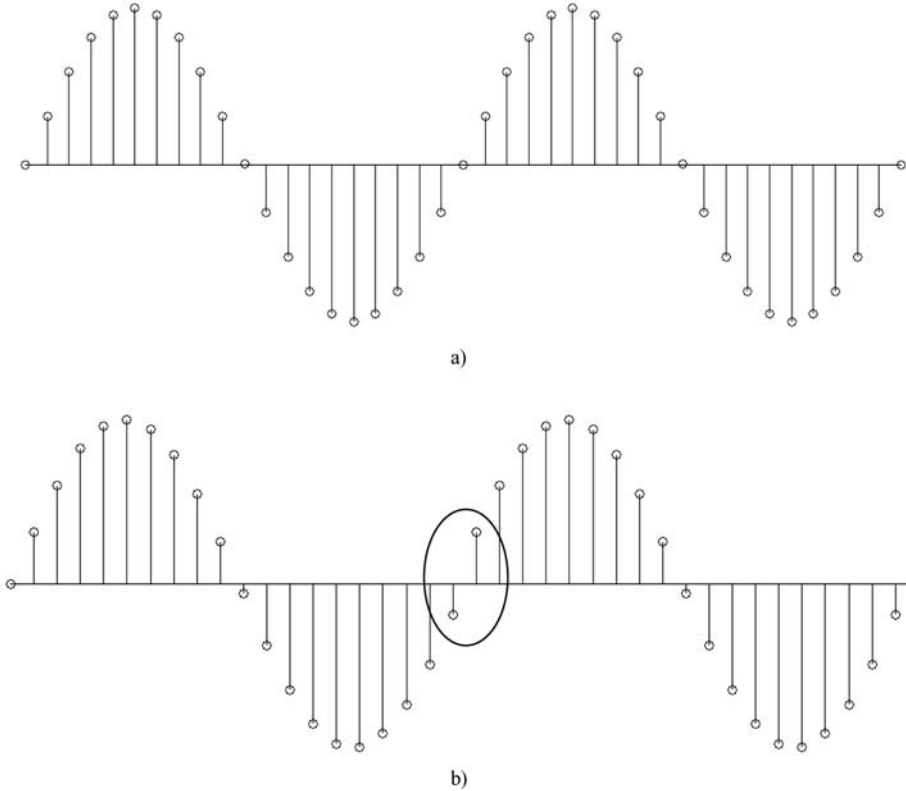


Figure 3.8. Example of correct, synchronous sampling (a) and incorrect, asynchronous sampling (b) of a periodic signal

Due to these reasons, the synchronous sampling conditions cannot generally be achieved, unless the signal period is a priori known and remains stable for the whole observation interval. It is therefore important to understand what kind of errors are generated by the lack of synchronism, quantify them and investigate the possible methods for their reduction.

The followed Fourier-series approach to the sampling theorem for periodic signals leads in quite straightforward way to the synchronous sampling condition, but does not allow to explain the errors that arise when this condition is not satisfied and, consequently, does not allow to identify the possible methods that can be adopted to reduce them.

### Convolution approach

The approaches to the sampling operation described in the previous sections are all based on a mathematical representation in terms of functions, in the continuous time domain, and sequences, in the discrete time domain.

A generalization of this approach is possible if distributions are considered instead of functions. If the sampling distribution  $g'(t)$ , shown in Fig. 3.9, is defined as a proper combination of Dirac impulses, equally spaced by the sampling period  $T_s$ :

$$g'(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT_s), \quad (3.33)$$

sampling a continuous-time signal  $s_{ct}$  expressed as a function of time  $s_{ct}(t)$  is mathematically represented by the product of  $s_{ct}(t)$  by the sampling distribution (3.33):

$$p'(t) = s_{ct}(t) \cdot g'(t) \quad (3.34)$$

where  $p'(t)$  is a distribution of Dirac impulses, equally spaced by the sampling period  $T_s$ , whose amplitude is weighed by the value taken by function  $s_{ct}(t)$  at each sampling time.

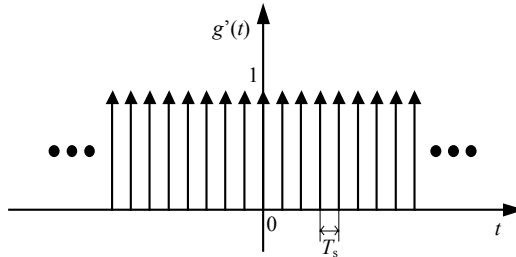


Figure 3.9. Representation of the sampling distribution for an infinite duration observation interval

In the frequency domain, the Fourier transform  $G'(j\Omega)$  of  $g'(t)$  is again a combination of Dirac impulses, scaled by the quantity  $1/T_s$  and equally spaced in angular frequency by a quantity  $2\pi/T_s$ .  $G'(j\Omega)$  is therefore expressed by:

$$G'(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} \delta\left(j\Omega + j\frac{2\pi k}{T_s}\right) \quad (3.35)$$

and graphically represented in Fig. 3.10.

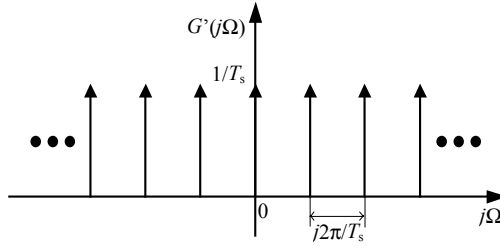


Figure 3.10. Frequency domain representation of the sampling distribution for infinite duration observation interval

Taking into account the properties of the Fourier transform, the Fourier transform of the sampled distribution  $p'(t)$  in (3.34) can be obtained as the convolution of the Fourier transforms of  $s(t)$  and  $g'(t)$ :

$$P'(j\Omega) = S_{ct}(j\Omega) * G'(j\Omega) \tag{3.36}$$

It can be immediately recognized that, due to the periodicity of the Dirac impulses that constitute  $G'(j\Omega)$ ,  $P'(j\Omega)$  is the periodic replica of  $S_{ct}(j\Omega)$  with period  $\Omega_s = 2\pi/T_s$ . If  $S_{ct}(j\Omega)$  is upperbounded at angular frequency  $\Omega_0 = \Omega_s/2$ , each period of  $P'(j\Omega)$  is a scaled replica of  $S_{ct}(j\Omega)$ , otherwise aliasing occurs, as already shown in the previous sections.

Let us now consider a periodic signal  $s_{ct}(t)$ , with period  $T$ . If the distribution theory is considered, it is possible to define the Fourier transform of this signal, as a combination of Dirac impulses, equally spaced, in angular frequency, by a quantity  $j2\pi/T$ , and weighed by the Fourier series complex coefficients appertaining to each harmonic component. Let the Fourier transform of  $s_{ct}(t)$  be upperbounded, in frequency, by the  $N$ -th harmonic component.

Let us now suppose to sample only one period of  $s_{ct}(t)$ , with sampling period  $T_s$ , so that  $2N+1$  samples are taken over the period, and the sampling theorem is satisfied. The observation interval is therefore  $T_0 = (2N+1)T_s$ . Under these conditions, the sampling distribution (3.33) becomes:

$$g(t) = \sum_{k=-N}^{+N} \delta(t - kT_s) \tag{3.37}$$

and the sampled signal can again be mathematically expressed by the product:

$$p(t) = s_{ct}(t) \cdot g(t) \tag{3.38}$$

In the frequency domain, the Fourier transform of  $g(t)$  is no longer given by a sequence of Dirac impulses, because of the finite duration of  $g(t)$  itself. The Fourier

transform of  $g(t)$  is given by:

$$G(j\Omega) = \frac{\sin\left(\frac{\Omega}{2}T_s(2N+1)\right)}{\sin\left(\frac{\Omega}{2}T_s\right)} \quad (3.39)$$

and is graphically represented in Fig. 3.11 for the case of  $N = 16$  and  $T_s = 0.628$  ms.

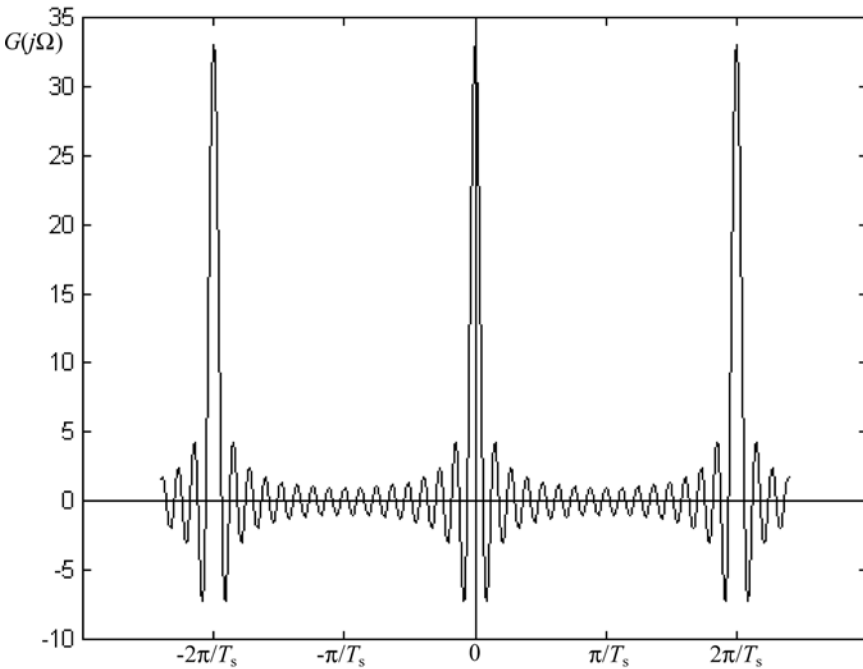


Figure 3.11. Fourier transform of the sampling distribution when the observation interval has finite duration. The case of  $N=16$  and  $T_s=0.628$  ms is represented.

It can be immediately recognized that  $G(j\Omega)$  is a continuous function and is again periodic with period  $2\pi/T_s$  in  $\Omega$ . Since (3.37) is an even distribution, (3.39) is a real function.

The Fourier transform  $P(j\Omega)$  of the sampled distribution (3.38) is given again by the convolution theorem:

$$P(j\Omega) = S_{ct}(j\Omega) * G(j\Omega) \quad (3.40)$$

In order to analyse the relationship between  $P(j\Omega)$  and  $S_{ct}(j\Omega)$ ,  $G(j\Omega)$  should be analysed. It can be noted that this function has its zero crossings at:

$$\Omega_0(k) = \frac{2\pi k}{(2N+1)T_s}; \quad \begin{cases} k \text{ integer} \\ k \neq 2N+1 \end{cases} \quad (3.41)$$

and has the peak value of the main lobes at:

$$\Omega_p(k) = \frac{2\pi k}{T_s}; \quad k \text{ integer} \quad (3.42)$$

It can be noted that both the zero crossings and the peak values are located at integer multiples of the inverse of the observation interval  $T_0 = (2N+1)T_s$ . If the synchronous sampling condition (3.22) is satisfied, the observation interval is the same as the signal period:  $T_0 = T = (2N+1)T_s$ . This means that the harmonic components of  $s_{ct}(t)$  are located either under the peak value of the main lobe or at the zero crossings of (3.93), as shown in Fig. 3.12.

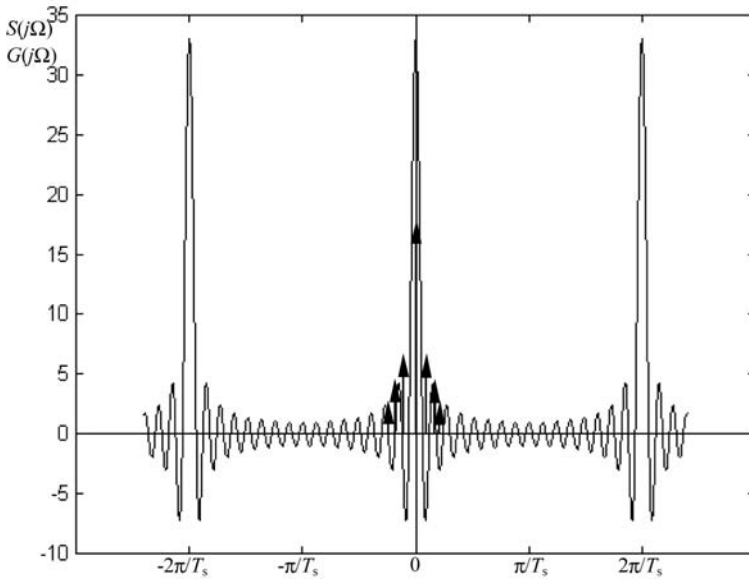


Figure 3.12. Relative position of the Fourier transforms of  $S_{ct}(j\Omega)$  and  $G(j\Omega)$  when the synchronous sampling condition is satisfied

When the convolution (3.40) is considered, at each step of the convolution, only one harmonic component is located under the peak of the main lobe of  $G(j\Omega)$ , while all other components are located at the zero crossings of  $G(j\Omega)$ . These components are therefore weighed by zero, and do not contribute to the computation of each convolution value. Therefore, the values taken by  $P(j\Omega)$  at each harmonic frequency of  $S_{ct}(j\Omega)$  are the same as those taken by  $S_{ct}(j\Omega)$ , except for a constant scale factor.

### The interpolation formula

As already stated when the general formulation of the sampling theorem has been derived, if the conditions of this theorem are met, the information associated with the continuous-time signal must be retrieved, from the discrete-time signal, also in the time domain. This statement led to the interpolation formula (3.17), due to Shannon, that has been derived under the assumption that the signal was observed for an infinite duration interval.

When the sampled signal  $s_{ct}(t)$  is periodic, with period  $T$ , the above proved sampling theorem states that, if its spectrum is upper-bounded at harmonic  $N$ , the signal itself can be observed for a finite observation interval equal to period  $T$ , provided that  $2N + 1$  samples are taken over this period and the synchronous condition (3.22) is met. This means that all other values taken by  $s_{ct}(t)$  outside the sampling instants must be retrieved by the  $2N + 1$  samples.

This can be proved by expressing the continuous-time signal  $s_{ct}(t)$  as a function of its harmonic components:

$$s_{ct}(t) = \frac{T_s}{2\pi} \sum_{k=-N}^N S_{ct} \left( e^{j\frac{2\pi k}{2N+1}} \right) e^{j\frac{2\pi kt}{T}}, \quad (3.43)$$

$T_s$  being the sampling period and  $\omega = \frac{2\pi}{2N+1}$  the fundamental angular frequency.

Since all conditions of the sampling theorem are met, the spectral components of the sampled signal can be obtained starting from the sequence of sampled values  $s(n) = s_{ct}(nT_s)$  as:

$$S_{ct} \left( e^{j\frac{2\pi k}{2N+1}} \right) = \sum_{n=-N}^N s_{ct}(nT_s) e^{-j\frac{2\pi kn}{2N+1}} \quad (3.44)$$

If (3.44) is considered in (3.43), the following relationship can be written:

$$s_{ct}(t) = \frac{T_s}{2\pi} \sum_{k=-N}^N e^{j\frac{2\pi kt}{T}} \sum_{n=-N}^N s_{ct}(nT_s) e^{-j\frac{2\pi kn}{2N+1}} \quad (3.45)$$



By separating the term that depends on  $n$  only by those that depend on  $n$  and  $k$ , it is possible to write:

$$s_{ct}(t) = \frac{T_s}{2\pi} \sum_{n=-N}^n s_{ct}(nT_s) \sum_{k=-N}^N e^{j\frac{2\pi kt}{T}} e^{-j\frac{2\pi kn}{2N+1}} = \frac{T_s}{2\pi} \sum_{n=-N}^n s_{ct}(nT_s) \sum_{k=-N}^N e^{-jk \left[ 2\pi \left( \frac{n}{2N+1} \frac{t}{T} \right) \right]} \quad (3.46)$$

Taking into account that:

$$\sum_{k=-N}^N e^{-j\omega k} = \frac{\sin \left[ \frac{\omega}{2} (2N+1) \right]}{\sin \left( \frac{\omega}{2} \right)}$$

and that  $T = (2N+1)T_s$ , equation (3.46) becomes:

$$s_{ct}(t) = \frac{T_s}{2\pi} \sum_{n=-N}^N s_{ct}(nT_s) \frac{\sin \left[ \frac{\pi}{T_s} (t - nT_s) \right]}{\sin \left[ \frac{\pi}{(2N+1)T_s} (t - nT_s) \right]} \quad (3.47)$$

Equation (3.47) represents the interpolation formula for periodic signals. It shows not only the same theoretical importance as Shannon interpolation formula (3.17), but it can also be of practical use, since it is a summation of a finite number of terms and can therefore be computed.

### The leakage errors

The convolution approach to the sampling theorem in the case of periodic signals shows that the Fourier transform (3.40) of the distribution (3.38) of the sampled data is given by the convolution of the Fourier transform  $S_{ct}(j\Omega)$  of the continuous time signal and the Fourier transform  $G(j\Omega)$ , given by 3.39, of the sampling distribution  $g(t)$ , given by 3.37.

When the synchronous sampling condition (3.22) is satisfied, that is when the signal period  $T$  and the sampling period  $T_s$  are related by:  $T = (2N+1)T_s$ , the harmonic angular frequencies  $\Omega_k = 2\pi k/T$  of the continuous time signal are the same as the zero-crossing frequencies (3.41) of  $G(j\Omega)$ , except for the case of  $k = (2N+1)$ , when the harmonic angular frequency of the continuous time signal is the same as the frequency of the peak of the main lobe of  $G(j\Omega)$ . This situation is represented in Fig. 3.13, which is an enlargement of Fig. 3.12 where the signal harmonic components are located.

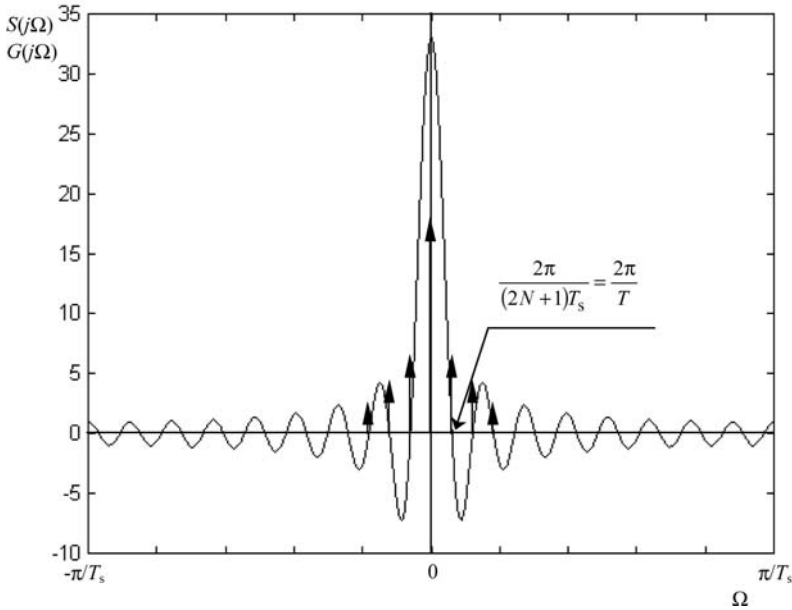


Figure 3.13. Position of the harmonic components of  $s_{ct}(t)$  when the synchronous sampling condition is satisfied

As it has been already shown, only one harmonic component of the continuous-time signal, the one that lies under the main lobe of  $G(j\Omega)$ , is weighed by a non-zero value at each step of convolution (3.40), so that the values of the harmonic components of the sequence of the sampled, discrete-time signal are the same as the harmonic components of the continuous-time signal.

On the contrary, when the synchronous sampling condition (3.22) is not satisfied, that is when  $T \neq (2N+1)T_s$ , the zero-crossing angular frequencies (3.41) for  $G(j\Omega)$  are not the same as the harmonic angular frequencies of the continuous-time signal any longer. Similarly, the angular frequency for the peak of the main lobe of  $G(j\Omega)$  does no longer correspond to an harmonic angular frequency of the continuous-time signal. This situation is depicted in Fig. 3.14, where the same signal as that in Fig. 3.13 is supposed to be sampled with a higher sampling frequency than the one adopted in Fig. 3.13, but without meeting the synchronous sampling condition (3.22).

Due to the different relative position of  $G(j\Omega)$  and  $S_{ct}(j\Omega)$ , at each step of convolution (3.40), the harmonic component  $P\left(j\frac{2\pi k}{(2N+1)T_s}\right)$  of the sampled signal, evaluated starting from the acquired samples, is related to the harmonic component  $S_{ct}\left(j\frac{2\pi k}{T}\right)$  of

the continuous time signal by the following relationship:

$$P\left(j\frac{2\pi k}{(2N+1)T_s}\right) = \alpha S_{ct}\left(j\frac{2\pi k}{T}\right) + \sum_{h \neq k} \beta_h S_{ct}\left(j\frac{2\pi h}{T}\right) \quad (3.48)$$

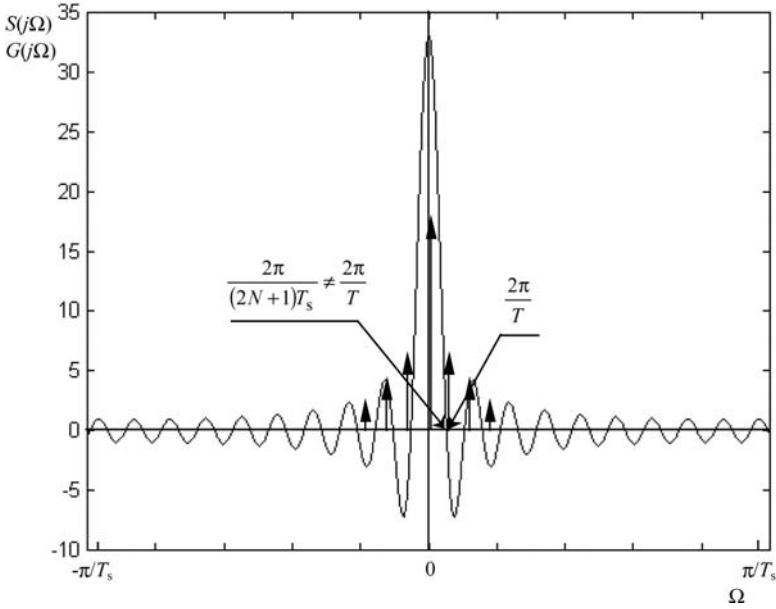


Figure 3.14. Position of the harmonic components of  $s_{ct}(t)$  when the synchronous sampling condition is not satisfied

The first term in the right side of (3.48) is due to the fact that the harmonic component of  $S_{ct}(j\Omega)$  that lies under the main lobe of  $G(j\Omega)$  is not centred exactly under the peak of the main lobe itself. If the amplitude of the main lobe of  $G(j\Omega)$  is normalized to 1, then the harmonic component of  $S_{ct}(j\Omega)$  that lies under the main lobe of  $G(j\Omega)$  is weighed by a coefficient  $\alpha$ , with  $0 \leq \alpha \leq 1$ . The value of  $\alpha$  depends on how displaced is the signal harmonic component with respect to the centre of the main lobe of  $G(j\Omega)$ . This displacement is given, in terms of angular frequency, by:

$$\Delta\Omega = 2\pi \left[ \frac{1}{(2N+1)T_s} - \frac{1}{T} \right] \quad (3.49)$$

and gives the distance of the actual sampling condition from the synchronous one. The value of  $\alpha$  depends also, of course, on the shape of the main lobe of  $G(j\Omega)$ .

The second term in the right side of (3.48) is due to the fact that the harmonic components of  $S_{ct}(j\Omega)$  that fall outside the main lobe of  $G(j\Omega)$  are not located at the zero-crossings of  $G(j\Omega)$ . Therefore, they are weighed by a coefficient  $\beta$ , with  $\beta \geq 0$ , and give a generally non-zero contribution to convolution (3.40). The values taken by each  $\beta_h$  coefficient depend on how displaced is the signal harmonic component with respect to the nearest zero-crossing of the side lobes of  $G(j\Omega)$ . Therefore, once again, they depend on the value of  $\Delta\Omega$  given by (3.49) and on the shape of the side lobes of  $G(j\Omega)$ .

Of course, under synchronous sampling conditions,  $\Delta\Omega = 0$ ,  $\alpha = 1$ ,  $\beta_h = 0$ ,  $\forall h$ , and therefore  $P\left(j\frac{2\pi k}{(2N+1)T_s}\right) = S_{ct}\left(j\frac{2\pi k}{T}\right)$ .

The errors coming from the non-synchronous sampling conditions are generally called *leakage errors*. This name comes from the fact that the spectral energy of each impulse of (3.35), that when the signal is observed for an infinite length interval is confined in an infinitesimal frequency range, leaks out in a finite frequency range, equal to the width of the lobes of (3.39), when the signal is observed for a finite length interval.

In particular, the error that arises when  $\alpha < 1$  is called *short range* leakage error, because it is due to the spectral leakage of the main lobe of (3.39). The error that arises when  $\beta_h \neq 0$  is called *long range* leakage error because it is due to the spectral leakage of the side lobes of (3.39). Since this error can also be seen as the interference of the harmonic components other than the one under evaluation with this last one, it is also called *spectral interference* error.

## METHODS FOR THE REDUCTION OF THE LEAKAGE ERRORS

### INTRODUCTION.

The discussion of the sampling theorem reported in the previous Chapter 3 showed that a periodic signal with period  $T$  and only  $N$  non-zero harmonic components is sampled correctly if at least  $2N + 1$  samples are taken over period  $T$  in such a way that the signal period and the minimum sampling period  $T_s$  are related by (3.22), that is:  $T = (2N+1)T_s$ .

Chapter 3 showed as well that leakage errors arise if this condition is not satisfied and showed that these errors are related to the difference between the signal period  $T$  and the observation interval  $T_o = (2N + 1)T_s$ , and the shape of the Fourier transform of the sampling distribution.

Of course, the only way to avoid the leakage errors is, theoretically speaking, the synchronization between the sampling clock and the signal period, so that (3.22) is satisfied. Unfortunately, this synchronization can hardly be attained in practice, due to the following reasons:

- the signal period is generally unknown;
- provided that the signal period has been measured, the sampling clock has generally not enough resolution to satisfy (3.22);
- provided that the signal period has been measured and the sampling clock has enough resolution to satisfy (3.22), the signal period is not generally stable enough to ensure that (3.22) is satisfied along the whole measurement process.

Due to the above reasons, a periodic signal is almost never sampled in the strict mathematical correct way prescribed by the sampling theorem. Suitable methods must therefore be investigated in order to reduce the leakage errors that arise because of the lack of synchronization between the signal period and the sampling period.

The most widely used methods rely on the use of windows, different from the rectangular one (3.37), for weighing the sampled signal. This Chapter will show how the use of suitable windows results in a reduction of the leakage errors and will discuss also other methods, such as the interpolation methods in the time and frequency domains, for reducing these errors further on.

At last, an hardware structure for the automatic synchronization of the sampling clock to the signal period will be shortly described, and the conditions under which the synchronization can be obtained in an effective way will be also discussed.

## THE WEIGHING WINDOWS.

### The rectangular window.

When a signal has to be observed for a limited time, such as the period for the periodic signals, the main purpose of a weighing window is to truncate the signal outside the required observation interval. The most immediate weighing window is the rectangular one, that weighs by 1 the samples which fall inside the observation interval, and by 0 those which fall outside this interval.

This window has already been discussed in the previous Chapter 3, where it has been defined by (3.37). In Chapter 3 it has been considered symmetrical with respect to the time origin for the sake of simplicity; in fact, in this case, it could be represented by the even distribution (3.37), so that its Fourier transform (3.39) is real and can be handled in a quite simple way.

In the practice, a signal is observed starting from the initial time  $t_i = 0$ , and therefore the rectangular window is mathematically described by:

$$w_r(t) = \sum_{k=0}^{2N} \delta(t - kT_s) \quad (4.1)$$

Since this window is used to weigh the acquired samples, it is generally applied to a sequence in the discrete-time domain, rather than to the continuous-time signal. It is therefore more convenient to re-write (4.1) as a sequence, in the discrete time domain:

$$w_r(n) = \sum_{k=0}^{2N} \delta(n - k), \quad (4.2)$$

where  $\delta(n)$  is the unit sample sequence defined by (2.2).

The rectangular window can be written also as:

$$w_r(n) = \begin{cases} 1, & 0 \leq n \leq 2N \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

The Fourier transform of (4.3) can be immediately obtained as:

$$W_r(e^{j\omega}) = \sum_{k=0}^{2N} e^{-j\omega k} = \frac{1 - e^{-j\omega(2N+1)}}{1 - e^{-j\omega}} = \frac{\sin\left[\frac{\omega}{2} \cdot (2N+1)\right]}{\sin\left(\frac{\omega}{2}\right)} \cdot e^{-j\omega N} \quad (4.4)$$

It is immediate to check that (4.4) is the same, in the discrete-time domain, as (3.39), except for the exponential term that takes into account the  $N$ -samples right-shift of (4.2) with respect to (3.37).

The magnitude of the Fourier transform of the rectangular window is given by:

$$|W_r(e^{j\omega})| = \left| \frac{\sin\left[\frac{\omega}{2} \cdot (2N + 1)\right]}{\sin\left(\frac{\omega}{2}\right)} \right| \tag{4.5}$$

and is graphically represented in Fig. 4.1 for  $N = 16$ .

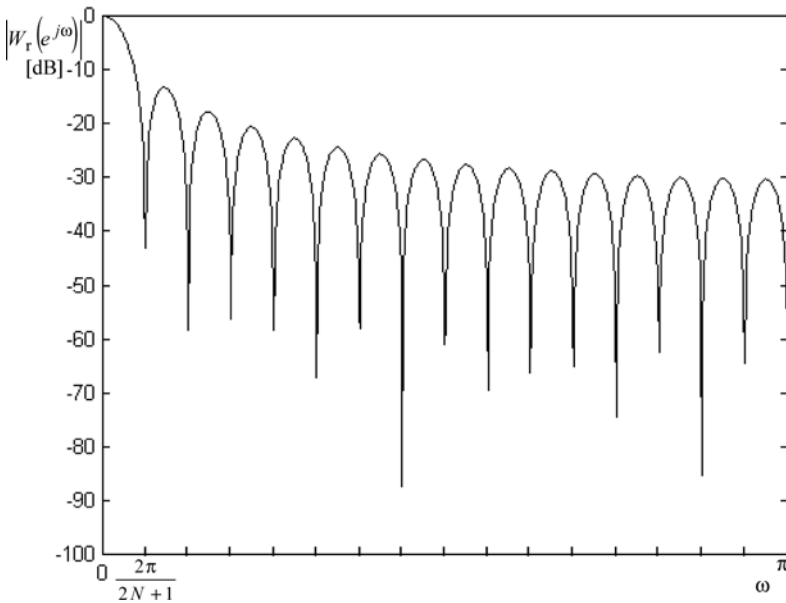


Figure 4.1. Magnitude of the Fourier transform of the rectangular window for  $N=16$

By analysing (4.5), the following interesting characteristics can be found for the magnitude of the Fourier transform of the rectangular window:

- the amplitude of the second lobe is only 13 dB below the amplitude of the main lobe;
- Fig. 4.1 shows also that the asymptotical decaying of the side lobes is very poor; this last characteristic of the Fourier transform of the rectangular window is directly related to the discontinuity of the window itself at each end in the time domain. Since the window and all its derivatives are not continue at the end of the window, the

Fourier transform shows the maximum amplitude of the side lobes and the minimum asymptotical decaying;

- at last, the width of the main lobe of the rectangular window is equal to  $2\pi/(N+1)$  and is equal to the width of all side lobes.

All above characteristics make the rectangular window very unsuitable to reduce the effects of the lack of synchronism between the signal period and the sampling period. In fact, for a given value of  $\Delta\Omega$  in (3.49), the rectangular window provides the maximum values for coefficients  $\beta_h$  and the maximum deviation from 1 for coefficient  $\alpha$  in (3.48). Therefore the maximum short and long-range leakage errors are obtained with this window.

## The cosine windows

According to the above considerations, a reduction in the leakage errors can be obtained if coefficients  $\beta_h$  in (3.48) are kept as close to zero as possible, and coefficient  $\alpha$  in (3.48) is kept as close to 1 as possible.

Since the value of the  $\beta_h$  coefficients is related to the amplitude of the side lobes of the weighing window, the reduction of these coefficients is obtained if the amplitude of the side lobes is reduced. The presence of side lobes in the Fourier transform of a weighing window, also known as the Gibbs phenomenon, is a well known mathematical issue and can be moderated through a less abrupt truncation of the window at its edges. This requires the use of a window that ensures continuity at its edges. Once this kind of windows is defined, the value of coefficient  $\alpha$  can be set close to 1 in a suitable range of values for  $\Delta\Omega$  by adapting the shape of the main lobe.

The most widely used weighing windows are the *cosine windows*, whose mathematical properties allow to tailor their shape in a quite easy way in order to optimise the values of coefficients  $\beta_h$  or  $\alpha$ .

The cosine windows are defined as:

$$w(n) = \sum_{\ell=0}^{L-1} A_\ell \cdot \cos\left(\frac{2\pi\ell n}{2N+1}\right), \quad (4.6)$$

where  $L$  is the *order* of the window,  $A_\ell$  are the coefficients of the window, and  $2N+1$  is the length of the window. This last value is the same as the observation interval, expressed in terms of number of samples. An example of cosine windows is shown in Fig 4.2, for  $L = 2$ ,  $A_0 = 0.5$  and  $A_1 = -0.5$ . This window is called the *Hanning* window, and the way it decreases to zero with continuity is quite evident

It can be immediately recognized that, for  $L = 1$  and  $A_0 = 1$ , (4.6) degenerates in the rectangular window, that can therefore be considered as a particular case of the cosine windows.

The Fourier transform of a cosine window can be obtained by applying the definition (2.44) of the Fourier transform of a sequence:



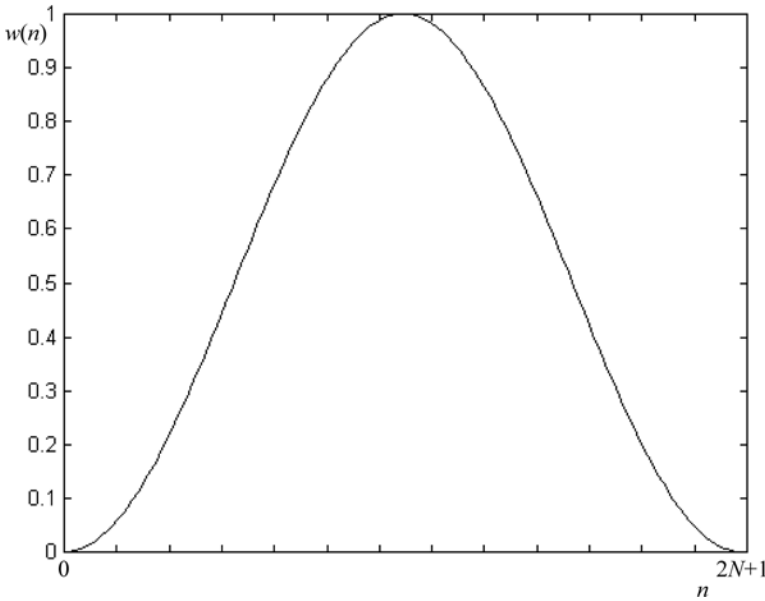


Figure 4.2. Example of cosine window for  $L = 2$ ,  $A_0 = 0.5$ ,  $A_1 = -0.5$  and  $N = 64$

$$W(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} w(n)e^{-j\omega n} = \sum_{n=0}^{2N} \sum_{\ell=0}^{L-1} A_{\ell} \cos\left(\frac{2\pi\ell n}{2N+1}\right) e^{-j\omega n} \quad (4.7)$$

The order of the two summations can be changed, so that (4.7) becomes:

$$W(e^{j\omega}) = \sum_{\ell=0}^{L-1} A_{\ell} \sum_{n=0}^{2N} \cos\left(\frac{2\pi\ell n}{2N+1}\right) e^{-j\omega n} \quad (4.8)$$

By expressing the cosine function in its exponential form, equation (4.8) becomes:

$$\begin{aligned} W(e^{j\omega}) &= \sum_{\ell=0}^{L-1} \frac{A_{\ell}}{2} \sum_{n=0}^{2N} \left( e^{j\frac{2\pi\ell n}{2N+1}} \cdot e^{-j\omega n} + e^{-j\frac{2\pi\ell n}{2N+1}} \cdot e^{-j\omega n} \right) = \\ &= \sum_{\ell=0}^{L-1} \frac{A_{\ell}}{2} \sum_{n=0}^{2N} \left( e^{-j\left(\omega - \frac{2\pi\ell}{2N+1}\right)n} + e^{-j\left(\omega + \frac{2\pi\ell}{2N+1}\right)n} \right) \end{aligned} \quad (4.9)$$

Finally, this equation converges to:

$$\begin{aligned}
 W(e^{j\omega}) = & \sum_{\ell=0}^{L-1} \frac{A_\ell}{2} \frac{\sin\left[\frac{\omega - \frac{2\pi\ell}{2N+1}}{2} \cdot (2N+1)\right]}{\sin\left(\frac{\omega - \frac{2\pi\ell}{2N+1}}{2}\right)} \cdot e^{-jN\left(\omega - \frac{2\pi\ell}{2N+1}\right)} + \\
 & + \sum_{\ell=0}^{L-1} \frac{A_\ell}{2} \frac{\sin\left[\frac{\omega + \frac{2\pi\ell}{2N+1}}{2} \cdot (2N+1)\right]}{\sin\left(\frac{\omega + \frac{2\pi\ell}{2N+1}}{2}\right)} \cdot e^{-jN\left(\omega + \frac{2\pi\ell}{2N+1}\right)}
 \end{aligned} \tag{4.10}$$

which is a linear combination of shifted Fourier transforms of the rectangular window. It is worth to note that, for  $L = 1$  and  $A_0 = 1$ , (4.10) is actually the Fourier transform (4.4) of the rectangular window (4.2).

The peak value of the main lobe of (4.10) can be obtained by evaluating (4.10) for  $\omega = 0$ :

$$\begin{aligned}
 W(e^{j0}) = & \sum_{\ell=0}^{L-1} \frac{A_\ell}{2} \left\{ \frac{-\sin\left(\frac{2\pi\ell}{2}\right)}{-\sin\left[\frac{2\pi\ell}{2 \cdot (2N+1)}\right]} \cdot e^{j\frac{2\pi\ell N}{2N+1}} + \frac{\sin\left(\frac{2\pi\ell}{2}\right)}{\sin\left[\frac{2\pi\ell}{2 \cdot (2N+1)}\right]} \cdot e^{-j\frac{2\pi\ell N}{2N+1}} \right\} = \\
 = & \sum_{\ell=0}^{L-1} \frac{A_\ell}{2} \frac{\sin\left(\frac{2\pi\ell}{2}\right)}{\sin\left[\frac{2\pi\ell}{2 \cdot (2N+1)}\right]} \cdot \left( e^{j\frac{2\pi\ell N}{2N+1}} + e^{-j\frac{2\pi\ell N}{2N+1}} \right) = \\
 = & \sum_{\ell=0}^{L-1} A_\ell \cos\left(\frac{2\pi\ell N}{2N+1}\right) \frac{\sin\left(\frac{2\pi\ell}{2}\right)}{\sin\left[\frac{2\pi\ell}{2 \cdot (2N+1)}\right]}
 \end{aligned} \tag{4.11}$$

It can be immediately recognized that all terms with  $\ell \neq 0$  in (4.11) are equal to zero, so that (4.11) becomes:

$$W(e^{j0}) = A_0 \cdot (2N + 1) \quad (4.12)$$

This equation has a very important meaning, because it states that, under synchronous sampling conditions, the Fourier series components of a periodic signal evaluated after having weighed it with a cosine window are the same as those evaluated after having weighed the signal with the rectangular window, except for the scale factor given by the window coefficient  $A_0$ .

In order to move from the general theoretical definition (4.6) of a cosine window to more practical definitions, parameters  $L$  and  $A_\ell$  must be defined. Different criteria can be followed in their definition, according to the different desired characteristics. The first desired window characteristic is common to all windows and requires to normalize the window amplitude to one.

Since the cosine windows show an even symmetry, they attain their peak value for  $n = (2N+1)/2$ . The normalization condition can therefore be written as:

$$w\left(\frac{2N+1}{2}\right) = \sum_{\ell=0}^{L-1} A_\ell \cos(\pi\ell) = 1$$

that leads to the following condition:

$$\sum_{\ell=0}^{L-1} (-1)^\ell A_\ell = 1 \quad (4.13)$$

Since the window must be convex, the  $A_\ell$  coefficients must have alternate signs, with  $A_0 > 0$ . If this last condition is taken into account, (4.13) becomes:

$$\sum_{\ell=0}^{L-1} |A_\ell| = 1, \quad (4.14)$$

and represents the normalization condition, that is the first condition set for coefficients  $A_\ell$  in order to define suitable weighing windows.

### *Cosine windows with maximum asymptotic decay*

It has already been stated that the analysis of the Gibbs phenomenon, which is responsible for the presence of the side lobes in the rectangular window, shows that the

amplitude of the side lobes is related to the discontinuity that the windows has at its edges.

The same mathematical analysis proves that, for a given order  $L$ , the windows that are continuous at their edges and have continuous derivatives at their edges are those with the maximum asymptotic decay of the side lobe amplitude.

This mathematical property suggests the conditions to set in order to find the  $A_\ell$  coefficients for this kind of windows. Since the cosine terms in the cosine windows (4.6) become equal to 1 at the windows edges, that is for  $n = 0$  and  $n = 2N + 1$ , the condition that must be satisfied to ensure window continuity at the edges, that is:  $w(0) = w(2N+1) = 0$ , is:

$$\sum_{\ell=0}^{L-1} A_\ell = 0 \quad (4.15)$$

If  $L = 2$  is taken, conditions (4.14) and (4.15) allow to determine the two  $A_0$  and  $A_1$  coefficients. In particular, we get:  $A_0 = 0.5$  and  $A_1 = -0.5$ , which define the Hanning window shown in Fig. 4.2. Fig. 4.3 shows the magnitude of the Fourier transform (4.10) in the case of the Hanning window, evaluated for  $N = 16$ . The magnitude of the Fourier transform of the rectangular window is also reported (dotted line) in the same figure, for direct comparison.

The advantage, in terms of reduction of the amplitude of the side lobes, and therefore in terms of reduction of the  $\beta_n$  coefficients, is quite evident. The amplitude of the second lobe of the Hanning window is 31 dB below the main lobe peak, while that of the second lobe of the rectangular window is only 13 dB below the main lobe peak.

Fig. 4.3 shows also another characteristic of this cosine window. The width of the main lobe is twice that of the main lobe of the rectangular window. This is mathematically proved by analysing (4.10); the width  $\Delta W_m$  of the main lobe of a cosine window is given by:

$$\Delta W_m = \frac{2\pi}{2N+1} \cdot L \quad (4.16)$$

Equation (4.16) shows that the width of the main lobe of a cosine window is directly proportional to the window order, and is  $L$ -times wider than that of the rectangular window. This may cause serious interference problems if the sampled signal has two or more contiguous harmonic components that fall under the main lobe of the employed cosine window and requires to adopt the countermeasures that will be described later in this Chapter.

Condition (4.15), that ensures the window continuity at its edges, is not the only condition that can be set to improve the asymptotic decay of the side lobes. The analysis of the Gibbs phenomenon shows that it can be mitigated if also the continuity of the window derivatives is ensured at the window edges.

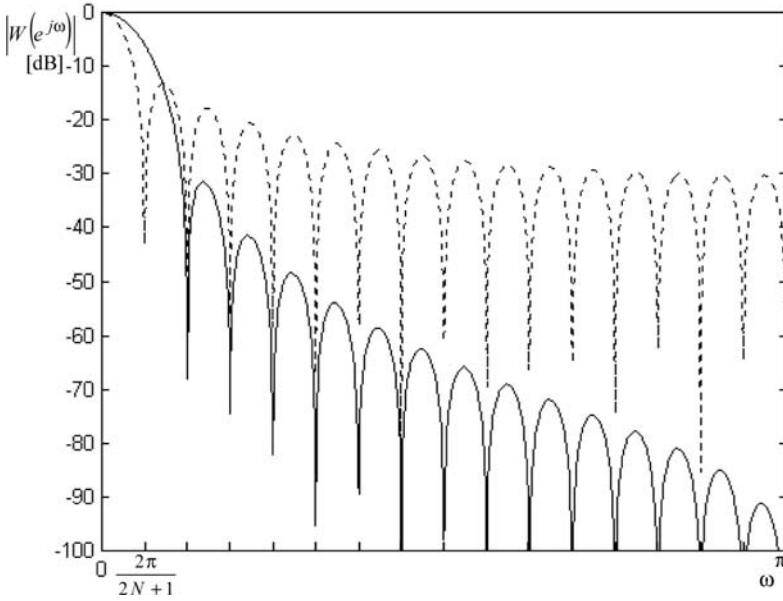


Figure 4.3. Magnitude of the Fourier transform of the Hanning (solid line) and rectangular (dotted line) windows in the case of  $N = 16$

One of the interesting mathematical properties of the cosine windows is that, since all their odd derivatives are expressed as a sum of sine terms, their continuity at the window edges is naturally ensured, without any need for additional conditions. The second derivative of a cosine window (4.6) is given by:

$$\frac{d^2 w(n)}{dn^2} = -\left(\frac{2\pi}{2N+1}\right)^2 \sum_{\ell=0}^{L-1} \ell^2 A_\ell \cos\left(\frac{2\pi \ell n}{2N+1}\right)$$

and its continuity at the window edges is ensured if the following condition is satisfied:

$$\sum_{\ell=0}^{L-1} \ell^2 A_\ell = 0 \tag{4.17}$$

Taking into account this condition, together with conditions (4.14) and (4.15), leads to the definition of the following coefficients of a third order window:  $A_0 = 0.375$ ,  $A_1 = -0.5$  and  $A_2 = 0.125$ . These coefficients define a third-order window, and the magnitude of its Fourier transform is reported in Fig. 4.4, always together with the magnitude of the Fourier transform of the rectangular window for direct comparison.

The effect, on the amplitude of the side lobes, of having ensured the continuity of the second derivative is quite evident. The amplitude of the second lobe of this window is 47 dB below the main lobe peak, that is about 20 dB below the second lobe of the Hanning window and about 30 dB below the second lobe of the rectangular window.

According to (4.16), the width of the main lobe is three times wider than that of the rectangular window, as clearly shown by Fig. 4.4.

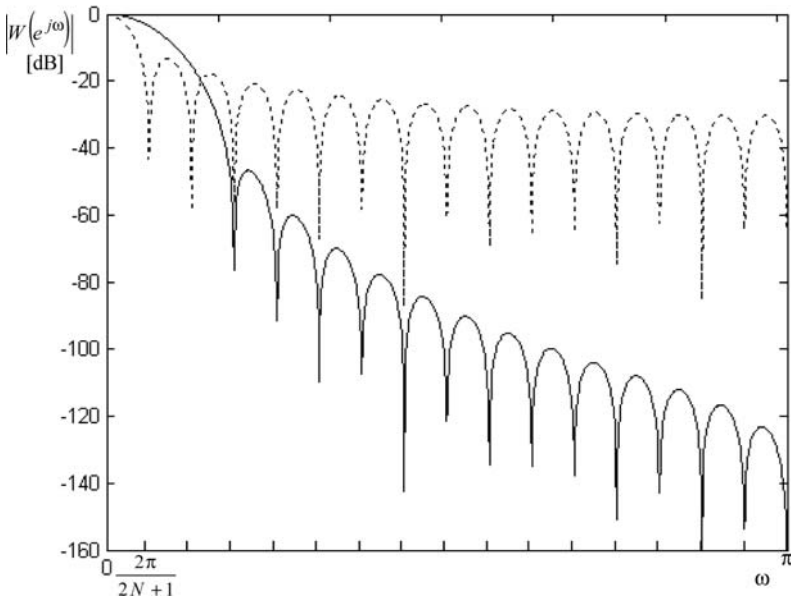


Figure 4.4. Magnitude of the Fourier transform of the third-order (solid line) and rectangular (dotted line) windows in the case of  $N=16$

The asymptotic decay of the side lobes can be further on increased if the continuity of the fourth derivative is imposed at the edges of the windows. The fourth derivative of (4.6) is given by:

$$\frac{d^4 w(n)}{dn^4} = \left(\frac{2\pi}{2N+1}\right)^4 \sum_{\ell=0}^{L-1} \ell^4 A_\ell \cos\left(\frac{2\pi\ell n}{2N+1}\right)$$

and its continuity at the window edges is ensured if the following condition is satisfied:

$$\sum_{\ell=0}^{L-1} \ell^4 A_{\ell} = 0 \quad (4.18)$$

This new condition, together with (4.14), (4.15) and (4.17) leads to the definition of the coefficients of a fourth order window, the so called  $\sin^6(\ell)$  window:  $A_0 = 10/32$ ,  $A_1 = -15/32$ ,  $A_2 = 6/32$  and  $A_3 = -1/32$ . The magnitude of the Fourier transform of this window is reported in Fig. 4.5, again with the magnitude of the Fourier transform of the rectangular window. It can be noted that the amplitude of the second lobe is 61 dB below that of the main lobe, thus ensuring a quite effective reduction of the long-range leakage errors. On the other hand, the main lobe, according to (4.16), is four times wider than the main lobe of the Fourier transform of the rectangular window.

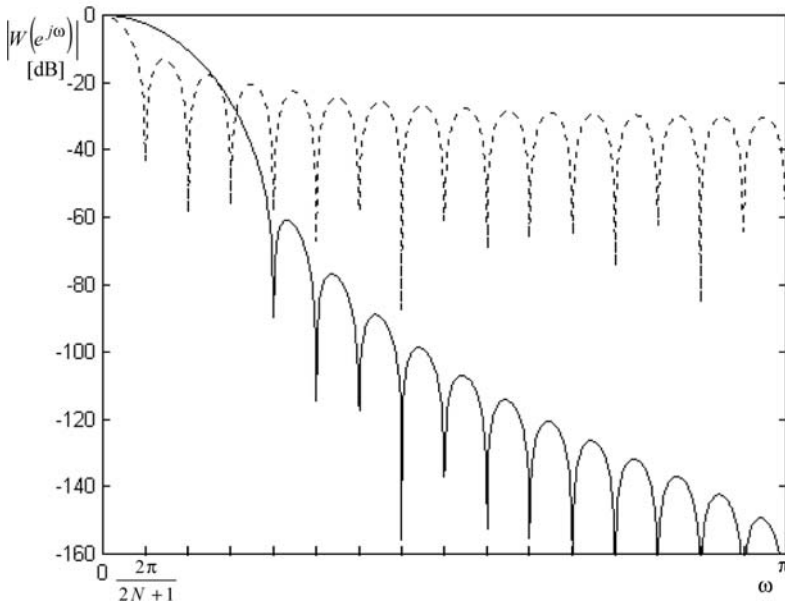


Figure 4.5. Magnitude of the Fourier transform of the  $\sin^6(\ell)$  (solid line) and rectangular (dotted line) windows in the case of  $N = 16$ .

### Cosine windows with minimum amplitude for the second lobe

The cosine windows with maximum asymptotic decay of the side lobe amplitude are optimized for the reduction of the long-range leakage errors. However this is not the only way a cosine window can be optimized. In several applications, the analyzed signals may show two spectral components at contiguous harmonic frequencies. In this case, a

window showing the minimum amplitude of the second lobe is expected to minimize the harmonic interference of the two contiguous spectral components on each other.

By reminding that the width of the main lobe of a cosine window is given by (4.16) and that the width of the side lobes is the same as the width of the lobes of the rectangular window, the minimization of the peak value of second lobe of a cosine window of given order  $L$  requires to determine the values of the  $A_\ell$  coefficients that minimize the magnitude of the Fourier transform of the cosine window for:

$$\omega_{p2} = \Delta W_m + \frac{1}{2} \frac{2\pi}{2N+1} = \frac{2\pi L}{2N+1} + \frac{\pi}{2N+1} = \frac{\pi}{2N+1} \cdot (2L+1) \quad (4.19)$$

Of course, the obtained values for the  $A_\ell$  coefficients depend on the window order  $L$ . If the magnitude of (4.10) is evaluated for  $L = 2$  and  $\omega = \omega_{p2}$ , it can be proven that the obtained value is minimized by  $A_0 = 0.54$  and  $A_1 = -0.46$ . These coefficients define the second order window called *Hamming* window. Fig. 4.6 shows the magnitude of the Fourier transform (4.10) in the case of the Hamming window, evaluated for  $N = 16$ . The magnitude of the Fourier transform of the rectangular window is reported again, in the same Fig. 4.6, for direct comparison.

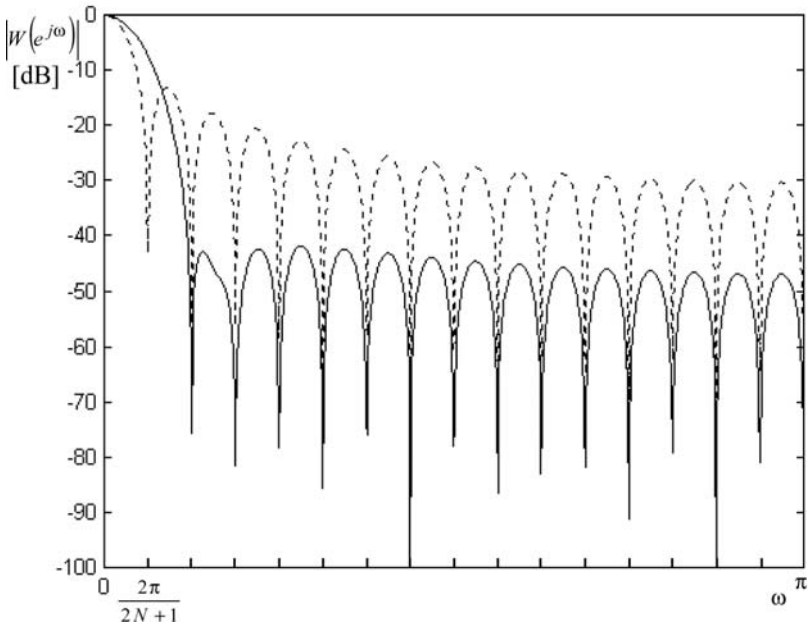


Figure 4.6. Magnitude of the Fourier transform of the Hamming (solid line) and rectangular (dotted line) windows in the case of  $N = 16$ .



The magnitude of the second lobe of the Hamming window is 43 dB below that of the main lobe. If this window is compared with the Hanning window (which is also a second order window), it can be noted that the amplitude of the second lobe of the Hamming window (-43 dB) is much lower than the amplitude of the second lobe of the Hanning window (-31 dB). On the contrary, the side lobes of the Hanning window (Fig. 4.3) are more rapidly decaying than those of the Hamming window, whose amplitudes remain almost constant. This can be easily explained by considering that the coefficients of the Hamming window do not satisfy condition (4.15) and therefore a fast decay of the side lobes cannot be expected.

The same optimization as that performed on the second lobe of the second order windows can be done also on the third order windows. In this case, the window with the minimum amplitude of the second lobe is the *Blackman-Harris* window, whose coefficients are:  $A_0 = 0.412323$ ,  $A_1 = -0.49755$ ,  $A_2 = 0.07922$ . The magnitude of the Fourier transform (4.10) of the Blackman-Harris window, evaluated for  $N = 16$ , is reported in Fig. 4.7, together with the magnitude of the Fourier transform of the rectangular window for direct comparison.

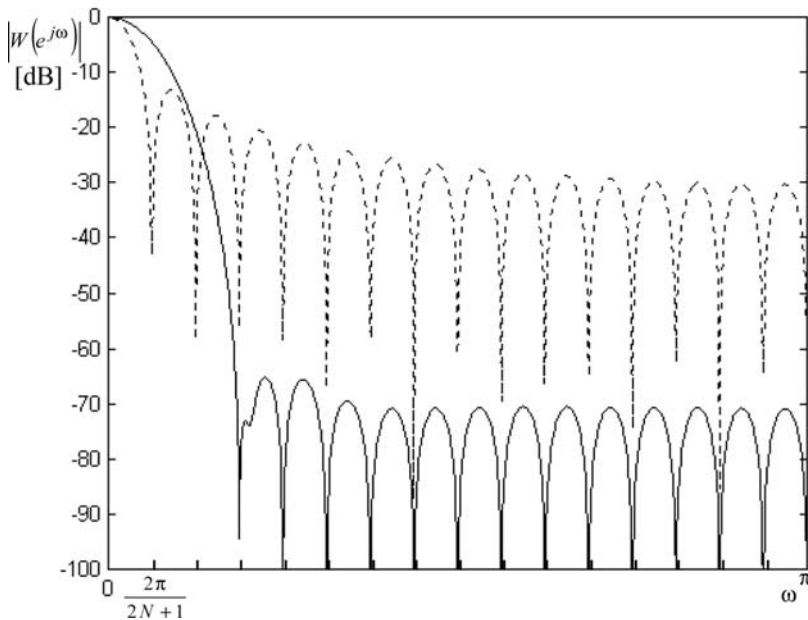


Figure 4.7. Magnitude of the Fourier transform of the Blackman-Harris (solid line) and rectangular (dotted line) windows in the case of  $N = 16$ .

The magnitude of the second lobe of the Blackman-Harris window is 65 dB below that of the main lobe. If this window is compared with the third-order window optimized for the maximum asymptotic decay of the side lobe amplitudes, it can be

noted that the amplitude of its second lobe (-65 dB) is lower than the amplitude of the second lobe of that window (-47 dB). Similarly to the Hamming window, however, the side lobes of the Blackman-Harris window do not decay significantly, in agreement with the fact that the coefficients of this window do not satisfy condition (4.15).

### *Flat-top cosine windows*

The cosine windows analyzed until now have been optimized in order to reduce the effects of the harmonic interference and, therefore, the long-range leakage errors. The performed optimization was in fact concerned with the amplitude of the side lobes of the window Fourier transform.

It has been shown that, on the contrary, the short-range leakage errors are related to the shape of the main lobe of the Fourier transform of the employed window. In order to mitigate the short-range leakage error, the window main lobe should be maximally flat in the origin. This condition requires to annihilate, for  $\omega = 0$ , as many derivatives as possible for the magnitude of the window Fourier transform (4.10), according to the window order and to the fact that the normalization condition (4.14) must always be satisfied.

This kind of optimization leads to the definition of the so-called *flat-top* windows. If a third order window is considered, this optimization yields the following coefficients:  $A_0 = 0.2811$ ,  $A_1 = -0.5209$ ,  $A_2 = 0.1980$ . The magnitude of the Fourier transform of this window is plotted in Fig. 4.8, in the case of  $N = 16$ .

The same figure shows also the magnitude of the Fourier transform of the third-order with maximum asymptotic decay (dashed line) and rectangular (dotted line) windows, so that the features of the obtained flat-top window can be immediately compared with those of these other two windows. The flatness of the main lobe is quite evident and shows that, for a given frequency displacement (3.49), the signal harmonic component that lies under the window main lobe is affected by a lower short-range leakage error than that caused by any other window of the same order  $L$ .

On the other hand, the flat-top window shows a second lobe with a peak value of -45 dB, higher than the peak value (-47 dB) of the second lobe of the third-order window, thus confirming that it has not been optimized for the mitigation of the long-range leakage error.

If a good performance is required for reducing both the short and long-range leakage errors, the window order must be increased. As an example, the fifth order flat-top window is considered, with the following coefficients:  $A_0 = 0.21557895$ ,  $A_1 = -0.41663158$ ,  $A_2 = 0.277263158$ ,  $A_3 = -0.083578947$ ,  $A_4 = 0.006947368$ . The magnitude of the Fourier transform (4.10) of this window, in the case of  $N = 16$ , is reported in Fig. 4.9.

The efficiency of this window in mitigating the short-range leakage error is well evidenced by the flatness of its main lobe. The efficiency in the harmonic interference reduction is also good, since the second lobe peak is 84 dB below the main lobe peak. The main drawback in the use of this window is the width of its main lobe, equal to 5 times the width of the main lobe of the rectangular window.

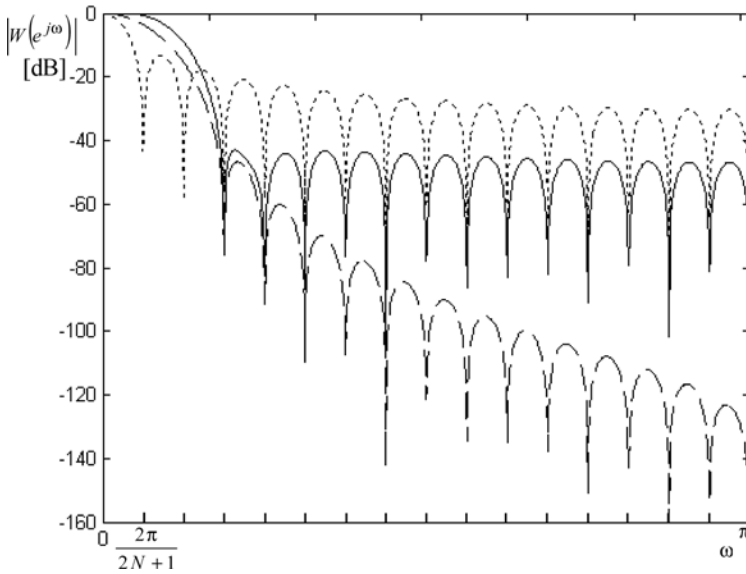


Figure 4.8. Magnitude of the Fourier transform of the third order flat-top (solid line), maximum asymptotic decay (dashed line) and rectangular (dotted line) windows in the case of  $N = 16$ .

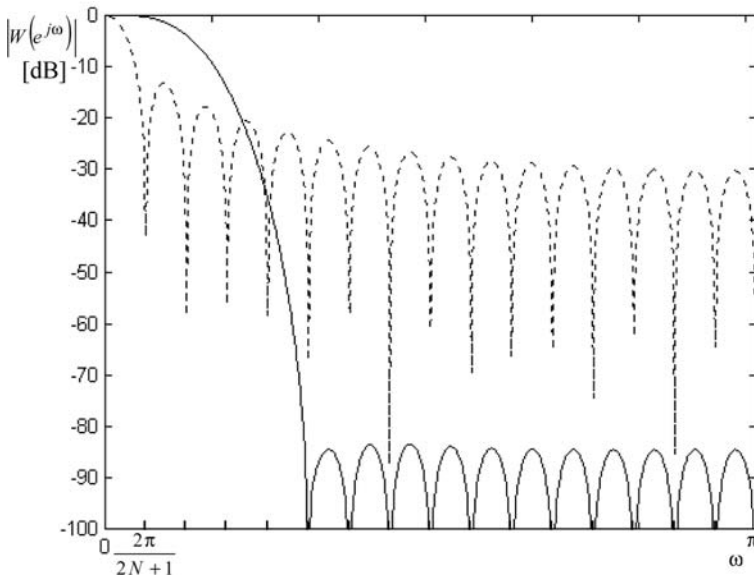


Figure 4.9. Magnitude of the Fourier transform of the fifth order flat-top (solid line) and rectangular (dotted line) windows in the case of  $N = 16$ .

## The selection of the observation interval

In the previous section, the width  $\Delta\omega_m$  of the main lobe of a cosine window was shown to be given by (4.16). This equation shows that the width of the main lobe of a cosine window is  $L$ -times wider than that of the rectangular window, where  $L$  is the window order.

This means that the first zero of the window Fourier transform is located in:

$$\omega_0 = \frac{2\pi}{2N+1} \cdot L \quad (4.20)$$

If this window is employed to sample a signal with a sampling period  $T_s$ , the first zero of the Fourier transform of the continuous-time window is:

$$\Omega_0 = \frac{2\pi}{(2N+1)T_s} \cdot L, \quad (4.21)$$

remembering that  $\omega = \Omega T_s$ . This means that even under synchronous sampling conditions,  $L$  spectral components of the sampled signal are located under the main lobe of the cosine window, as shown by Fig. 4.10, where the Hanning window is considered as an example.

This situation is absolutely intolerable, since a very large spectral interference appears between the two adjacent spectral components of the signal that fall under the main lobe of the window Fourier transform.

In order to avoid this situation, the width of the main lobe of the Fourier transform of the selected window must be reduced by the same enlargement factor  $L$  due to the window. According to (4.20) this can be done by processing a number of samples equal to:  $M = (2N+1) \cdot L$ . In principle this increment in the number of processed samples can be obtained either by oversampling the signal with an oversampling ratio  $L$ , or by observing the signal for an observation interval:

$$T'_0 = L \cdot (2N+1) \cdot T_s \quad (4.22)$$

If (4.21) is considered, it can immediately be recognized that oversampling the signal by  $L$  does not solve the problem, because the total number of considered samples is always given by:  $M = L \cdot (2N+1) \cdot T_s / L = (2N+1)$ . On the contrary, increasing the observation interval by  $L$ , according to (4.22), solves the problem, as shown in Fig. 4.11 where the observation interval was doubled, since the Hanning window was employed.

As a general rule, when a  $L$ -order cosine window is used, the observation interval must be increased by  $L$  times with respect to the minimum duration observation interval required by the sampling theorem.

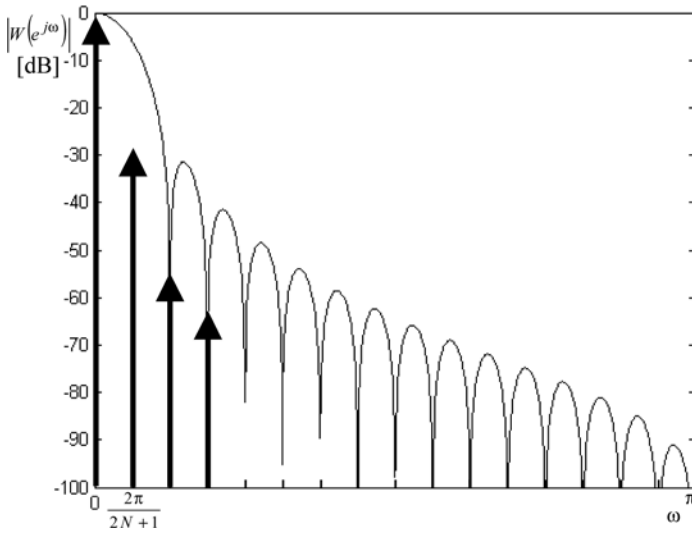


Figure 4.10. Relative position of the Hanning window Fourier transform and the spectral components of the sampled signal

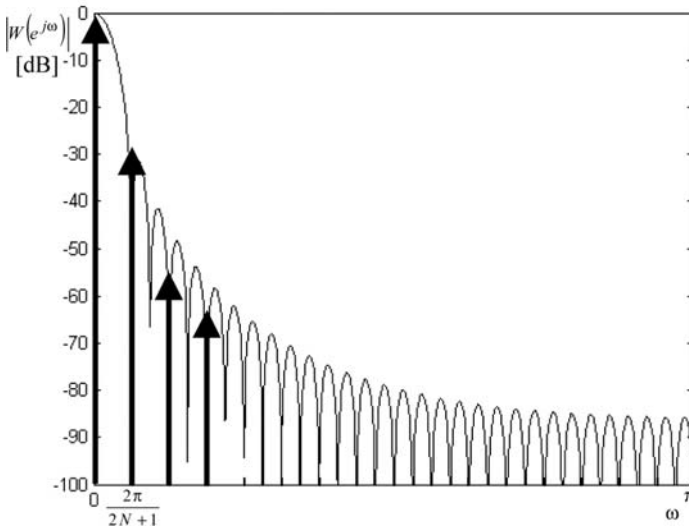


Figure 4.11. Relative position of the Hanning window Fourier transform and the spectral components of the sampled signal after having observed the signal itself for two periods

From the practical point of view, however, this condition should be handled with care. In fact, whenever an observation interval longer than the minimum-length one required by the theory is employed and the observed signal is supposed to be periodic, errors may arise if the signal itself does not remain stable for the whole observation interval. Therefore, before using the windows defined in the previous paragraphs, the stability of the sampled signal over the observation interval must be checked.

At last, it is worth while observing that the leakage errors caused by the lack of synchronism between the signal period and the sampling period can be mitigated by using suitable windows, but this is paid in terms of an increment in the number of samples that must be processed and, consequently, in the observation interval: both these quantities become larger than the minimum ones allowed by the strict observation of all assumptions of the sampling theorem.

## THE INTERPOLATION METHODS

The use of smoothing windows, such as the cosine windows shown in the previous paragraphs, is not the only possible method for reducing the leakage errors, though it is probably the most widely employed one.

Another effective method is based on interpolation techniques that can be implemented both in the time domain, in order to evaluate the signal period more accurately, and in the frequency domain, in order to evaluate the signal spectral components more accurately.

### Time-domain interpolation method

Let us consider a threshold  $a$ , taken in such a way that the sampled signal  $s_{ct}(t)$  crosses this threshold, with a given slope, only once in a period. Let us also suppose that the signal can be considered linear in a given interval around the threshold crossing. If the sampling period  $T_s$  is shorter than this interval, the signal can be supposed to vary linearly within the sampling period. Therefore, if two samples are taken in such a way that:

$$s_{ct}((k-1)T_s) = s(k-1) \leq a \quad \text{and} \quad s_{ct}(kT_s) = s(k) > a$$

the actual time at which the signal crosses the threshold can be obtained by linear interpolation.

If the example in Fig. 4.12 is considered, we have that:

$$\Delta s_{ct}(t) = s_{ct}(kT_s) - s_{ct}((k-1)T_s) \quad (4.23)$$

$$\Delta' s_{ct}(t) = a - s_{ct}((k-1)T_s) \quad (4.24)$$

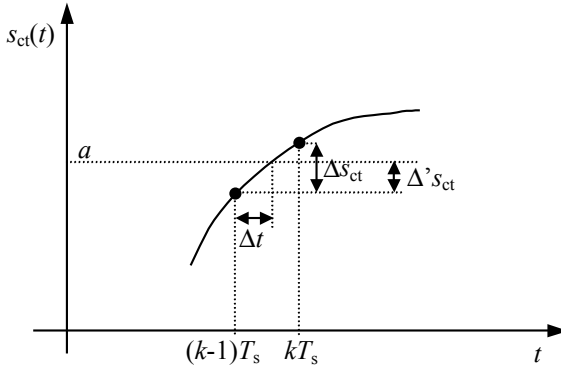


Figure 4.12. Example of time-domain linear interpolation

and, due to the similarity of the two triangles in Fig. 4.12, we get:

$$\Delta t = \frac{\Delta' s_{ct}(t)}{\Delta s_{ct}(t)} T_s \quad (4.24)$$

By substituting (4.22) and (4.23) into (4.24) we get:

$$\Delta t = \frac{a - s_{ct}((k-1)T_s)}{s_{ct}(kT_s) - s_{ct}((k-1)T_s)} T_s \quad (4.25)$$

so that the actual crossing time of threshold  $a$  is given by:

$$t_c = (k-1)T_s + \Delta t \quad (4.26)$$

If the sequence of samples in the discrete time is considered, (4.25) becomes:

$$\Delta = \frac{a - s(k-1)}{s(k) - s(k-1)} \quad (4.27)$$

and the actual crossing of threshold  $a$  occurs at the fictitious sample  $k - 1 + \Delta$ .

If also the successive crossing time  $t_{2c} = (m-1)T_s + \Delta t_2$  is evaluated, with  $m > k$ , the period of the continuous time signal can be estimated as:

$$T = t_{2c} - t_c = (m-k)T_s + \Delta t_2 - \Delta t \quad (4.28)$$

It can be noted that the correction factor  $\Delta t_2 - \Delta t$  is applied to interval  $(m-k)T_s$  obtained

by processing the acquired samples, without applying the interpolation method. It can also be noted that, under synchronous sampling conditions, the threshold crossing occurs always at the same distance from the last sample below the threshold itself, so that  $\Delta t_2 = \Delta t$ , and no correction is needed, as expected from the theory.

This method is quite accurate if the signal can be well approximated by a linear function within the sampling period. Generally, this assumption is valid if a sampling period shorter than the one given by the sampling theorem is employed, that is if the signal is oversampled. This means that, in general, for a given bandwidth of the sampled signal, this method requires the use of a more expansive structure than the one prescribed by the sampling theorem, since a higher sampling frequency is required.

Moreover, any measurement algorithm applied to the acquired samples requires to process a number of samples distributed over non uniform sampling periods. In fact, the first sampling period, in the acquired signal period, has a duration  $1 - \Delta t$ , while the last sampling period has a duration  $\Delta t$ . For this reason, dedicated algorithms must be implemented, which imply heavier computation burden and hence, once again, a higher cost of the employed structure, for a given bandwidth of the acquired signal.

For the above reasons, the time-domain interpolation method is not widely employed in the measurement instruments based on DSP techniques.

## Frequency-domain interpolation method

It has been shown, in the previous Chapter 3, that the leakage errors depend on the angular frequency shift  $\Delta\Omega$  (given by (3.49)) between the signal harmonic frequency and the inverse of the observation interval, and on the window shape. For a given window, and a given frequency shift  $\Delta\Omega$ , coefficients  $\alpha$  and  $\beta_h$  in (3.48) are given.

If the windows are defined in the discrete time, as done in the previous sections of this chapter, (3.49) becomes:

$$\Delta\omega = 2\pi\left(\frac{1}{2N+1} - \frac{T_s}{T}\right) \quad (4.29)$$

It is quite obvious that, if the long-range leakage error can be neglected (that is the  $\beta_h$  coefficients can be neglected), and  $\alpha$  is known, the leakage errors can be totally corrected. Unfortunately,  $\alpha$  depends on  $\Delta\omega$ , which is an unknown quantity, so that the correction is not viable in this simple way.

From the practical point of view, however, the following considerations apply:

- a cosine window with maximum asymptotic decay must be used, in order to mitigate the long-range leakage error to a negligible level;
- the width of the main lobe of the employed cosine window, in the frequency domain, is  $L$ -times wider than that of the rectangular window;
- if the observation interval is correctly selected, according to (4.22), each single spectral component of the sampled signal remains under the main lobe of the window



for  $2L - 1$  steps of convolution (3.40).

Under these conditions, the highest spectral value obtained is the one corresponding to the convolution step with the spectral component closer to the centre of the window main lobe; this value represents the signal spectral component affected by the short-leakage error. The other spectral components that appear at frequencies  $\pm 2\pi/(2N+1)$  do not represent any signal frequencies; since these frequencies are not close to the signal harmonic frequencies; they appear because of the large spectral interference of the actual spectral component (which is still located under the window main lobe at these convolutions steps) and these zero spectral components. The situation is graphically depicted in Fig. 4.13, where the main lobe of a Hanning window is considered.

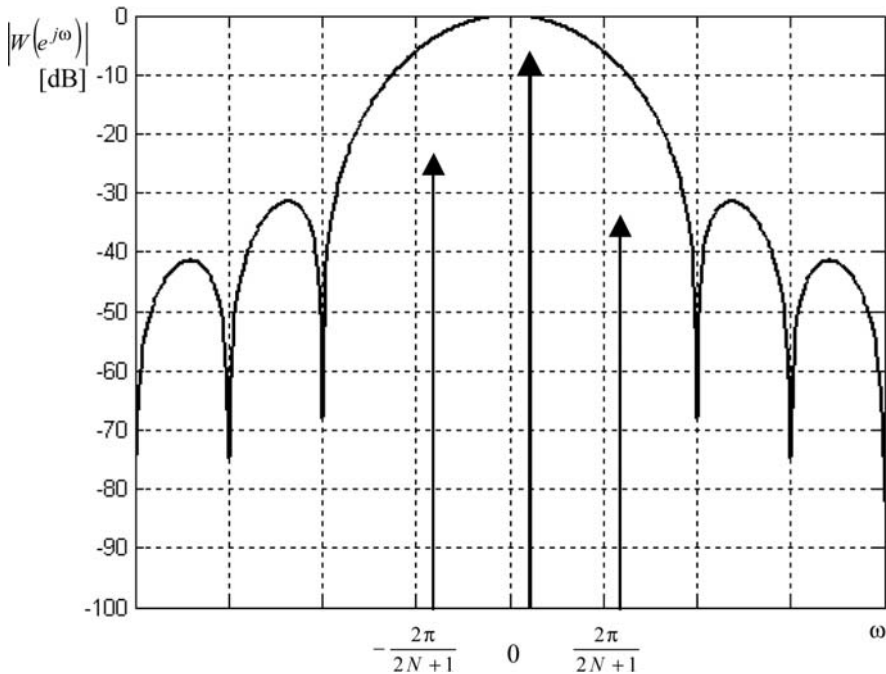


Figure 4.13. Relative position of the main lobe of a Hanning window and the obtained spectral components during the steps of convolution (3.40)

These fictitious spectral components can be usefully employed to estimate the unknown frequency shift  $\Delta\omega$ . In order to prove this statement, let us first consider the particular case of a sampled sinewave:

$$s(t) = Ae^{j\left(2\pi\frac{t}{T} + \varphi\right)} \quad (4.30)$$

Let us suppose that this sinewave is sampled with sampling period  $T_s$  over an observation interval  $T_o = MT_s$ , where the number of samples  $M$  is imposed by the sampling theorem and the order of the employed window.

Let us also suppose that the synchronous sampling condition has been met, so that  $T_o = T$ . Let us also suppose that a second order cosine window is used, so that the frequency-domain components of the sampled signal are given by the convolution of the signal spectrum with the window spectrum. Since the synchronous sampling condition is supposed to be met, the only spectral component of (4.30) falls exactly under the peak of the main lobe of the window spectrum in one step of the convolution, so that the convolution result is given by:

$$A_0 = Ae^{j\varphi_0} W(e^{j0}), \quad (4.31)$$

$W(e^{j0})$  being the peak value of the window spectrum.

Due to the width of the main lobe of the window spectrum, the signal spectral component falls under this lobe also at the previous and next steps of the convolution, as shown in Fig. 4.13. Therefore, the convolution result will be different from zero also for these two steps, even if the synchronous sampling conditions are met. In particular, the convolution result for the previous step will be given by:

$$A_{-1} = Ae^{j\varphi_0} W\left(e^{-j\frac{2\pi}{M}}\right) \quad (4.32)$$

and, similarly, the convolution result for the next step will be given by:

$$A_{+1} = Ae^{j\varphi_0} W\left(e^{j\frac{2\pi}{M}}\right) \quad (4.33)$$

It can be readily checked that, due to the symmetry of  $W(e^{j\omega})$ , (4.32) and (4.33) deliver the same result, so that:  $A_{-1} = A_{+1}$ . It is also immediate to check that:  $A_{-1} = A_{+1} < A_0$ .

This result can be extended to any cosine window: under synchronous sampling conditions, the highest obtained spectral component is the one that falls under the peak of the window main lobe and corresponds to the actual signal spectral component. However,

since the signal spectral component remains under the window main lobe for  $2L - 1$  steps of the convolution, other  $2L - 2$  fictitious components are provided by the convolution, with pair-wise amplitudes  $A_{-k} = A_{+k}$ . This last property can be practically exploited to verify, from the obtained frequency domain components of a sampled sinwave, whether it has been sampled under synchronous sampling conditions or not.

Let us now suppose that the synchronous sampling conditions have not been met, and that the frequency shift (4.29) appears between the sampling frequency and the signal frequency. The highest spectral component provided by the convolution is now given by:

$$\tilde{A}_0 = Ae^{j\varphi_0} W\left(e^{j\Delta\omega}\right) \quad (4.34)$$

Similarly, the results provided by the previous and next convolution steps are no longer given by (4.32) and (4.33), but, respectively, by:

$$\tilde{A}_{-1} = Ae^{j\varphi_0} W\left(e^{-j\left(\frac{2\pi}{M}-\Delta\omega\right)}\right) \quad (4.35)$$

and:

$$\tilde{A}_{+1} = Ae^{j\varphi_0} W\left(e^{j\left(\frac{2\pi}{M}+\Delta\omega\right)}\right) \quad (4.36)$$

Since the signal spectral component does no longer fall under symmetrical positions, with respect to the window main-lobe peak, in the previous and next steps of the convolution, it is:  $\tilde{A}_{-1} \neq \tilde{A}_{+1}$ . It is also  $\tilde{A}_{+1} > \tilde{A}_{-1}$  if  $\Delta\omega > 0$ , and  $\tilde{A}_{+1} < \tilde{A}_{-1}$  if  $\Delta\omega < 0$ .

Let us now consider the ratio between  $\tilde{A}_0$  and the greatest between  $\tilde{A}_{-1}$  and  $\tilde{A}_{+1}$ . This ratio is given by:

$$\frac{\tilde{A}_0}{\tilde{A}_{-1}} = \frac{W\left(e^{j\Delta\omega}\right)}{W\left(e^{-j\left(\frac{2\pi}{M}-\Delta\omega\right)}\right)} \quad (4.37)$$

if  $\Delta\omega < 0$ , while it is given by:

$$\frac{\tilde{A}_0}{\tilde{A}_{+1}} = \frac{W(e^{j\Delta\omega})}{W\left(e^{j\left(\frac{2\pi}{M} + \Delta\omega\right)}\right)} \quad (4.38)$$

if  $\Delta\omega > 0$ . It can be immediately recognized that both (4.37) and (4.38) depend only on the shape of the window main lobe in the frequency domain and  $\Delta\omega$ .

From a strict theoretical point of view,  $\Delta\omega$  can be obtained by inverting (4.37) (or (4.38), according to the sign of  $\Delta\omega$ ). However, due to the analytical complexity of  $W(e^{j\omega})$ , as shown by (4.10) in the case of cosine windows, this inversion is generally impractical, so that only approximate expressions are given, valid for high values of  $M$  and low order windows.

From a practical point of view, however, a number of values can be computed for (4.37) and (4.38) and stored in a look-up table, together with the corresponding values of  $\Delta\omega$ , for a given window. Once the frequency-domain components  $\tilde{A}_0$ ,  $\tilde{A}_{-1}$  and  $\tilde{A}_{+1}$  have been computed, the ratio between  $\tilde{A}_0$  and the greatest between  $\tilde{A}_{-1}$  and  $\tilde{A}_{+1}$  is also computed. The corresponding value of  $\Delta\omega$  is then retrieved from the look-up table by linear interpolation between the two stored values closest to the computed ratio.

Once  $\Delta\omega$  has been estimated, the exact frequency location of the signal spectral component can be obtained. The exact value of the signal spectral component can be also obtained from (4.34). Since the computation of  $W(e^{j\Delta\omega})$  can be impractical too, a look-up table can be employed for its estimation too.

The above described method, known as the *frequency-domain interpolation method*, has been derived under the assumption that the sampled signal was a pure sinewave. However this method can be applied also if the spectral interference is negligible. In this case, in fact, the computed values of each  $\tilde{A}_{\pm k}$  depend only on the signal spectral component that falls under the window main lobe, so that the same situation as the one described above is obtained again. In most practical situations, the spectral interference can be reduced to negligible levels if a cosine window with maximum asymptotic decay is employed.

## HARDWARE SYNCHRONIZATION

All methods described in the previous sections of this chapter allow to mitigate, at a greater or lower extent, the errors due to the spectral leakage, but none of them allows to eliminate them completely, since, according to the developed theory, this can be done only if the synchronous sampling condition (3.22) is satisfied.

This condition cannot generally be attained in an easy way because of the reasons briefly listed in the introduction to this chapter. However, it is possible to employ an

hardware device that synchronizes the sampling period to the signal period, in such a way that condition (3.22) is satisfied.

This device is basically a frequency multiplier and it must be designed in order to satisfy the following requirements:

- the ratio between the output frequency and the input frequency must be programmable, so that the proper sampling frequency can be set, in order to avoid aliasing;
- the output frequency must be stable when the input frequency is stable, in order to avoid jitter errors on the sampling period;
- the output frequency must be locked on the fundamental frequency of the input signal, not on the frequency of any other harmonic component, in order to avoid errors in the implementation of (3.22).

The hardware structure that better complies with the above requirements is the one shown in Fig. 4.14, based on a Digital Phase-Locked Loop (DPLL) structure.

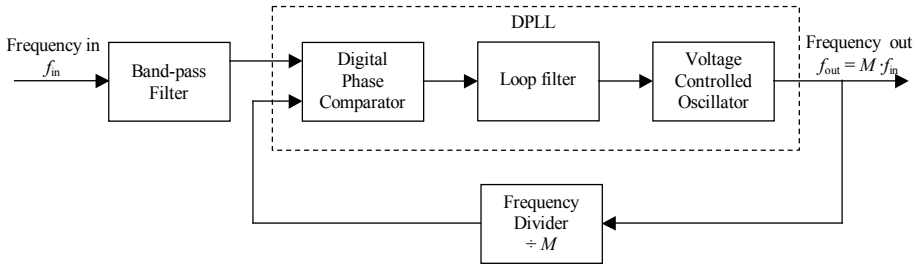


Figure 4.14. Sampling clock synchronization device

In such a structure, the frequency of the output signal is divided by  $M$  (where  $M$  is the desired multiplying factor) and compared with the frequency of the input signal by the Digital Phase Comparator (DPC). The digital output signal of the DPC is converted into an analog reference signal to the Voltage Controlled Oscillator (VCO) by the Loop filter (generally realized by a charge integrating circuit), so that the output frequency of the VCO is adjusted to  $M$  times that of the input signal. The frequency divider placed on the feedback path of the structure can be easily realized in such a way that the dividing factor  $M$  is selected by the operator within a given range, so that the first of the above requirements is satisfied.

The main advantage in using a DPC is that it cannot lock on harmonics of the centre frequency, so that the correct multiplying factor  $M$  is obtained. Once the DPLL is locked on the input signal frequency, the frequency of the output signal is at least as stable as that of the input signal, thus satisfying the second of the above requirements. Moreover, the digital nature of this device makes the output frequency intrinsically insensitive to temperature variations.

On the other hand, the DPC is characterized by a very low noise rejection. This means that noise components in the input signal can introduce a trigger jitter error or noise pulses, thus causing an error in the value of the output frequency. This problem can be

overcome by band-pass filtering the input signal to the DPC. Of course, the centre frequency of the band-pass filter must be the fundamental one of the signal to be sampled.

The presence of the band-pass filter centred around the input signal fundamental frequency limits the capability of this kind of synchronous sampling clock generator to track the frequency variations of the input signal. Actually, this limitation is inherent to the DPLL structure, which is characterized by its pull-out range, that is the dynamic limit of stable operations of the DPLL itself. The presence of the filter only makes this limitation stricter.

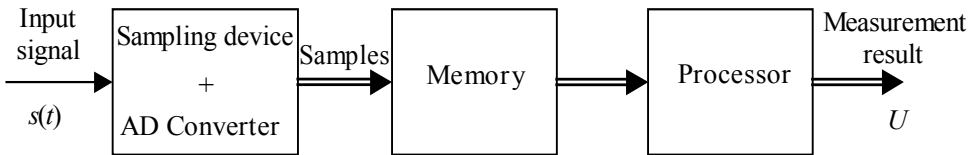
In conclusion, condition (3.22) can be met, if such a synchronous sampling clock generator is employed, only if the fundamental frequency of the input signal is stable enough not to exit the pass band of the input filter. Under these conditions, the input signal can be sampled in the theoretically ideal conditions and no leakage errors appear.

## ARCHITECTURE AND PERFORMANCE OF DSP-BASED INSTRUMENTS

### INTRODUCTION

The previous chapters have shown the mathematical basis of the sampling techniques, which represent the theoretical background for the Digital Signal Processing. In order to implement these techniques on a real instrument, a suitable architecture must be devised, capable of sampling the input signals, converting them into a digital code, store them into a memory device and, finally, process them in order to get the desired measurement result.

Such an architecture has been already shown in Chapter 1, and will be discussed more in detail in this Chapter. Its block diagram is reported in Fig. 5.1.



*Figure 5.1. Block diagram of a DSP-based instrument*

It is possible to state, in a very intuitive way, that the capability of such a structure to cope with the requirements of the DSP theory depends, in a very strict way, on the performance of each block of the structure itself. Let us consider, for instance, the sampling theorem. According to the mathematical derivations in Chapter 3, the input signal must be sampled at a rate that is at least twice the maximum frequency in the input signal spectrum. Very intuitively, the rate at which the input samples can be acquired and processed depends on the rate at which the Analog-to-Digital Converter (ADC) converts the input signal into a digital code, the rate at which the converted code is stored into the memory, and the speed at which the processor executes the measurement algorithm. Therefore, since the sampling rate of the structure in Fig. 5.1 depends on the performance of each block, and the bandwidth of this structure is upper-bounded at half the sampling rate, the analysis of the performance of each block is the key element in assessing the bandwidth of a DSP-based instrument.

On the other hand, still very intuitively, the accuracy with which the measurement result can be obtained depends basically on the accuracy with which the ADC converts the samples of the input signal into digital codes.

Therefore, it is important to have at least a rough idea on how the performance of the single blocks in Fig. 5.1 can be evaluated in order to estimate the metrological performance of a DSP-based instrument in the correct way.

This Chapter is aimed at providing such skills, by analysing the main characteristics of the blocks in Fig. 5.1. No details will be given about how these blocks are realized, since this is not the aim of this book, but only the concepts necessary to understand which feature of each block may affect the metrological performance of the whole structure.

At last, single-channel and multi-channel structures will be considered and their performance evaluated.

## THE ANALOG-TO-DIGITAL CONVERSION

### Definitions

The Analog-to-Digital (AD) conversion can be defined as the process that transforms the input analog quantity  $x$  (generally a voltage signal) into an output digital code  $Y$  (generally a binary code), as schematically shown in Fig. 5.2. Since the output code features a finite number of digits (*bits*), the conversion process gives origin to a *quantization* effect. Moreover, the whole conversion process requires a finite conversion time to be accomplished.

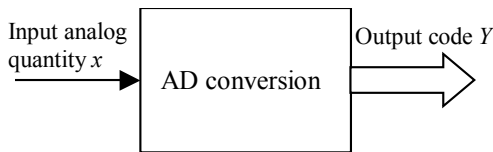


Figure 5.2. The AD conversion process

The device that realizes the AD conversion is called *Analog-to-Digital Converter* (ADC) and is generally characterized by a voltage analog input, a digital input for the clock signal that marks the conversion process and  $n$  digital outputs that provide the output code on  $n$  bits. The block diagram of such a device is shown in Fig. 5.3.

The following definitions are given for an ADC.

- Number of bits  $n$ : it represents the number of binary digits of the code into which the input signal is converted. Since  $n$  is a finite number, the ADC features only  $2^n$  possible different output codes.
- Full-scale voltage  $U_{fs}$ : it is conventionally defined as twice the value of the input voltage corresponding to the output code featuring the most significant bit (MSB) equal to one, and all other bits equal to zero. For instance, if a 3-bit ADC is



considered, this binary code is equal to 100. Let us now suppose that this code is obtained when the input voltage is  $U_h = 5$  V. The full-scale voltage is therefore twice this value, that is:  $U_{fs} = 10$  V. It is worth noting that this voltage would correspond to the binary code 1000, that cannot be issued by the ADC, since it features 3 bits only. Therefore, the full-scale voltage is greater than the maximum input voltage that can be converted by an ADC, and that corresponds, in this case of a 3-bit ADC, to the binary code: 111.

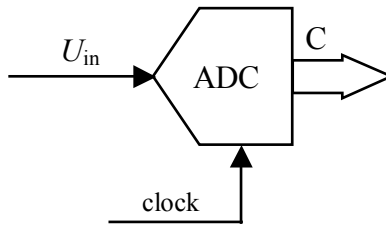


Figure 5.3. Block diagram of an ADC

- Resolution  $R$ : it represents the least variation in the input signal that is capable of causing a variation in the output code. Since the least variation in the output code corresponds to the Least Significant Bit (LSB), that is to  $2^{-n}$  of the full scale, the resolution can be written, in terms of the full-scale voltage, as:

$$R = \frac{U_{fs}}{2^n} \quad (5.1)$$

Since the value given by (5.1) represents the input voltage associated with the LSB, it can be seen also as the ADC “constant”, that is the dimensional quantity that allows to obtain the input voltage  $U$  from a given code  $C$ :

$$U = R \cdot C \quad (5.2)$$

- Conversion time  $t_c$ : it represents the time the ADC takes to convert the input voltage value into the corresponding output code. At first, the input voltage will be supposed to remain constant during the conversion time. This assumption will be removed later in this Chapter.

### The ideal ADC characteristic

According to the definitions given above, an ideal ADC provides an output code proportional to the value of the input signal, except for the error due to its finite

resolution  $R$  given by (5.1). The relationship between the analog input value and the output code can be graphically represented by the step-like characteristic drawn in Fig. 5.4 for a 3-bit, ideal ADC with full-scale voltage equal to 1 V.

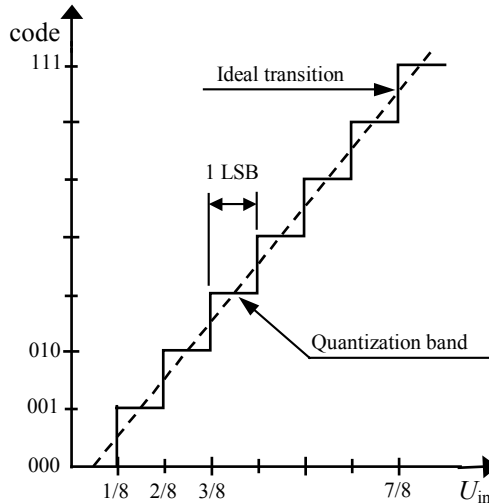


Figure 5.4. Input-output characteristic for a 3-bit ideal ADC with full-scale voltage equal to 1 V

The quantization effect is quite evident: the output code provided by the ADC does not change until the analog input value does not change for more than 1 LSB, that is the width of the *quantization band*.

A  $n$ -bit ADC features  $2^n$  quantization bands. Since no other effects than quantization are supposed to affect an ideal ADC, all its quantization bands have the same width, equal to the ADC resolution.

Due to this property, the line that joins the middle points of each quantization band (that is the geometrical locus of the middle points of all quantization bands) is a straight line, as shown by the dashed line in Fig. 5.4. For this reason, the characteristic of an ideal ADC is generally referred to as a *linear* characteristic. This is correct only if this geometrical locus is considered, otherwise a step-like characteristic as the solid one in Fig. 5.4 cannot be considered linear at all. However, it is worth noting that the step-like characteristic degenerates into the geometrical locus of the middle points of the quantization bands if the number of bits of the ADC tends to infinity: in this respect, the limit characteristic of an ideal ADC can be considered as linear.

The quantization effect gives rise to a *quantization error* that, supposing to know the actual value of the input voltage  $U_{in}$ , and supposing that  $C$  is the corresponding code issued by the ADC, can be mathematically expressed as the analog quantity:

$$\epsilon' = R \cdot C - U_{in} \tag{5.3}$$

The same quantity can be expressed also as the digital quantity:

$$\epsilon = C - \frac{U_{in}}{R}, \tag{5.4}$$

where  $R$  is the ADC resolution given by (5.1).

If the input voltage is supposed to vary from 0 to the full-scale value, error  $\epsilon$  takes the saw-tooth shape shown in Fig. 5.5.

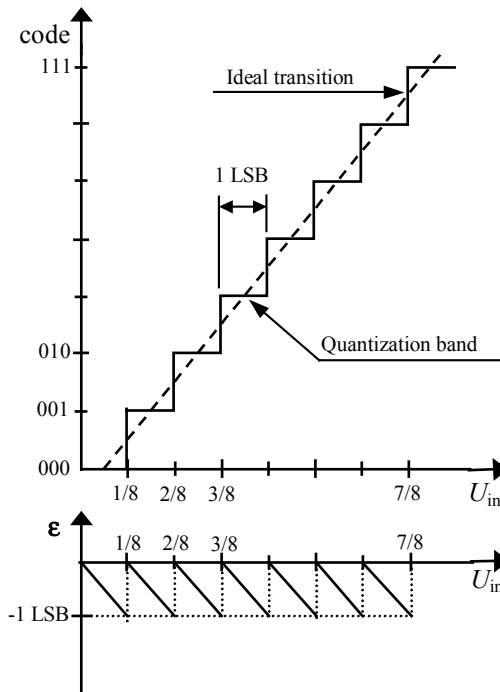


Figure 5.5. Quantization error plot

It can be readily checked that the quantization error is always negative and varies linearly in the range 0 – 1 LSB. Its average value is therefore  $-1/2$  LSB, and is therefore different from zero. This means that the quantization error is biased, so that any statistical processing of this error cannot be effectively applied in order to mitigate it.

This problem can be overcome by properly shifting the whole conversion characteristic in such a way that the first transition occurs at  $\frac{1}{2}$  LSB instead of 1 LSB, as shown in Fig. 5.6.

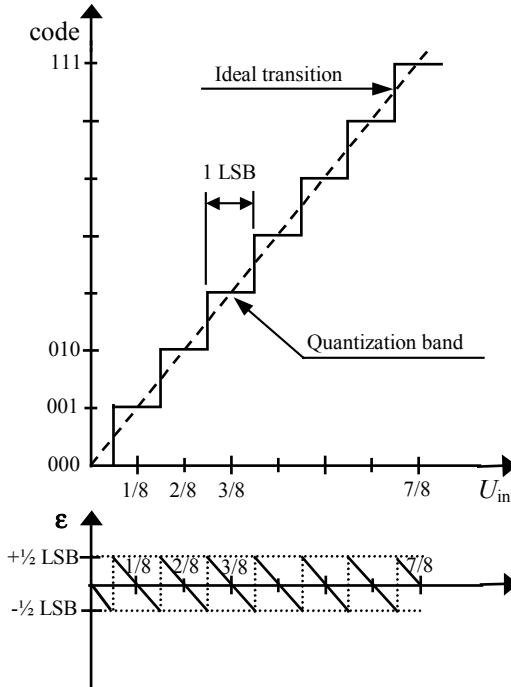


Figure 5.6. Shifted ADC characteristic, in order to have zero mean value of the quantization error

The plot in Fig. 5.6 clearly shows that the total width of the error interval is always 1 LSB, as in the case of Fig. 5.5. In this case, however, the error interval is distributed around the zero value, so that the quantization error has zero mean value in the quantization band.

### The dithering technique

Having shifted the quantization error in such a way that it is no longer biased in the quantization band allows to implement suitable statistical processing that are expected to mitigate the quantization error. The most employed technique is the so called *dithering* technique, which is based on the superposition of *dither* on the input analog signal.

The *dither* is an unbiased noise showing uniform probability distribution function in the quantization band. Therefore, being  $R$  the width of the quantization band, the probability distribution function of the dither is:

$$p_d(x) = \begin{cases} \frac{1}{R} & \text{for } U - \frac{R}{2} \leq x \leq U + \frac{R}{2} \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

where  $U$  is the voltage value to which this noise has been superimposed. Function (5.5) is graphically represented in Fig. 5.7.

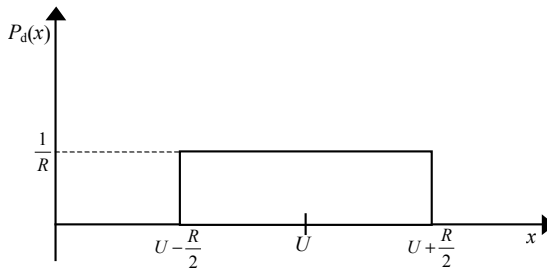


Figure 5.7. Probability distribution function of dither

Let us now consider a generic constant input voltage  $U$ , to which noise (5.5) is superimposed. The effect of noise is to make the input voltage to the ADC vary randomly in the quantization band, as graphically shown in Fig. 5.8, where the quantization band corresponding to the input voltage  $U$  is represented, together with the previous and next ones.

Because of the superposition of dither  $d(t)$ , the input voltage to the ADC is now variable with time and is given by:

$$u(t) = U + d(t) \quad (5.6)$$

Due to the property of the superimposed dither, the input signal to the ADC varies for no more than plus or minus half a quantization band around the given constant input voltage  $U$ , as shown in Fig. 5.8. Therefore, the results of different conversions will always show a quantization error within the  $\pm \frac{1}{2}$  LSB interval. This means that if a single conversion result is considered, the conversion error will remain in the same range, no matter whether the dither has been superimposed to the input signal or not.

Let us now suppose to perform several conversions. Due to the presence of dither, the input voltage value considered by each conversion  $j$  will be given by:

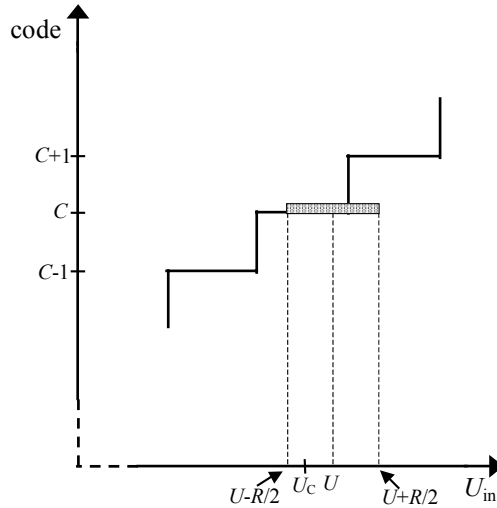


Figure 5.8. Dither effect

$$U_j = U + d_j, \tag{5.7}$$

$d_j$  being an extraction of noise value, taken with uniform probability in the  $\pm \frac{1}{2}$  LSB interval. The code issued by the ADC for the  $U_j$  input will be:

$$\begin{cases} C & \text{if } \left( U_C - \frac{R}{2} \right) \leq U_j \leq \left( U_C + \frac{R}{2} \right) \\ C-1 & \text{if } U_j < \left( U_C - \frac{R}{2} \right) \\ C+1 & \text{if } U_j > \left( U_C + \frac{R}{2} \right) \end{cases} . \tag{5.8}$$

In the example of Fig. 5.8, where the constant input voltage is  $U > U_C$ , only codes  $C$  and  $C+1$  can be issued by the ADC. Of course, the number of codes issued with value  $C$  will be proportional to the number of occurrences of input values  $U_j$  within the quantization band corresponding to code  $C$ .

If now  $N$  different  $U_j$  values are taken, their average value is given by:

$$\bar{U} = \frac{1}{N} \sum_{j=1}^N (U + d_j) = U + \frac{1}{N} \sum_{j=1}^N d_j \quad (5.9)$$

Due to the characteristic of the superimposed dither, its average value tends to zero as  $N$  tends to infinity. Therefore, the average value  $\bar{U}$  of the input values supplied to the ADC will tend to  $U$ . Similarly, the average value of all issued codes  $C_j$  will be given by:

$$\bar{C} = \frac{1}{N} \sum_{j=1}^N C_j = \frac{\bar{U}}{R} \xrightarrow{N \rightarrow \infty} \frac{U}{R}, \quad (5.10)$$

thus approaching, by the scale factor  $R$ , the actual value  $U$  of the input voltage. Of course,  $N$  will be never infinite, so the actual value  $U$  can never be obtained, but only approximated, and the approximation will be the better, the larger is  $N$ .

The dithering technique is therefore very helpful whenever an averaging algorithm can be applied on the acquired samples, since it uncorrelates the quantization errors on the different samples, so that the quantization error on the mean value of the samples is reduced.

## The real ADC characteristic

The conversion characteristic of a real ADC differs, in general, from that of an ideal ADC, shown in Fig. 5.4, because of different effects that cause new conversion errors to be added to the quantization error.

### *The offset error*

Due to offset phenomena occurring in the electronic circuits that realize an ADC, the conversion characteristic may be shifted with respect to the ideal characteristic, as shown in Fig. 5.9.

It can be readily checked that the result of any conversion performed by the ADC is affected by the same additive *offset* error. This means that, if  $C$  is the correct code, corresponding to the input voltage  $U$ , the actual issued code  $C'$  is given by:

$$C' = \text{INT}(C + \epsilon_0), \quad (5.11)$$

where  $\epsilon_0$  is the offset error, expressed in terms of LSB. The same effect can be considered as if it were affecting the input voltage, that is as if the actual input voltage were given by:

$$U' = U + \epsilon'_0 \tag{5.12}$$

where  $\epsilon'_0$  is the same offset error as in (5.11), but now expressed in terms of the input voltage.

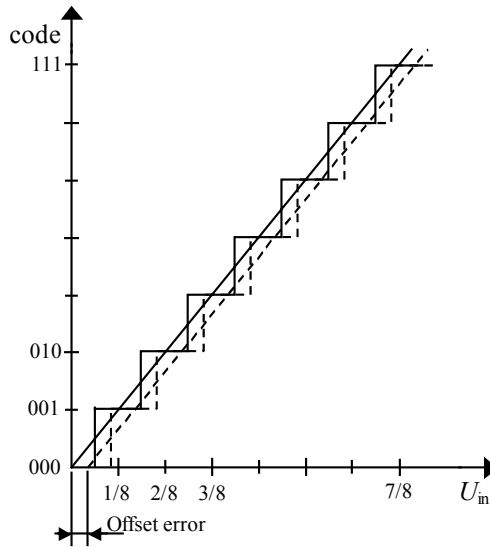


Figure 5.9. ADC characteristic in the presence of an offset error (dashed line) and ideal characteristic (solid line)

It can be also seen that, if the real characteristic is shifted on the right, with respect to the ideal one, as shown in Fig. 5.9, each issued code is lower than the correct ideal one; therefore, in this situation, error  $\epsilon_0$  in (5.11) is negative. The opposite occurs if the real characteristic is shifted on the left.

Generally, the offset error can be compensated if suitable compensation circuits are realized within the ADC itself. In this case, the residual offset error, after calibration, is confined in the  $\pm \frac{1}{2}$  LSB range.

### The gain error

Another possible way the real characteristic can be changed with respect to the ideal one, because of imperfections in the ADC circuits, is shown in Fig. 5.10, where the real characteristic is rotated, with centre in the axes origin, with respect to the ideal one.

It can be readily checked that the result of any conversion performed by the ADC is affected by the same multiplicative *gain* error. This means that, if  $C$  is the correct code,



corresponding to the input voltage  $U$ , the actual issued code  $C'$  is given by:

$$C' = \text{INT}(C \cdot \epsilon_g), \tag{5.13}$$

where  $\epsilon_g$  is the gain error, expressed in terms of LSB. The same effect can be seen again as if it were affecting the input voltage, that is if the actual converted input voltage were:

$$U' = U \cdot \epsilon'_g, \tag{5.14}$$

where  $\epsilon'_g$  is the same gain error as in (5.13), now expressed in terms of the input voltage.

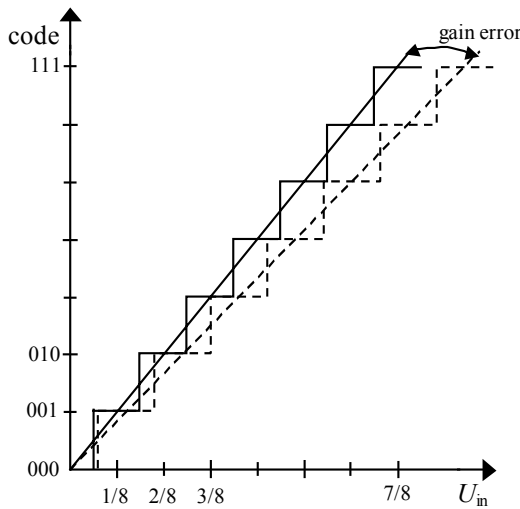


Figure 5.10. ADC characteristic in the presence of a gain error (dashed line) and ideal characteristic (solid line)

In order to better quantify the gain error, it can be noted that, if the real characteristic is rotated clockwise, as shown in Fig. 5.10, each issued code is lower than the ideal one; therefore, the gain error is lower than 1. On the contrary, if the real characteristic is rotated counter-clockwise, the gain error is greater than 1.

In a similar way as the offset error, the gain error can be compensated too by suitable compensation circuits. In this case, the residual gain error, after calibration, is confined in the  $\pm \frac{1}{2}$  LSB range.

*The non-linearity error*

The ideal ADC characteristic features quantization bands with the same width, equal to 1 LSB. However, due to imperfections in the electronic circuit, a real ADC may feature different widths for the different quantization bands, as shown in Fig. 5.11.

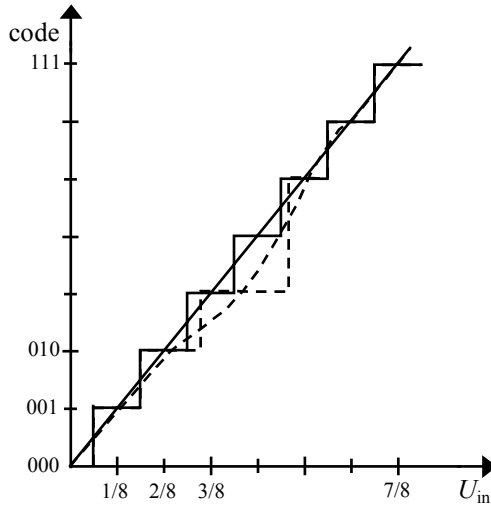


Figure 5.11. ADC characteristic in the presence of a non-linearity error (dashed line) and ideal characteristic (solid line)

Due to the different widths of the quantization bands, the geometrical locus of the middle points of all quantization bands is no longer a straight line, as in the case of the ideal ADC. For this reason, the error caused by the different widths of the quantization bands is called *non-linearity* error.

The most immediate way to characterize the non-linearity error is to evaluate, for each quantization band  $j$ , the difference:

$$\epsilon_{nl_j} = W_j - 1\text{LSB} \tag{5.15}$$

where  $W_j$  is the actual width of the considered quantization band, expressed in terms of LSB. However, in this way  $2^n$  different errors should be processed for a  $n$ -bit ADC and this is not efficient. For this reason, the maximum difference  $\epsilon_{nl} = \max(\epsilon_{nl_j})$  is considered and is referred to as the *differential non-linearity error*.

The non-linearity error may also have a quite critical consequence when, for a generic quantization band  $j$ , it is:  $\epsilon_{nl_j} \geq 1 \text{ LSB}$ . In this case, in fact, the width of band  $j + 1$  is

zero, and hence the code corresponding to this band is never issued. This situation is graphically depicted in Fig. 5.12 and is generally referred to as the *missing code error*.

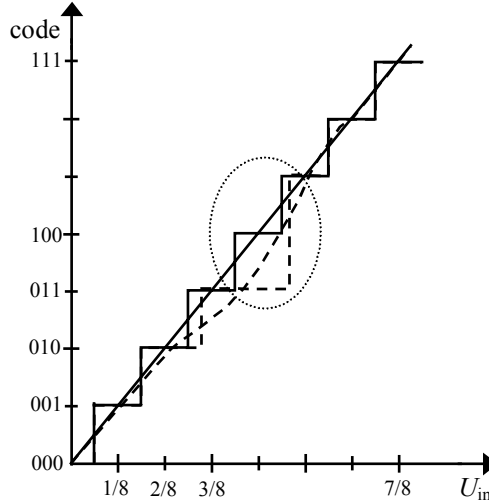


Figure 5.12. Missing code error due to a large non-linearity error: code 100 is never issued, since the non-linearity error on band 011 is greater than 1 LSB

The above defined differential non-linearity error, being only the maximum of all possible non-linearity errors on the real ADC characteristic, does not provide a complete description of the ADC behaviour. For this reason, the *integral non-linearity error* is defined as the integral of all non-linearity errors on the whole dynamics of the ADC. Since the ADC dynamics is given by a finite ( $2^n$ ) number of quantization bands, the integral non-linearity error can be expressed as:

$$\epsilon_i = \sum_{j=0}^{2^n-1} \epsilon_{nl_j} \quad (5.16)$$

When the two different non-linearity errors, the differential and the integral ones, are considered together, they provide a more detailed description of the ADC behaviour. In fact, a large integral non-linearity error together with a small differential non-linearity error shows that each quantization band does not differ too much from the ideal one (small differential non-linearity error), but all (or almost all) errors have the same sign (large integral non-linearity error). This situation is quite critical when the data converted

by the ADC are integrated by the measuring algorithm. In this case, in fact, the resulting error can be as large as  $\epsilon_i$  if the input signal spans over the whole dynamics of the ADC.

On the contrary, when a large differential non-linearity error goes together with a small integral non-linearity error, at least one quantization band differs from the ideal one significantly (large differential non-linearity error), but the signs of errors  $\epsilon_{nl_j}$  are almost equally distributed between the negative and positive ones, so that the integral non-linearity error is small. This situation is not critical when the measuring algorithm integrates the converted data, but becomes critical when, for instance, it derivates them.

Since the non-linearity errors involve each quantization band, and not the whole characteristic, as the offset and gain errors, they cannot be reduced by means of compensation circuits. However, they too are generally confined in the  $\pm \frac{1}{2}$  LSB range for good quality ADCs.

### ADC behaviour in the presence of a variable input signal

All concepts presented in the previous sections have been derived under the assumption that the input signal to the ADC remains constant during the conversion interval and therefore they are strictly correct if and only if the above assumption is verified.

However, this assumption is quite strong, especially from the practical point of view, since most of the measurement applications, and in particular those requiring a complex processing of the input signals that do hence benefit of the DSP techniques, deal with variable signals.

On the other hand, if the finite resolution of the ADC is taken into account, it is possible to loosen the above requirement. In fact, the ADC cannot sense any variation in the input signal that remains within the resolution of the ADC itself. Therefore, it is possible to state that the above assumption can be reformulated so that the input signal is not required to remain constant any longer, but more simply to show, in the conversion time, variations that are lower than the ADC resolution.

According to this new formulation, it is interesting to evaluate if an ADC can be still used to convert variable signals and under which conditions. This can be done by considering a sinusoidal input signal, with frequency  $f$  and with a peak-to-peak amplitude equal to the ADC full-scale value:

$$u_{in}(t) = \frac{U_{fs}}{2} \sin(2\pi ft) \quad (5.17)$$

In order to check whether this signal can be converted correctly or not, the maximum variation of the signal, within the conversion time  $t_c$ , must be considered and compared with the ADC resolution.

Since the input signal is sinusoidal, its maximum variation occurs at zero crossing. Moreover, if the conversion time is supposed to be much lower than the input signal

period, as is the case in most practical situations, the sine value can be approximated by the arc, so that the maximum signal variation is given by:

$$\Delta u_{\max} = U_{\text{fs}} \pi f t_c \quad (5.18)$$

Let us now suppose that the input value falls at the centre of the corresponding quantization band. The maximum allowed variation in order to stay within that quantization band is hence  $\pm \frac{1}{2}$ LSB. Therefore, if (5.18) is taken into account, it is:

$$U_{\text{fs}} \pi f t_c \leq \frac{1}{2} \frac{U_{\text{fs}}}{2^n} \quad (5.19)$$

From this equation, the maximum frequency allowed for the input signal, in order to perform the conversion correctly, for a given conversion time  $t_c$  and a given number of bits  $n$ , is given by:

$$f = \frac{1}{2^{n+1} \pi t_c} \quad (5.20)$$

Fig. 5.13 shows the plot of the maximum allowable frequency (5.20) vs. the conversion time  $t_c$ , for different given resolutions.

As an example, let us consider a 12-bit ADC, featuring a conversion time  $t_c = 1 \mu\text{s}$ . This is a typical, high-quality ADC used in many instruments based on DSP techniques. If (5.20) is applied, the resulting maximum allowable frequency is  $f = 38.86 \text{ Hz}$ . It is quite evident that this frequency is very low, and that the result provided by (5.20) prevents to employ DSP-based techniques in almost every practical measurement application, unless suitable countermeasures are taken.

### *The Sample & Hold devices*

The previous section proved that only slowly variable signals can be converted into digital by an ADC without introducing any additional error to those introduced by the conversion characteristic and already described.

It has been also proved that the source of this additional error is represented by the variation of the input signal, during the conversion process, when it exceeds the ADC resolution.

These considerations lead, in a quite straightforward way, to identify a very effective way for overcoming the problem. In fact, if the problem is the variation of the input signal during the conversion process, the most immediate countermeasure is to employ a sort of *analog memory* for keeping the input signal constant, or at least reduce its

variation to less than the ADC resolution, for a time interval equal to the ADC conversion time.

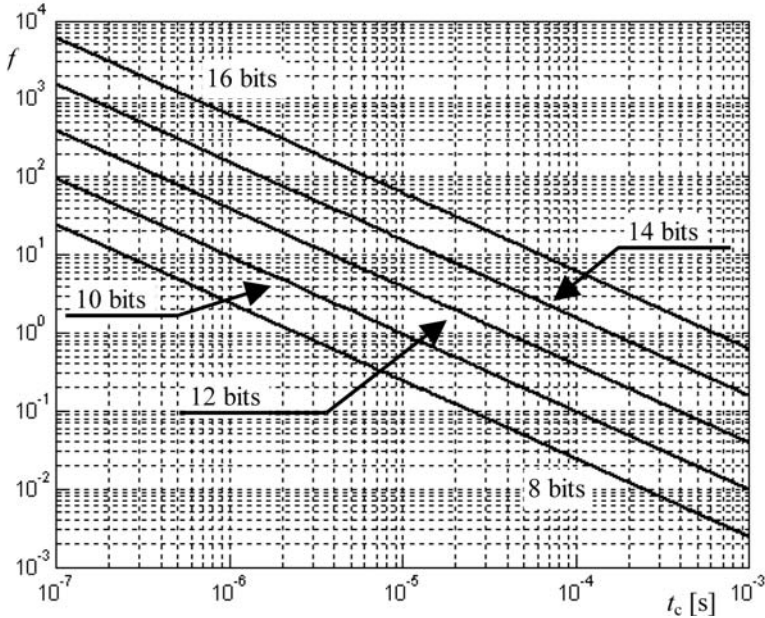


Figure 5.13. Maximum allowable frequency for the input signal to the ADC, in order to avoid signal variations for more than the ADC resolution in the conversion interval

Since the input signal to the ADC is generally a voltage signal, the *analog memory* can be easily realized, in principle, by employing a capacitor and a switch. When the switch is closed, the capacitor is connected to the signal source and is charged (*sampling stage*). When the switch is open (*hold stage*), the charged capacitor holds the voltage to which it has been charged, that is the voltage value provided by the input signal at the switch open time, for an ideally infinite period of time.

Of course, this is true only for an ideal capacitor, and with a null resistance of the charging circuit and an infinite input resistance for the ADC, so that the capacitor is not discharged during the hold stage. In practical situations all this does not apply, so that a slightly more complex circuit, employing two voltage followers, must be implemented, as shown in Fig. 5.14. The resulting device is called the *Sample & Hold* device.

In such a device, when switch S is kept close by the control input, the voltage follower connected to the input signal  $U_{in}$  transfers this voltage to capacitor C, charging it. The

charge time constant  $\tau_{ch}$  is given by the very low output resistance of this voltage follower.

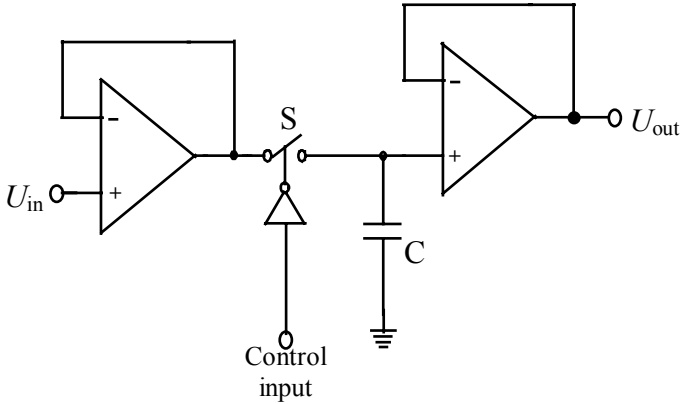


Figure 5.14. Sample & Hold device

The accuracy with which the input voltage is transferred to the terminals of capacitor C depends on  $\tau_{ch}$  and the time interval  $t_{ch}$  during which switch S remains closed. In fact, the voltage value  $U_c$  found at the capacitor terminals at time  $t_{ch}$  is given by:

$$U_c = U_{in} \left( 1 - e^{-\frac{t_{ch}}{\tau_{ch}}} \right) + U_0 e^{-\frac{t_{ch}}{\tau_{ch}}}, \quad (5.21)$$

where  $U_0$  is the initial voltage on the capacitor. Therefore, at the end of the sampling stage, the voltage found at the capacitor terminals differs from the input voltage by a quantity:

$$\Delta U_s = U_c - U_{in} = (U_0 - U_{in}) \cdot e^{-\frac{t_{ch}}{\tau_{ch}}} \quad (5.22)$$

This last equation shows that the accuracy depends directly on the difference between the last sampled value and the new one, and can be improved by reducing the charge time constant and increasing the sampling time. However, for a given capacitor, the reduction of the charge time constant is limited by the value of the output resistance of the

employed voltage follower, which is small, but not zero. Similarly, the increment of the sampling time is upper bounded by the bandwidth requirements of the whole instrument, as it will be shown later in this Chapter.

At the end of the sampling stage, switch S in Fig. 5.14 opens, so that capacitor C is no longer supplied by the input signal. The voltage value at which it has been charged during the sampling stage is now transferred to the output signal  $U_{\text{out}}$  by the voltage follower on the right of Fig. 5.14.

Due to the input resistance of this voltage follower, which is high, though not infinite, capacitor C will slowly discharge in time according to a discharge time constant  $\tau_d$ , so that the output voltage is given by:

$$U_{\text{out}}(t) = U_c \cdot e^{-\frac{t}{\tau_d}} \quad (5.23)$$

It is therefore possible, from (5.23), to evaluate the maximum time interval, in the hold stage, for which the variation in the output voltage is lower than the ADC resolution. In fact, the maximum allowed variation is given by:

$$U_c \left( 1 - e^{-\frac{t}{\tau_d}} \right) = \frac{U_{\text{fs}}}{2^{n+1}} \quad (5.24)$$

from which we get, for a positive value of  $U_c$ :

$$t = -\tau_d \cdot \ln \frac{U_c - \frac{U_{\text{fs}}}{2^{n+1}}}{U_c} \quad (5.25)$$

In order to avoid conversion errors, (5.25) must provide a time value  $t > t_c$ , where  $t_c$  is the ADC conversion time. Since the input resistance of the voltage follower is given, once the voltage follower has been chosen, this condition can be achieved by increasing the capacitance of capacitor C. However, according to (5.22), an increment of this capacitance causes, for a given charging time  $t_{\text{ch}}$ , a decrease of the accuracy with which the input signal is sampled. Therefore, the correct choice of C is a trade-off between a good accuracy in the sampling stage, and a low variation with time of the output voltage in the hold stage.

Generally, Sample & Hold devices are available, where the two voltage followers, the switch and its control circuits are integrated in a single integrated circuit, and an external capacitor C, properly chosen by the user, is connected to this integrated circuit.



## ADC behaviour in the presence of noise

It has been already shown that the only error affecting an ideal ADC is the quantization one. Moreover, if the ADC characteristic is properly positioned in the conversion plane, as shown in Fig. 5.6, the quantization error is distributed over the quantization band  $\pm \frac{1}{2}$  LSB and has zero mean value.

In an ideal ADC this means that, if the input voltage is uniformly distributed over the quantization band, the quantization error is uniformly distributed too over the same band. Under this assumption, that can be considered also for the real ADC, at least as a first approximation, the mean square quantization error is the same as its variance and is given by:

$$s_{\text{ADC}}^2 = \frac{1}{R} \int_{-\frac{R}{2}}^{\frac{R}{2}} x^2 dx = \frac{R^2}{12} \quad (5.26)$$

Taking into account (5.1), (5.26) can be written in terms of the converter full scale voltage and bit number:

$$s_{\text{ADC}}^2 = \frac{U_{\text{fs}}^2}{12 \cdot 4^n} \quad (5.27)$$

Equation (5.27) represents the square mean value of the quantization error of an ideal ADC with  $n$ -bit resolution. However, a second interpretation can be assigned to this equation.  $s_{\text{ADC}}^2$  can be seen as the mean square value of a uniformly distributed noise that is added to the input signal of an ideal, infinite-resolution ADC, so that only a finite number  $2^n$  of output codes are issued by the ADC itself.

If this second interpretation is followed, (5.27) can be solved with respect to  $n$ , since the quantization effect is now seen as an effect of a particular noise, which is added to the input signal internally to the ADC. The following equation is hence obtained:

$$4^n = \frac{U_{\text{fs}}^2}{s_{\text{ADC}}^2} \cdot \frac{1}{12}$$

which leads to:

$$n = \log_4 \left( \frac{U_{\text{fs}}^2}{s_{\text{ADC}}^2} \right) - \log_4(12) = \log_4 \left( \frac{U_{\text{fs}}^2}{s_{\text{ADC}}^2} \right) - 1.7925 \quad (5.28)$$

The argument of the logarithm in (5.28) is very similar to a Signal-to-Noise-Ratio (SNR), if only its numerator were a mean square value. It is possible to take into account a SNR by supposing that the ADC input signal is a sinewave, with peak-to-peak amplitude equal to the ADC full-scale voltage. Therefore, the input signal rms value is given by:

$$U = \frac{U_{fs}}{2 \cdot \sqrt{2}} \quad (5.29)$$

If (5.29) is considered, (5.28) becomes:

$$\begin{aligned} n &= \log_4 \left( \frac{8 \cdot U^2}{s_{ADC}^2} \right) - 1.7925 = \log_4 \left( \frac{U^2}{s_{ADC}^2} \right) + \log_4(8) - 1.7925 \\ &= \log_4 \left( \frac{U^2}{s_{ADC}^2} \right) - 0.2925 \end{aligned} \quad (5.30)$$

This last equation does not provide, up to now, any further information with respect to (5.27), from which it has been directly obtained. However, it is very useful because it allows to take into account the effect of other kinds of noise that may disturb the input signal, beyond the quantization one.

In fact, let us suppose that a noise, with mean square value  $s_a^2$ , is superimposed to the input signal. Let us also suppose that this noise and the quantization noise are not correlated at all, so that the mean square value of the overall noise is given by:

$$s^2 = s_{ADC}^2 + s_a^2 \quad (5.31)$$

If this noise is considered in (5.30), instead of the simple quantization noise, the following number of bits is obtained:

$$n_{\text{eff}} = \log_4 \left( \frac{U^2}{s^2} \right) - 0.2925 \quad (5.32)$$

Fig. 5.15 shows that, for a given quantization noise, corresponding to a given rated number of bits (and hence a given rated resolution), the *effective* number of bits  $n_{\text{eff}}$  decreases as the input signal SNR decreases.

In a very intuitive way, this can be explained by considering that, as the input noise increases, it forces the less significant bits of the output code to change because of the noise level and not because of real changes in the input signals. Therefore, the sensitivity

of the ADC to the input signal variations is limited by the presence of noise and, in this case, it is worthless to have a high rated resolution of the ADC, because its actual one can be much lower.

Fig. 5.15 shows also that the effective resolution of the ADCs tends to be the same as the input signal SNR decreases, no matter on the nominal number of bits of the ADC itself.

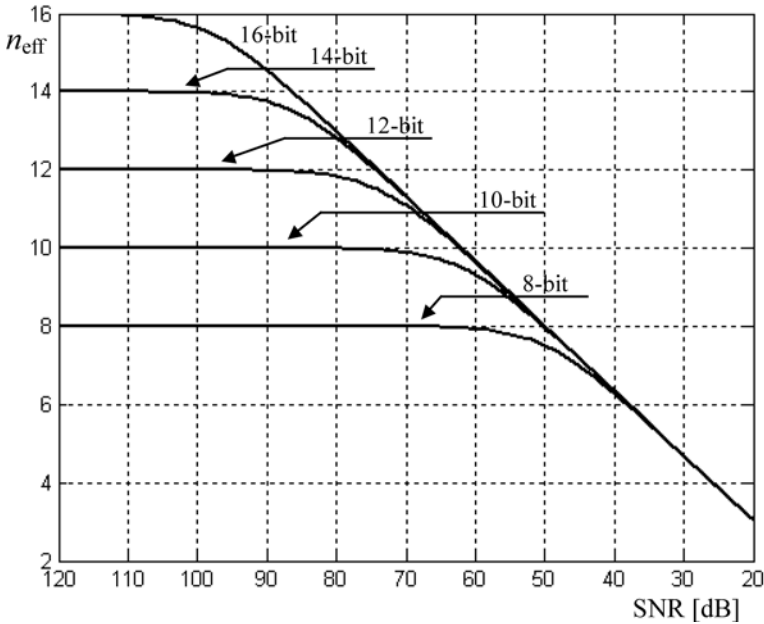


Figure 5.15. Plot of the ADC effective resolution vs the input signal SNR

The concept of the effective bit number is very important, because it shows, in a very clear and effective way, the role that the noise superimposed to the input signal plays in assessing the actual ADC resolution.

It is extremely important to consider that part of the noise is added to the input signal inside the ADC itself, due to the intrinsic noise of all electronic devices that realize the ADC circuit. Therefore, the resolution of an ADC is upper limited mainly by the noise level of the electronic components used to realize the ADC itself.

This leaves a very small margin to the noise that can be superimposed to the input signal before entering the ADC, especially when a nominal resolution greater than 12 bits is considered. Generally, the noise level generated inside the ADC is given by the manufacturer, and must hence be taken into account by the final user of the ADC,

because it allows to evaluate the noise margin allowed on the input signal, before entering the ADC, in order to preserve the ADC nominal resolution.

## ADC architectures

Several circuitual architectures can be employed to realize the Analog-to-Digital conversion with the properties shown in the previous sections. The discussion of these architectures and the analysis of their electronic circuits is not the aim of this book: a great number of dedicated textbooks is available and the interested reader can refer to them in order to study the ADCs more in depth.

The following short sections are aimed at providing some basic indications about the main features of the most used architectures, in order to give the reader some useful indications in their correct selection.

### *The integrating converters*

These converters generally perform a voltage-to-frequency, or a voltage-to-time conversion by integrating the input signal. Since, when the input signal is constant, its integral is a ramp, they are also called ramp, or slope converters. According to the particular kind of the internal integrating circuit, they can be single-slope, dual-slope or multiple-slope converters.

Due to the internal integrator, these converters feature a low internal noise, and therefore they have a very good resolution, and a high linearity. The main drawback of these converter is the relatively high conversion time, which ranges from a few milliseconds to a hundred of milliseconds.

Due to their intrinsic slowness, these converter are not generally used in DSP-based instruments, unless a very high accuracy and linearity is required on a very low frequency band. On the other hand, they are widely used to realize Digital Voltmeters (DVM) and Digital Multimeters (DMM) with a resolution up to 8 digits.

### *The successive approximation converters*

These converters are presently the most used ones in a great variety of practical applications. They are based on a closed-loop structure, where the feedback path is realized by a  $n$ -bit resolution Digital-to-Analog Converter (DAC) and the value of the input analog signal is approximated by means of a binary search algorithm. Therefore, the input value is converted into a  $n$ -bit digital code with only  $n$  iterations.

Using such a structure it is possible to reach a resolution as high as 18 bits, with a conversion time in the order of the microsecond. In the majority of practical applications, 12-bit to 16-bit resolution ADCs are used with a conversion time as short as some hundreds of nanoseconds.

The offset and gain errors can be easily confined within the  $\pm\frac{1}{2}$  LSB range by properly calibrating the device. The non-linearity errors are intrinsic of this structure and

arise in the DAC structure. However, high-quality devices may feature differential and integral non-linearity errors in the LSB range, so that missing code errors are avoided.

### *The flash converters*

These converters are the fastest ones, being based on the direct comparison of the input value with all quantization levels. This high conversion speed, in the order of the nanosecond, is paid in terms of a high complexity of the electronic circuit, that limits the resolution to 8 bits.

The complexity of the electronic circuit, and the presence of a great number of elements that must nominally show the same characteristic and the same behaviour are also responsible for a relatively large non-linearity error, which tends to increase as the resolution increases.

The flash converters are mainly used in the digital oscilloscopes, where a very high conversion speed is needed to ensure a wide frequency band, but the resolution does not represent a strict requirement, since it cannot be fully exploited in the available graphic display.

### *The $\Delta\Sigma$ converters*

The  $\Delta\Sigma$  converters represent the last generation of the ADCs. They are realised by an analog modulator, based on an integrating structure, followed by a digital, low-pass FIR filter.

The analog modulator converts the input signal into a digital signal (the so-called *bit stream*), whose duty cycle, and hence its average value, is proportional to the input value. This digital signal features also a very interesting characteristic in the frequency domain: all noise components are shifted to the high frequencies, so that the signal spectral components at the lower frequencies are almost noise free.

If the cut-off frequency of the digital filter is suitably selected, the filter plays the double role of removing the noise components and extracting the average value from the bit stream signal, so that the output code is proportional to the input signal value.

The number of bits of the output code is given by the resolution with which the digital filter processes its data and is limited only by the residual noise introduced by the modulator in the signal band.

Presently, the  $\Delta\Sigma$  converters attain a resolution up to 24 bits, though their conversion time is limited in the order of ten microseconds. A reduced conversion time can be obtained with a lower resolution, but it remains still longer than the one typical of the successive approximation converters.

A further advantage of the  $\Delta\Sigma$  converters is their very low, virtually zero non-linearity error, due to the integrating structure of the modulator.

## THE PROCESSING UNIT

The processing unit is represented by the last two blocks on the right in the block diagram of Fig. 5.1, and accomplishes the task of processing the samples coming from the ADC unit, according to a specific measurement algorithm, in order to provide the final measurement result.

Of course, the processing unit and its performances can be described and analysed using the well known analysis tools usually employed to represent a computing system. However, for this particular application of Digital Signal Processing, only a few features are critical in assessing the overall performance of the DSP-based instrument, and will be briefly analysed in this section.

The most critical feature, as far as the instrument bandwidth is concerned, is the memory access time, the computation speed of the processor and its capability of parallelizing the internal operations. If the final accuracy and resolution are considered, the most important feature is the number of bits of the data bus, and the processor capability of processing data represented with this resolution with single elementary instructions.

For this reason, the block diagram in Fig. 5.1 considers the processor and the memory in which the sampled data are stored as two separate blocks.

Before shortly analysing the main features of these two blocks, it is worth while defining two possible operating modes of a DSP-based instrument, since they can deeply affect the performance required to the processing unit: *real-time* operation and *deferred* operation.

A measuring instrument operates in real time when the measurement result is provided at the end of the observation period, before the successive observation interval expires. This means that real-time operations are closely dependent on the required observation period, and the performance requested to the processing unit can be more or less strict depending on the required measurement, and not only on the characteristics of the input signal.

As an example, let us consider the implementation of a digital filter. In this case, the observation period is the sampling period, since, as it will be shown in the next Chapters, an output sample is expected at each new input sample. Therefore, the processing unit must be able to process the whole filter algorithm within one sampling period.

On the other hand, let us consider the implementation of an averaging algorithm, such as, for instance, the instantaneous power averaging over the signal period in order to measure the electric active power under periodic conditions. In this case, a whole signal period is needed, at least, as the observation period. Therefore the processing unit must be able to process the acquired samples while the next signal period is sampled and acquired and complete the measurement algorithm within the next signal period in order to track the active power evolution and provide real-time measurement. Similar considerations still apply, for example, for rms measurement and frequency-domain component measurement.

Let us now consider an instrument that samples the input signals over the observation period, and stores the acquired samples into its memory, without processing them. At the

end of the observation period, the instrument stops sampling the signals, and starts processing the acquired samples. Once the measurement algorithm has been completed and the measurement result provided, the instrument starts sampling the input signals again. This kind of operation is called *deferred* operation, because the measurement result is not necessarily obtained within the next observation period and the input signals are not acquired in a continuous way.

Deferred operations are very useful in laboratory measurements, when the measurement set-up allows repetitive operations under steady state conditions. Under these conditions, there is no need to track the input signals in real-time, since they are stable in the successive observation periods. Therefore, all processor resources can be devoted to a fast acquisition of the input signals, and all processing can be deferred after the observation period has expired. Moreover, complex algorithms can be executed, since no time constraints are present.

On the contrary, real-time operations are strictly required for in-field measurements, when no assumptions can be made about the repeatability and stability of the input signals along the successive observation periods and about the absence of transient phenomena. Therefore, the input signals must be tracked continuously, and a new measurement value is required at the end of each observation period.

In this last case, of course, part of the processor resources must be devoted to processing the input signals, and not only to their acquisition. Since this turns into a reduction of the sampling rate, as it will be shown in the next sections, suitable tricks must be adopted in order to optimise the instrument performance.

## The processor

Basically, every kind of processors can be employed in a DSP-based instrument, and a wide variety of solutions is actually available.

At the beginning, the first applications of the DSP techniques to measurement operations employed microprocessor architectures as the processing unit. The major drawback of these architectures is, still now, their relatively low computation speed, so that they are not suitable for real-time operations, unless the measurement algorithm is very simple.

More recently, the microprocessor architectures have evolved towards the Digital Signal Processor (DSP)<sup>2</sup> architecture. The internal architecture of a DSP is a special modification of the standard internal architecture of a microprocessor, realized in such a way to allow a very fast implementation of the multiply and accumulation operations, that are the fundamental operations of the Digital Signal Processing techniques.

---

<sup>2</sup> The same acronym (DSP) is universally used for indicating both the device (Digital Signal Processor) and the technique (Digital Signal Processing). Attention must be paid to the context, in order to understand whether the device or the technique is considered.

This modification is basically the duplication of some internal devices, so that parallel operations can be performed and the execution time optimised. Presently, the modern DSPs feature up to some hundreds MOPS (Million Operations Per Seconds) and even MFLOPS (Million Floating-point Operations Per Second) and simultaneous access to at least two different memory banks, and handle data coded on up to 32 bits, in fixed and floating-point notation.

More recently, different structures have evolved to performances that make them suitable for DSP-based measurement applications: structures based on ASICs and FPGAs.

An ASIC (*Application-Specific Integrated Circuit*) is a particular integrated circuit with a number of elementary specific circuits that are not connected to each other. Different connections can be realized by the final user by properly configuring the ASIC in a similar way as a memory is programmed. In this way, complex functions can be implemented by direct hardware programming, so that very fast structures are obtained.

An FPGA (*Field-Programmable Gate Array*) is a particular integrated circuit containing thousands of gates that implement the elementary logical operations. The connections between the gates can be programmed by the final user, so that very complex logical functions can be implemented and even basic algorithms. Similarly to an ASIC, a specific function can be implemented on an FPGA by suitably programming its hardware architecture. Very fast operations can therefore be obtained.

It can be readily understood that architectures based on these structures, and in particular architectures obtained by a suitable implementation of all these techniques, are particularly suitable for real-time operations, due to their optimized internal structure and their computation speed. Their major drawback is that they generally require long programming operations and, being tailored on a specific measurement function, they cannot be adapted to new functions in a flexible way.

An alternative solution, when a high flexibility is required to the measurement system, is the use of a PC as the processing unit of the DSP-based instrument. The performances of the CPUs used in modern PCs are sufficiently high to virtually allow also real-time operations. The main problem, when real-time operations must be assured, is given by the operating system running on the PC: generally, the operations performed by the operating system are not under control of the final user, so that it is not possible to keep the program flow of the measurement application under total control.

On the other hand, the performances of a modern PC in terms of computational capabilities, graphic interface, mass storage and user-friendly man-machine interface allow a relatively easy and immediate programming of the measurement functions, even when they are extremely complex. Development tools are presently available with large comprehensive libraries of measurement functions and graphic symbols, so that the implementation of a new measurement function, or the modification of an existing one are quite immediate.

PC-based instruments are particularly suited for lab applications, where flexibility requirements are generally more important than real-time ones. On the contrary, DSP and hardware configurable architectures are best suited for in-field applications, where real-time requirements are much more important than flexibility, since changes in the



measurement functions are generally very few, but the input signals must be tracked continuously.

## The memory

The main requirement for the memory block is to ensure an access as fast as possible, so that the samples coming from the ADC can be stored in a very short time. The access time for the memory write operation is a critical parameter in assessing the sampling period both in real-time and deferred operations, and must be therefore carefully selected.

In general, Direct Access Memory (DMA) operations are preferable for data transfer from the ADC to the memory. When real-time operations are required, the use of a dual-port memory appears to be the best solution, because it allows a simultaneous access to the memory from the ADC for data storage, and the CPU for data reading and processing.

A correct memory allocation is also extremely important when real-time operations are required and the observation period spans over several input samples. It has already been shown that, in this case, the measurement algorithm must be processed as soon as all samples belonging to one observation period have been acquired, and its execution must not stop the acquisition of the samples belonging to the next observation period.

The most convenient way to allocate the memory, in this case, is to organize it in a double circular buffer, as shown in Fig. 5.16.

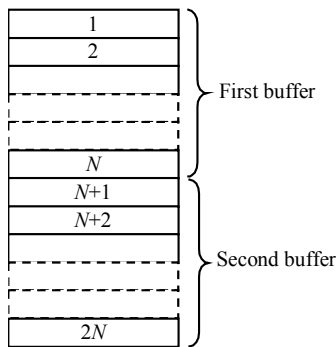


Figure 5.16. Double circular buffer memory allocation

Let us suppose that all data coming from the ADC are stored into the memory buffers in a DMA mode. Let us also suppose that each circular buffer contains  $N$  samples, and that the observation period contains  $N$  samples. Therefore, the first circular buffer is completely filled at the end of an observation period, and the stored data are available to be processed by the CPU. Under the assumption that data coming from the ADC can be stored in a DMA mode, all samples of the next observation period can be stored in the second buffer while the data in the first one are processed, without interrupting or slowing down the execution of the measurement algorithm. Once the second buffer has

been filled, its data can be processed and the new samples can be stored in the first buffer, and the whole process restarts.

If the time required to execute the measurement algorithm is shorter than the observation period, real-time operations can be ensured by this particular memory allocation.

## BANDWIDTH ESTIMATION

The synthetic description of the most important components of a DSP-based instrument and the information provided are sufficient to estimate the maximum sampling rate allowed by a given structure, and hence, according to the sampling theory explained in the previous chapters, allow to estimate the frequency bandwidth of the instrument.

This operation is the most critical one, when DSP-based instruments are considered, because the correct estimation of the maximum sampling rate of the employed structure prevents gross mistakes that may have dramatic consequences, as incurring in the aliasing error.

In order to see how the sampling rate can be estimated, it is convenient to analyze single-channel and multi-channel structures separately.

### Single-channel structures

According to the considerations in all previous paragraphs, the *single-channel* structure that is needed to acquire and process a single signal is schematically reported in Fig. 5.17.

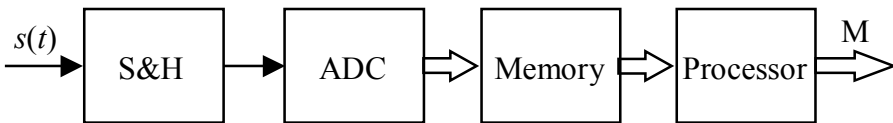


Figure 5.17. Single-channel structure

All blocks in the diagram of Fig. 5.17 introduce a delay in the data flow through the structure. In particular:

- Each sample of the input signal is available at the output terminals of the Sample and Hold (S&H) block at the end of the charging time  $t_{ch}$  of the hold capacitor. The value of  $t_{ch}$  is obtained by (5.22), once the charge time constant  $\tau_{ch}$  and the maximum allowed deviation  $\Delta U_s$  between the input value and the voltage on the capacitor are given.
- The digital code corresponding to the input sample value is provided by the ADC block with a time delay equal to the ADC conversion time  $t_c$ .
- This code is stored in the data memory in a time equal to the memory access time  $t_m$ .

Therefore, the total time required to acquire a sample and store it into the data memory is given by:

$$T_s = t_{ch} + t_c + t_m . \quad (5.33)$$

This time represents also the minimum sampling period for deferred operations. Therefore, the maximum sampling rate is given by:

$$f_s = \frac{1}{T_s} \quad (5.34)$$

and, according to the sampling theorem, the input signal bandwidth must be upper-bounded to the frequency value:

$$f_u = \frac{1}{2} f_s . \quad (5.35)$$

If real-time operations are required, the minimum sampling time should include the processing time  $t_p$  needed to execute the measurement algorithm strictly within the next observation period, as stated in the previous paragraphs. Therefore, (5.33) becomes:

$$T_{s_{rt}} = t_{ch} + t_c + t_m + t_p \quad (5.36)$$

## Multi-channel structures

Several measurement applications require to analyze and process more than one input signal. In this case, the structure shown in Fig. 5.17 must be suitably modified in order to cope with this requirement.

The most employed multi-channel structures are the so-called *non-multiplexed* and *multiplexed* structures.

### *The non-multiplexed structure*

The most immediate way to adapt the structure in Fig. 5.17 to acquire more than one signal is to duplicate all components involved with signal sampling and conversion. The resulting structure is shown in Fig. 5.18.

It can be readily checked that the sampling, conversion and storage operations are performed in parallel on all input channels. The converted codes are then sequentially read by the processor from each data memory bank. A slight reduction in complexity and cost can be attained by using only one memory bank, which is sequentially accessed by each ADC, though the whole conversion and storage process is slowed down.

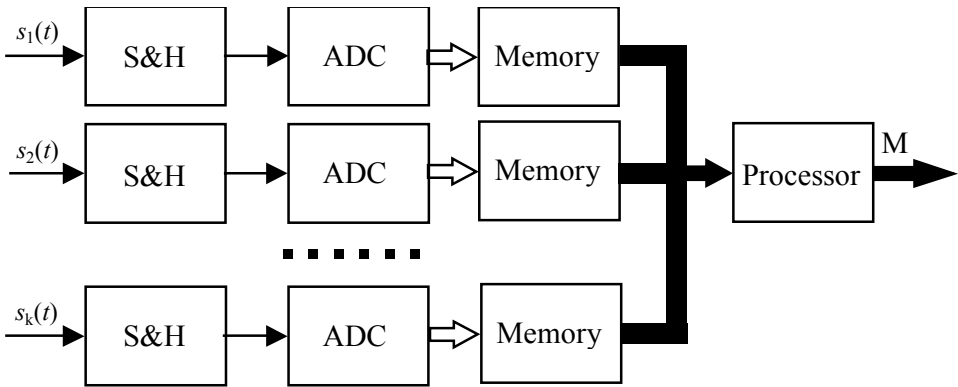


Figure 5.18. Multi-channel, non-multiplexed structure

With such a structure, the total time required to acquire one sample on all  $k$  channels and store them into memory is given again by (5.33) and is the same as the total time required to acquire one sample on one single channel. Therefore, the structure in Fig. 5.18 is the fastest one, since it features always the same sampling period, when deferred operations are considered, no matter on the number of input channels.

In case only one memory bank is employed, (5.33) is slightly modified as:

$$T_s' = t_{ch} + t_c + k \cdot t_m. \quad (5.37)$$

Of course, the maximum sampling rate and the upper limit for the input signal bandwidth are always given by (5.34) and (5.35) respectively, and the processing time  $t_p$  must be added to  $T_s$  when real-time operations are considered.

An important feature of this structure is that, if all Sample and Hold devices are controlled by the same signal, it allows to sample all input signals simultaneously, thus avoiding any phase error in the following measurement operations.

The major drawback of this structure is its cost, since the duplicated elements in each channel are generally the most expensive ones.

### The multiplexed structure

When the input signal frequency band is not too wide, so that fast sampling and conversion operations are not strictly required, the cost and complexity of the acquisition structure can be reduced by multiplexing the input signals to a single ADC, as shown in Fig. 5.19.

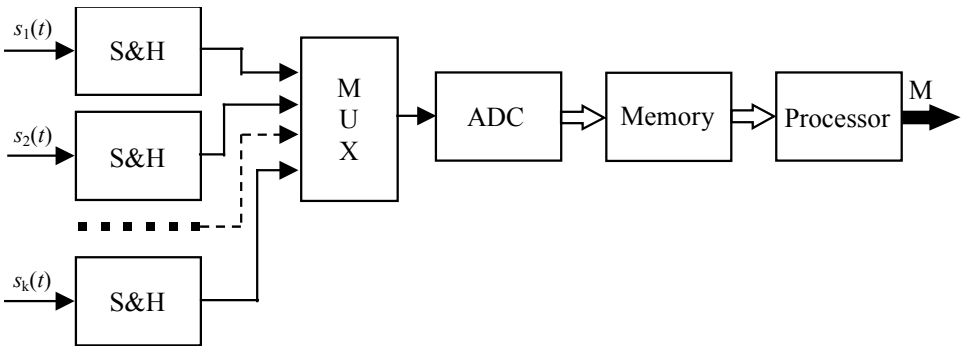


Figure 5.19. Multi-channel, multiplexed structure

It can be immediately recognized that all samples coming from the different input signals are sequentially brought to the ADC input by the multiplexer MUX, and are hence sequentially converted and stored into memory.

According to this structure, the total time required to acquire one sample on all  $k$  channels and store them into memory is given by:

$$T_s = t_{ch} + k \cdot (t_c + t_m). \quad (5.38)$$

The comparison between (5.33) and (5.38) shows in a quite clear way that this structure is much slower than that in Fig. 5.18, especially when the number of input channels increases, since the sampling period is proportional to the number  $k$  of the input channels.

The presence of a Sample and Hold device on each channel ensures that all input signals are sampled simultaneously, so that no phase error is caused. This is the reason for using a Sample and Hold device on each channel, and not only one inserted immediately before the ADC, as would be sufficient in order to avoid conversion errors in the presence of variable input signals.

The maximum sampling rate and the upper limit for the input signal bandwidth are given once again by (5.34) and (5.35) respectively. If real-time operations are required, the processing time  $t_p$  must be always added to  $T_s$  in (5.38).

The major benefit of using the multiplexed structure in Fig. 5.19 is its relatively low cost, due to the fact that only one ADC and one memory bank are employed. Though this structure is intrinsically slower than the non-multiplexed one, it is still effective in many industrial applications, where low-frequency signals must be processed and the number of input channels does not exceed 8 to 16 channels.

## ANTI-ALIASING METHODS AND SAMPLING STRATEGIES

### INTRODUCTION

Last Chapter 5 showed how to estimate the maximum sampling rate of an hardware acquisition structure, starting from its characteristic features. The sampling theorem, discussed in Chapter 3, proves that, in order to ensure a correct sampling, a continuous-time signal must be upper-bounded in the frequency domain, and relates the sampling rate to the maximum allowable frequency.

It may happen that the maximum sampling rate of a given structure is not high enough to satisfy the sampling theorem for the desired measurement application, and the increment of the sampling rate is not technically or economically viable. Furthermore, in many practical applications, the input signals cannot be considered as upper-bounded in the frequency domain because of the presence of noise. Therefore, suitable countermeasures must be taken in order to prevent aliasing errors.

The possible countermeasures depend on the nature of the signals to be sampled. According to their nature, it is possible to modify the signals themselves in order to adapt their bandwidth to the one required by the available sampling frequency, or adopt different sampling strategies. The first approach requires suitable hardware devices to be used as the input stage of a DSP-based instrument, while the second approach requires to modify the way in which the signals are sampled and the acquired samples are stored into the data memory.

The following sections will discuss the most widely employed approaches and will also show how the architecture of the data acquisition and conversion structure can be modified in order to increase the sampling rate, when the signal nature prevents from adopting any other less expensive countermeasure.

### THE ANTI-ALIASING FILTER

From a theoretical point of view, the simplest situation is the one given by a signal whose upper frequency is higher than half the sampling rate, but whose frequency-domain components located between half the sampling rate and the upper frequency are useless, or even harmful, in the measurement process. This is the typical case of the noise

components, that do not carry useful information to the measurement process, and can be hence neglected without affecting the measurement result.

In this case, the aliasing error can be avoided if the signal is filtered by an analog low-pass filter before sampling it. Supposing that an ideal filter is employed, and that its cut-off frequency is set at half the sampling rate  $f_s$ , the frequency band of the signal after the filter is graphically shown in Fig. 6.1.

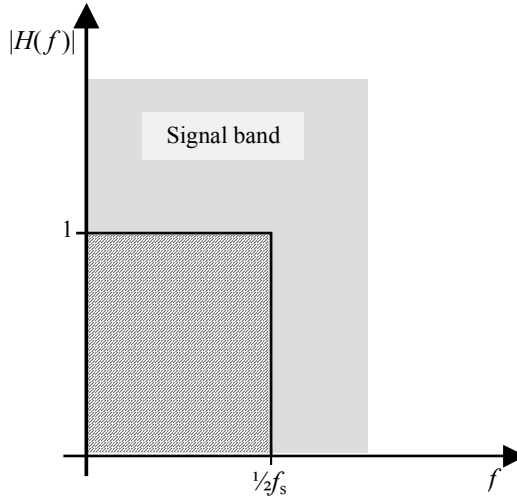


Figure 6.1. Ideal anti-aliasing filter effect. The light grey area represents the signal frequency band; the darker area represents the signal frequency band after it has been filtered

Of course, this is an ideal situation that will never occur, since all real filters are only an approximation of the ideal one, whose frequency response is shown in Fig. 6.1.

If a real low-pass filter is employed, with cut-off frequency equal to half the sampling rate, its frequency response differs from the ideal one, as shown in Fig. 6.2, where an example of the possible frequency response of a Butterworth low-pass filter is reported.

It can be noted that, as expected from the design specifications, the attenuation at the cut-off frequency is 3 dB, and that it is monotonically increasing both in the pass and stop bands. Moreover, a finite width transition band appears. These considerations lead to the following conclusions.

- Due to the presence of a finite width transition band, a number of signal components located at frequency values greater than half the sampling rate is not attenuated enough to avoid the aliasing error.
- The attenuation in the pass band is not zero over the whole band, but changes from 0 to 3 dB, thus modifying the signal components that are useful to the required measurement.

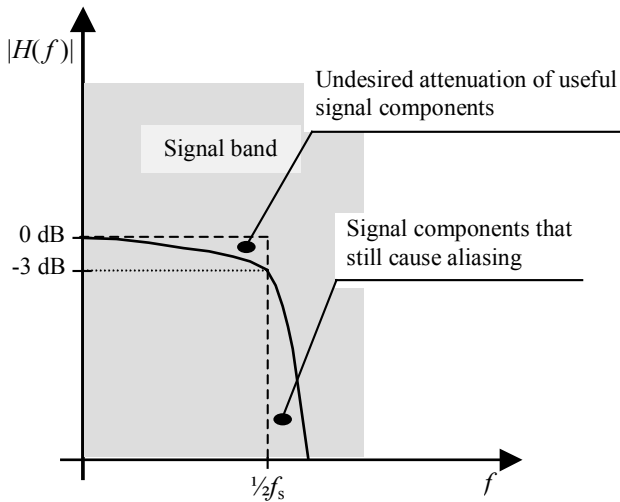


Figure 6.2. Impact of the frequency response of a real anti-aliasing filter on the signal

It can be immediately realized that the errors caused by the filter non-ideal behaviour can be mitigated if the cut-off frequency is modified. However, the following situations occur.

- If the filter cut-off frequency is decreased, all signal components with frequency values higher than half the sampling rate fall in the filter stop band, and no aliasing error will occur. However, this means that most of the filter transition band falls in the frequency range where the useful signal components are located. Therefore, the virtual absence of any aliasing error is paid by the attenuation of useful signal components, which results in a further reduction of the instrument bandwidth.
- If the filter cut-off frequency is increased, the error due to the attenuation of the useful signal components is mitigated, but a larger aliasing error may occur, since a larger number of signal components with frequency values higher than half the sampling rate is supposed to be not sufficiently attenuated.

According to all above considerations, it can be stated that the correct selection of the filter cut-off frequency and frequency response is a trade-off between these two conflicting needs. In order to find the optimal trade-off, it is then important to assess under which conditions the errors introduced by the anti-aliasing filters can be neglected.

The key point in this assessment is, once again, the resolution of the ADC that converts the samples of the input signal into digital codes. In fact, the ADC is unable to sense signal components whose amplitude is lower than its resolution, and is also unable to sense variations in the input signal that do not exceed its resolution.

As far as the aliasing error is concerned, it does not occur if the filter frequency response is such that the filter output is lower than the ADC resolution for all frequency



values higher than half the sampling rate. If an input signal is considered that spans over the whole ADC dynamic range  $-\frac{U_{fs}}{2}, +\frac{U_{fs}}{2}$ , the filter gain, for frequency values higher than half the sampling rate, must satisfy the following relationship:

$$G \cdot U_{fs} < \frac{U_{fs}}{2^n}, \quad (6.1)$$

where  $n$  is the number of bits of the employed ADC. This means that:

$$G < \frac{1}{2^n}, \quad (6.2)$$

which leads, for  $n = 12$ , to  $G < -72$  dB. Since the gain at the cut-off frequency is  $-3$  dB, the gain variation from the cut-off frequency and half the sampling rate must be  $-69$  dB. Therefore, if a 4<sup>th</sup> order low-pass filter is considered, featuring a  $-80$  dB/decade attenuation outside the pass band, the cut-off frequency must be set at about one decade before half the sampling rate in order to ensure that no aliasing errors occur.

Let us now consider the gain variation in the pass band. As stated, a gain variation is reflected in the ADC output code only if the variation in the filter output signal is greater than 1 LSB. Thus, a gain variation  $\Delta G$  can be neglected only if it is:

$$\Delta G \cdot U_{fs} < \frac{U_{fs}}{2^n}, \quad (6.3)$$

which leads to:

$$\Delta G < \frac{1}{2^n} \quad (6.4)$$

If the filter dc gain is 1, the minimum allowed gain that does not produce variations in the ADC output code is given by:

$$G_{\min} = 1 - \frac{1}{2^n}, \quad (6.5)$$

that, for a 12-bit ADC, defines a band limit of  $-2.12 \cdot 10^{-3}$  dB, much smaller than the  $-3$  dB conventional pass band.

Similar considerations can be also derived for the phase frequency response. In fact, an ideal filter features a linear phase response that does not introduce any phase distortion in

the filter output signal. On the other hand, a real filter shows a phase distortion in the output signal, since its phase delay is a function of frequency.

Once again, the effect of the phase distortion can be neglected if it produces variations in the output signal that are lower than the ADC resolution.

Let us consider a  $\Delta\tau(f)$  phase delay variation in the pass band at signal frequency  $f$ , and let us refer it to the reference phase delay  $\tau_0$  at frequency  $f_0$ :

$$\Delta\tau(f) = \tau(f) - \tau_0 \quad (6.6)$$

Let us consider a sinewave as the input signal, with peak-to-peak amplitude equal to the ADC full-scale value, and let us suppose that the filter phase delay is much smaller than the signal period. Under these assumptions, the maximum variation in the signal amplitude caused by a phase delay variation occurs at the zero crossing times, where the sine function can be well approximated by its argument.

Therefore, the phase delay variation can be neglected if it is:

$$\frac{U_{fs}}{2} \cdot 2\pi f \cdot \Delta\tau(f) < \frac{1}{2} \cdot \frac{U_{fs}}{2^n} \quad (6.7)$$

If period  $T = 1/f$  is considered, the maximum relative phase delay is obtained from (6.7) as:

$$\frac{\Delta\tau}{T} < \frac{1}{\pi \cdot 2^{n+1}} \quad (6.8)$$

If a 12-bit ADC is considered again, (6.8) provides a relative phase delay  $\frac{\Delta\tau}{T} = 3.89 \cdot 10^{-5}$ , that limits the maximum allowable phase shift in the pass band to 0.244 mrad.

The results provided by (6.5) and (6.8) show that the design of an anti-aliasing filter is a quite complex task and that its use causes a reduction of the useful bandwidth much greater than the theoretical one given by the ideal filter. The use of the anti-aliasing filter becomes more and more critical as the ADC resolution increases, as shown by (6.2), (6.5) and (6.8) and therefore a great attention must be paid to this component, if present, when discussing the accuracy of a DSP-based instrument.

## THE HETERODYNE CONVERSION

The *heterodyne conversion* is a frequently used method for shifting a signal from a frequency band to a different one. It is based on the mathematical property of the product of two sinewaves. In fact, given two sinewaves at different frequencies:

$$s_1(t) = S_1 \cdot \cos(2\pi f_1 t + \varphi_1); \quad s_0(t) = S_0 \cdot \cos(2\pi f_0 t + \varphi_0)$$

their product can be decomposed into the sum of two sinewaves, one at frequency  $f_1 - f_0$ , and the other one at frequency  $f_1 + f_0$ :

$$s_1(t) \cdot s_0(t) = \frac{S_1 S_0}{2} \cdot \cos[2\pi(f_1 - f_0)t + (\varphi_1 - \varphi_0)] + \frac{S_1 S_0}{2} \cdot \cos[2\pi(f_1 + f_0)t + (\varphi_1 + \varphi_0)] \quad (6.9)$$

If the superposition principle is applied, it can be readily checked that (6.9) can be still applied if one of the signals is no longer a sinewave, but has a finite bandwidth in the frequency domain, as shown in Fig. 6.3. In this case, the whole spectrum of the signal, after its multiplication by a sinewave, is decomposed into two equal replicas, the first one shifted towards the lower frequencies, and the second one shifted towards the higher frequencies, as shown in Fig. 6.4.

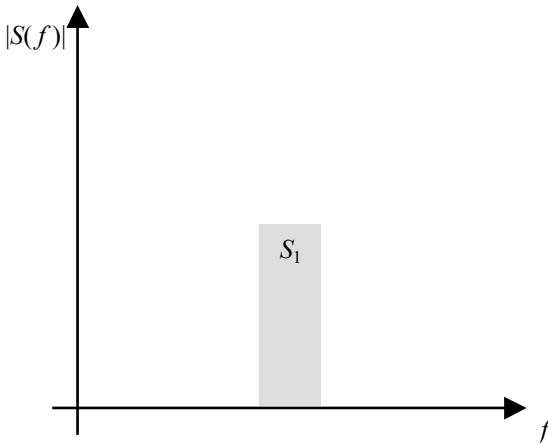


Figure 6.3. Typical finite bandwidth signal spectrum

It is then possible to discard one of the two replicas, by properly filtering the signal provided by the multiplier. If a high-pass filter is applied, the replica of the original signal shifted in frequency by  $+f_0$  is obtained. If a low-pass filter is applied, as shown in Fig. 6.4, the replica of the original signal shifted in frequency by  $-f_0$  is obtained.

If this last solution is adopted, it is possible to translate the whole signal to a frequency band lower than half the sampling rate, provided that the following conditions are satisfied:

- if  $f_{\min}$  is the lower bound and  $f_{\max}$  is the upper bound of the signal frequency band, so that the signal bandwidth is given by  $f_{\max} - f_{\min}$ , it must be:  $f_{\max} - f_{\min} < f_s/2$ ,  $f_s$  being the sampling rate;
- frequency  $f_0$  of the local oscillator in the heterodyne converter shown in Fig. 6.5 must be selected in such a way that  $f_{\max} - f_0 < f_s/2$ .

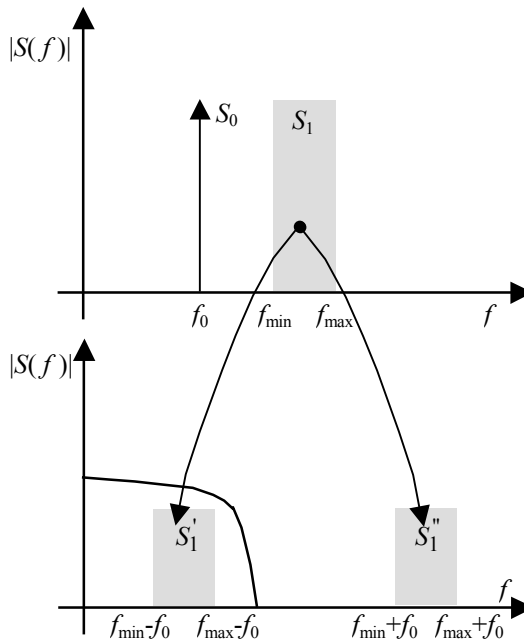


Figure 6.4. Frequency translation of the input signal  $s_1$  after multiplication by a sinewave with frequency  $f_0$ . The signal replica at the lower frequencies can be extracted by means of a low-pass filter

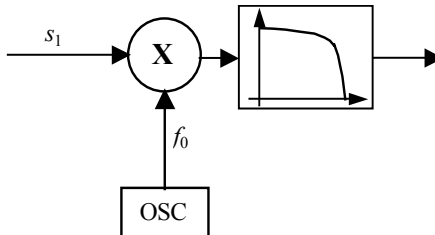


Figure 6.5. Block diagram of an heterodyne converter

The above described heterodyne conversion can be implemented by the architecture schematically shown in Fig. 6.5. The local oscillator OSC generates a sinewave with the desired frequency  $f_0$ , that multiplies the input signal  $s_1$  in multiplier X. The output signal of multiplier X is low-pass filtered, so that the low frequency replica of the input signal is obtained as the output signal of the heterodyne converter.

The most critical blocks in this structure are the multiplier and the local oscillator, since they both can deeply affect the measurement accuracy.

In particular, the local oscillator is required to generate an extremely pure sinewave, otherwise each frequency component of the input signal  $s_1$ , located at the generic frequency  $f$ , will be replicated to all frequencies  $f - kf_0$ ,  $k$  being the harmonic order of the frequency-domain components of the oscillator output signal. Of course, if the oscillator output signal is a pure sinewave, it shows only one frequency-domain component for  $k = 1$ , and the heterodyne converter provides an undistorted replica of the input signal. Otherwise, the output signal is a distorted replica of the input signal, and the greater is the distortion in the oscillator output, the greater is the distortion in the converter output.

As for the multiplier, it must feature a very high linearity, otherwise it will cause too some distortion on the output signal.

Once again, the maximum allowable error is given by the resolution of the ADC employed to convert the input signal: the higher is its resolution, the lower is the allowable error.

Presently, the heterodyne conversion is employed with good results in measurement applications, especially in the field of telecommunications and radar applications, where high frequency signals must be processed.

## THE EQUIVALENT-TIME SAMPLING TECHNIQUE

Periodic signals represent a very important class of signals, that can be met in several practical applications. The main property of a periodic signal  $s(t)$ , with period  $T$ , is that:

$$s(t) = s(t + nT), \quad n = 1, 2, \dots, \infty. \quad (6.10)$$

This means that the value taken by signal  $s$  at any generic sampling time  $kT_s$  is taken again at each new period:

$$s(kT_s) = s(kT_s + nT), \quad n = 1, 2, \dots, \infty. \quad (6.11)$$

Let us now suppose that signal  $s$  is upper-bounded, in the frequency domain, by harmonic  $N$ ; hence, according to the formulation of the sampling theorem reported in Chapter 3 for the periodic signals, at least  $2N + 1$  samples must be taken over period  $T$ .

However, it may happen that the hardware structure employed to acquire and process signal  $s$  is unable to provide the required sampling rate, and only a number of samples lower than  $2N + 1$  can be taken over period  $T$ .

Due to (6.11), this is not a problem, provided that a sufficient number of signal periods can be observed, since it is always possible to find the required samples in one of the successive periods of the input signal.

This situation is well depicted in Fig. 6.6, where 10 samples per period have to be acquired from a sinewave.

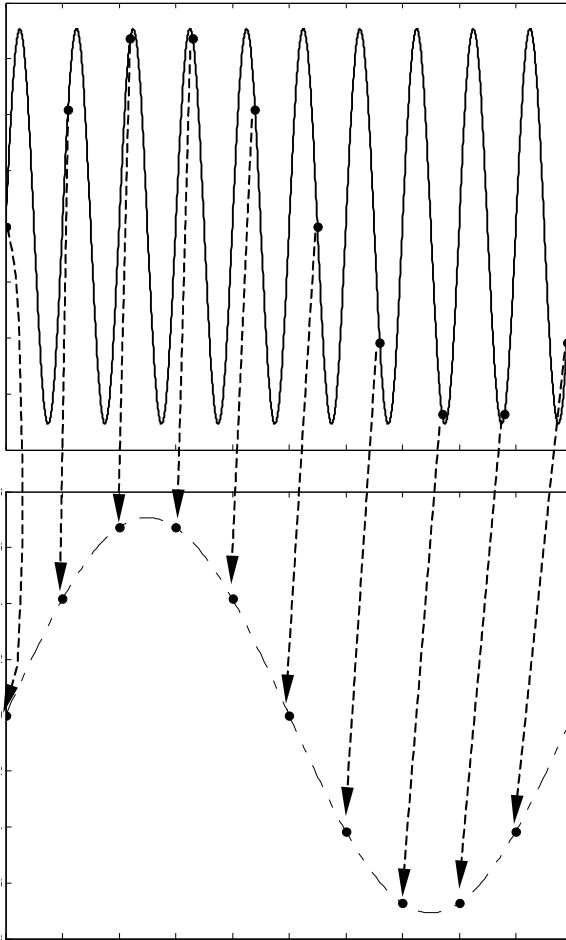


Figure 6.6. Equivalent-time sampling technique. The upper graph shows the input signal and the acquired samples (one per period). The lower graph shows how the acquired samples are ordered so that one period of the input signal can be reconstructed.

The input signal is observed for 10 periods, and only one sample is acquired from each period, as shown in the upper graph of Fig. 6.6. The actual sampling period is given by:

$$T_{s_a} = T + T_s,$$

where  $T$  is the input signal period, and  $T_s$  is the desired sampling period.

One period of this signal is then reconstructed, as shown in the lower graph of Fig. 6.6, by ordering the acquired samples in such a way that the time interval between two successive samples is equal to the desired sampling period  $T_s$ .

This sampling technique is called the *equivalent-time sampling* technique, because it provides a signal, in the discrete time domain, equivalent to the one obtained by sampling a continuous-time signal with sampling period  $T_s$ . It allows to push the equivalent sampling frequency to virtually every value, by acquiring one sample every  $M$  periods of the input signal, provided that at least  $M \cdot (2N+1)$  periods are observed, if the signal is supposed to be reconstructed with  $2N + 1$  samples per period.

This technique is based on the mathematical condition (6.10), and leads to acceptable results only if this theoretical condition is met in practice too. This requires that the input signal remains stable for the whole observation interval needed to acquire the required number of samples. Of course, this condition is barely satisfied when the required observation interval grows longer and longer, thus limiting the obtainable sampling rate from the virtually very high values to lower values, depending on the stability of the input signal. If the input signal does not remain stable during the observation interval, the reconstructed signal will be affected by an error which acts as an equivalent non-linearity error.

As for the hardware structure needed to implement the equivalent-time sampling technique, it basically requires the implementation of a trigger circuit that recognizes a specified threshold crossing condition, so that the beginning of a new signal period is detected and the next sampling time accordingly set. Trigger errors are reflected into the reconstructed signal as an equivalent linearity error again.

This technique is generally implemented in the digital storage oscilloscopes (DSO), in order to increment their bandwidth when the input signals are repetitive.

## MULTI CONVERTER STRUCTURES

All methods presented in the last sections can be usefully employed only if the input signals show some well defined characteristics: no useful components over half the sampling rate in order to apply the anti-aliasing filter, limited bandwidth in order to apply the heterodyne conversion, or periodicity and stability in order to apply the equivalent-time sampling technique.

It may happen that the input signals do not satisfy any of the above conditions, and none of the acquisition structures shown in Chapter 5 features a sampling rate high enough to sample the input signals correctly. In this case, it is still possible to increase the sampling rate by modifying the acquisition structure as shown in Fig. 6.7.

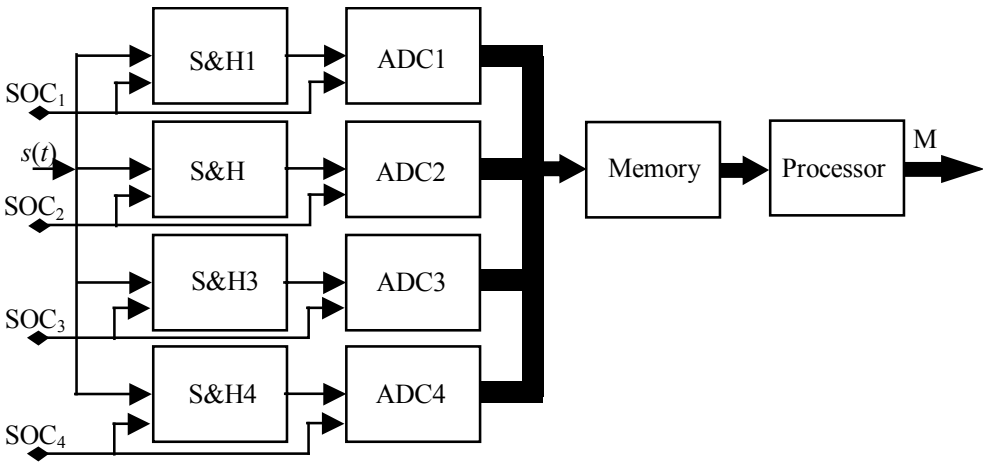


Figure 6.7. Multi-converter acquisition structure featuring  $K = 4$  converters

The same signal is fed to all  $K$  converters, that are controlled by different Start of Conversion (SOC) signals, shifted by a quantity equal to  $t_{\text{conv}}/K$ ,  $t_{\text{conv}}$  being the ADC conversion time. The time diagram of the control signals is shown in Fig. 6.8, for the same example with  $K = 4$  as the one in Fig. 6.7.

It can be immediately recognized that, after the first conversion has expired, the converted data coming from the converters are stored into the data memory at a rate as high as  $T_s = t_{\text{conv}}/K$ , thus providing an equivalent sampling rate equal to  $1/T_s$ .

This architecture allows, at present, the highest frequency rates, up to some gigahertz when fast converters are employed.

The capability of pushing the sampling rate at even higher values is limited by the number of ADCs that can be employed on each input channel. This limit is given by both technical and economical reasons.

The economical reason is the high cost that such structures have, due to the ADC cost and the complexity of the generator of the high-frequency control signals.

As far as the technical side of the problem is concerned, the multi-converter structure can be considered equivalent to a single, high-speed converter structure only if all  $K$  channels to which the input signal is fed feature the same performances and the same accuracy. If the behaviour of the sampling and conversion devices is different on the different channels, the discrete-time signal provided by this structure will appear to be affected by a strong equivalent non-linearity error.

For this reason, the multi-converter structures are presently limited to a very few converters for each input signal.



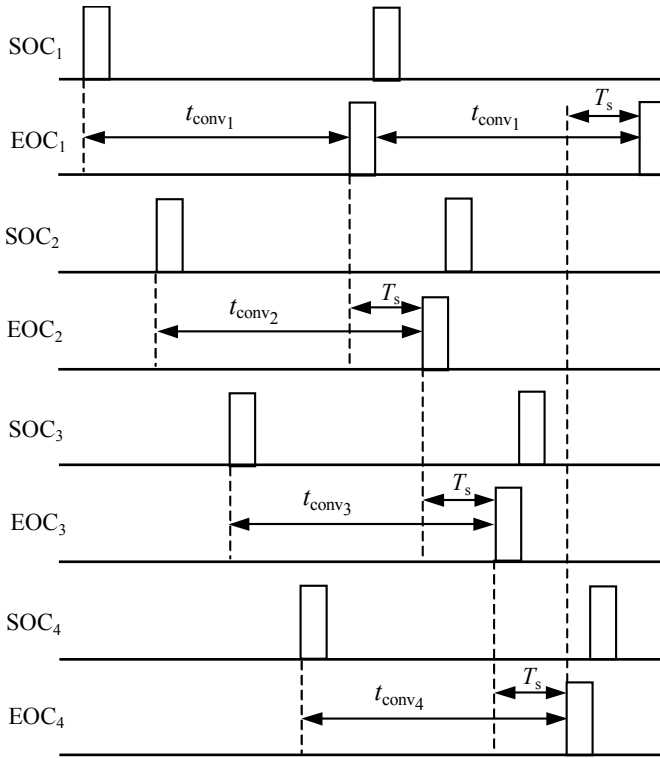


Figure 6.8. Time diagram of the control signals for the structure in Fig. 6.7

## CHAPTER 7

# DYNAMICAL DISCRETE SYSTEMS

## *LINEAR DIFFERENCE EQUATIONS AND z-TRANSFORM*

### INTRODUCTION.

In this chapter the concept of discrete linear shift invariant dynamical systems and their mathematical description in terms of  $n$ -domain and  $z$ -domain representations will be introduced, and the preliminary concepts proposed in Chapter 3 will be revisited more in depth.

### DYNAMICAL DISCRETE SYSTEMS.

A discrete system  $S$  establishes a functional relationship between the input sequence  $\{u\}$  and the output sequence  $\{y\}$  (see fig. 7.1).

If the system is linear the sequence  $a \cdot \{u_1\} + b \cdot \{u_2\}$  is transformed in the sequence  $a \cdot \{y_1\} + b \cdot \{y_2\}$ . If the system is shift invariant, for any integer value of  $n_0$ , the sequence  $\{u(n-n_0)\}$  is transformed in  $\{y(n-n_0)\}$ ; when the index  $n$  is associated to time the shift invariant system is called time invariant.

The mathematical expression of the functional relationship  $S$  between the input and output sequences represents the mathematical model of the discrete system. It can be derived observing the input and the output sequences of the discrete system  $S$  for any  $n \geq n_0$ .

Experimental evidence shows that in general the value  $y(n)$  is not univocally determined by the sequence  $\{u\}$  only: as a matter of fact applying the same input sequence to two identical systems the two output sequences can be different. This behavior is due to the “internal situation” which can be not the same for the two systems.

It is therefore necessary to introduce a set of  $q$  numbers (state variables)  $x_1(n_0), \dots, x_q(n_0)$  for defining the “internal situation” (state) of the system when  $n = n_0$ , in order to establish the analytical relationship between the input and the output sequences for any  $n \geq n_0$ :

$$y(n) = F[x_1(n_0), \dots, x_q(n_0); u(n_0), \dots, u(n); n], \quad \forall n \geq n_0 \quad (7.1)$$

The minimum value of  $q$  is the order of the discrete system.

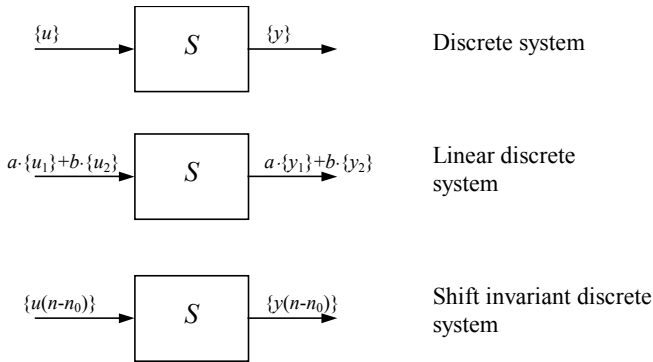


Figure 7.1. A discrete system establishes a relationship between an input and an output sequence

To illustrate this property of discrete systems let us consider the following simple example: the system  $S$  is an accumulator, i.e. a system summing up all the input values  $\{u(k)\}$ , for  $k$  from 0 up to  $n$ :

$$y(n) = \sum_{k=0}^n u(k)$$

Figure 7.2 shows the output sequences  $\{y_1\}$  and  $\{y_2\}$  for two identical accumulators with different state for  $n$  in the range  $0 \leq n \leq 10$ , when the input sequence is the unit impulse at  $n=3$ :  $\{u(n)\} = \{\delta(n-3)\}$ .

When it is possible to split equation (7.1) in the following two separate parts:

$$\begin{cases} x_1(n) = \phi_1[x_1(n_0), \dots, x_q(n_0); u(n_0), \dots, u(n-1); n] \\ \vdots \\ x_q(n) = \phi_q[x_1(n_0), \dots, x_q(n_0); u(n_0), \dots, u(n-1); n] \end{cases}, \quad \forall n > n_0 \quad (7.2a)$$

$$y(n) = g[x_1(n), \dots, x_q(n); u(n); n], \quad \forall n \geq n_0 \quad (7.2b)$$

the system is called “dynamical”.

Using (7.2a) it is possible to evaluate the state evolution of the system for any  $n$  once given the state of the system at  $n = n_0$  and the input sequence  $\{u\}$  in the range  $[n_0, n-1]$ , while the transformation (7.2b) allows the evaluation of the output for any  $n$  given the state and the input for the same value of the index  $n$ . It is important to point out the different nature of the two relationships (7.2a) and (7.2b); the former transformation for the evaluation of the state requires as arguments the whole sequence  $\{u\}$  in the range

$[n_0, n-1]$  together with the value of the state at  $n = n_0$ , while the latter is a simple algebraic relationship between the value of the output of the discrete system  $y(n)$  for a given index  $n$  and the values of the state  $x_1(n) \dots x_q(n)$  and the system input  $u(n)$  for the same value of the index  $n$ .

The equations in the system (7.2a) are called *state equations* while relationship (7.2b) is called *output transformation*.

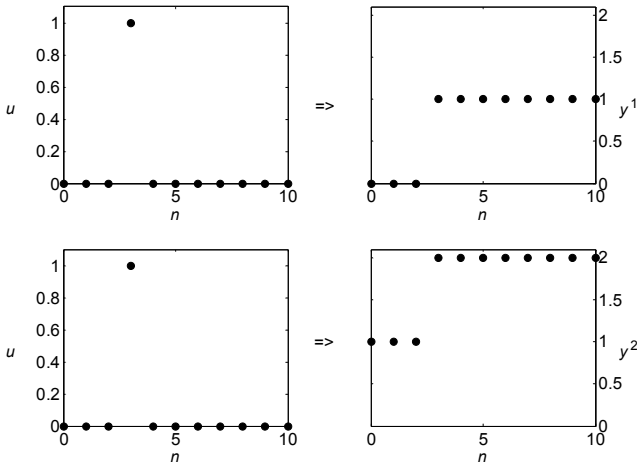


Figure 7.2. Accumulator response to the unit impulse at  $n=3$  for two different values of the accumulator internal state

***n*-DOMAIN REPRESENTATION OF DYNAMICAL DISCRETE SYSTEMS: DIFFERENCE EQUATIONS.**

We have already pointed out that the peculiarity of the state equations (7.2a) is that for evaluating the state for a certain value of the index  $n > n_0$  it is necessary to know the state at  $n = n_0$  and the input sequence  $\{u(k)\}$  for  $n_0 \leq k \leq n-1$ . It is trivial to verify that equations (7.2a) can be rewritten recursively considering the initial condition for  $n_0 = n-1$ , in the following format:

$$\begin{cases} x_1(n) = f_1[x_1(n-1), \dots, x_q(n-1); u(n-1); n] \\ \vdots \\ x_q(n) = f_q[x_1(n-1), \dots, x_q(n-1); u(n-1); n] \end{cases}, \quad \forall n > n_0 \quad (7.3)$$

where the state variables are calculated now from their previous values of the state variables  $x_1(n-1), \dots, x_q(n-1)$  and the input  $u(n-1)$ .

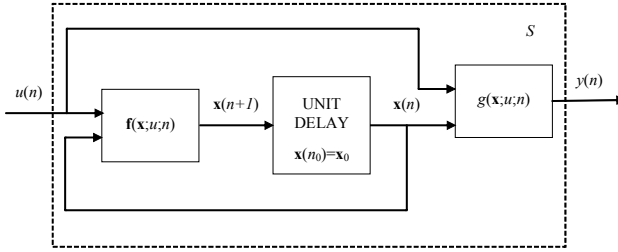


Figure 7.3. Block diagram representing in the  $n$ -domain a dynamical discrete system; this representation is valid  $\forall n \geq n_0$

The system of equations (7.3) is known in mathematics as a particular instance of a system of difference equations of order  $q$ . It represents the counterpart of the system of differential equations for describing continuous systems. The unknowns of this set of equations are not numbers, as in ordinary equations, but the sequences  $\{x_1\}, \dots, \{x_q\}$  (as for differential equations where the unknowns are functions).

Usually in practice it comes out to be more simple to write the state equations (7.3) and the output transformation (7.2b) in the following equivalent and more compact vector notation:

$$\mathbf{x}(n) = \mathbf{f}[\mathbf{x}(n-1); u(n-1); n], \forall n > n_0 \tag{7.4a}$$

$$y(n) = g[\mathbf{x}(n); u(n); n], \forall n \geq n_0 \tag{7.4b}$$

where  $\mathbf{x}(n)$  is the  $q \times 1$  state vector:

$$\mathbf{x}(n) = [x_1(n) \quad \dots \quad x_q(n)]^T$$

and  $\mathbf{f}[\cdot]$  is the following  $q \times 1$  vector of functions  $f_i(\cdot)$  of  $q+2$  variables:

$$\mathbf{f}[\cdot] = [f_1[\cdot] \quad \dots \quad f_q[\cdot]]^T$$

A dynamical system represented by the state equations (7.4a) and by the output transformation (7.4b) can be represented schematically as shown in figure 7.3.

Once defined a dynamical system, its *state-space* (or *phase-space*) is a mathematical  $q$ -dimensional space containing the system vector state  $\mathbf{x}$ . We can imagine the states of a dynamical system as points in its state-space. Assigned an initial state, as  $n$  varies the state of a dynamical system traces out a path in the state-space called a *trajectory*. The state *motion* is the state trajectory plotted as a function of  $n$  and can be represented in a  $q+1$  dimensional space.

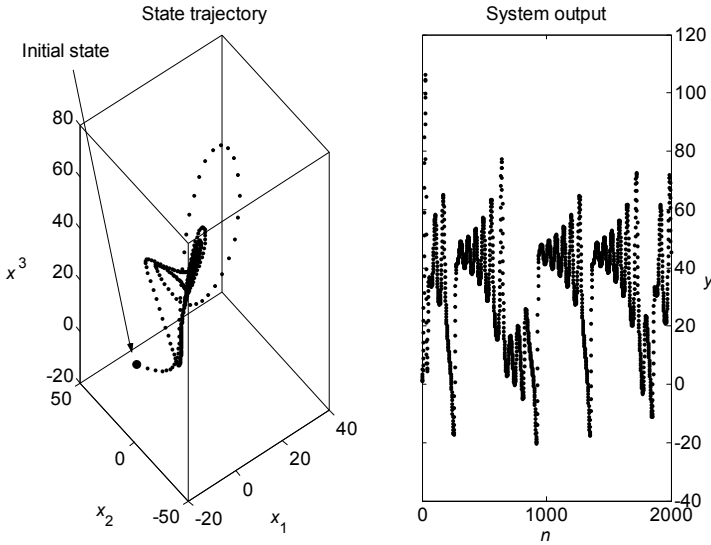


Figure 7.4. Example of trajectory for a 3<sup>rd</sup> order dynamical system into its 3 dimensional state-space together with the system output sequence

For example figure 7.4 shows the trajectory in the state-space for the following dynamical system state equations (Lorentz equations):

$$\begin{cases} x_1(n) = x_1(n-1) + 0.1 \cdot [x_2(n-1) - x_1(n-1)] \\ x_2(n) = x_2(n-1) + 0.01 \cdot [28 \cdot x_1(n-1) - x_2(n-1) - x_1(n-1)x_3(n-1)] \\ x_3(n) = x_3(n-1) + 0.01 \cdot [x_1(n-1)x_2(n-1) - 3 \cdot x_3(n-1)] \end{cases}$$

with initial state  $\mathbf{x}(0) = [-10 \ 20 \ -5]^T$ .

Figure 7.4 shows the output of this dynamical system as well resulting from the following output transformation:

$$y(n) = x_1(n) + x_2(n) + x_3(n)$$

Despite of the simple structure of the finite different equation (7.4a), it can describe complex behavior of dynamical systems. Let us consider for example the following state equation and output transformation:

$$\begin{cases} x_1(n) = x_1(n-1)\cos(\alpha) - [x_2(n-1) - x_1(n-1)^2]\sin(\alpha) \\ x_2(n) = x_1(n-1)\sin(\alpha) + [x_2(n-1) - x_1(n-1)^2]\cos(\alpha) \end{cases}$$

$$y(n) = x_1(n) + x_2(n)$$

with  $\alpha = 1.3$ . Figure 7.5 shows the trajectory associated to the initial condition:

$$x_1(0) = x_2(0) = 0.21$$

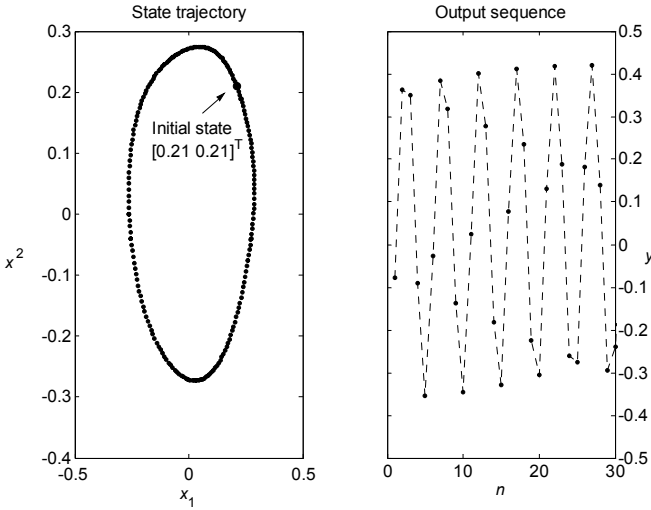


Figure 7.5. Example of trajectory and output sequence for a limit cycle

The states of this trajectory lay on a closed curve. This situation is defined a limit cycle, since the behavior of the state vector is periodic: when it reaches the initial conditions it repeats indefinitely the previous trajectory. This periodic behavior of the state vector is observable from cycle oscillations of the system output  $y$ , as shown in figure 7.5.

Figure 7.6 shows the trajectory for the same dynamical system subject to a different initial conditions:

$$x_1(0) = x_2(0) = 0.33$$

In this case simply changing the initial state conditions the trajectory degenerates in five limit cycles. This behavior can be seen also from the system output  $y$  shown in figure 7.6, where the five limit cycles are associated to five oscillation of the output sequence  $\{y\}$ .

If the discrete system is linear also the state equations (7.3) and the output transformation (7.2b) are linear:

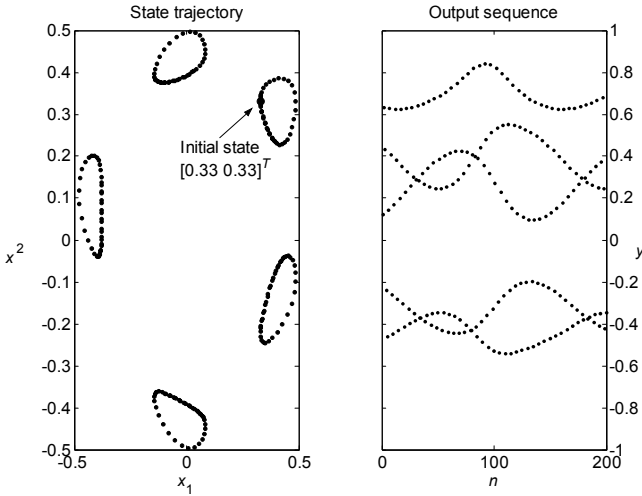


Figure 7.6. Degeneration of the limit cycle of figure 7.5 into five limit cycles due to a change in the initial state vector value

$$\begin{cases} x_1(n) = a_{11}(n) \cdot x_1(n-1) + \dots + a_{1q}(n) \cdot x_q(n-1) + b_1(n) \cdot u(n-1) \\ \vdots \\ x_q(n) = a_{q1}(n) \cdot x_1(n-1) + \dots + a_{qq}(n) \cdot x_q(n-1) + b_q(n) \cdot u(n-1) \end{cases}, \forall n > n_0 \quad (7.5a)$$

$$y(n) = c_1(n) \cdot x_1(n) + \dots + c_q(n) \cdot x_q(n) + d(n) \cdot u(n), \forall n \geq n_0 \quad (7.5b)$$

Also (7.5) can be rewritten in vector notation as:

$$\begin{aligned} \mathbf{x}(n) &= \mathbf{A}(n) \cdot \mathbf{x}(n-1) + \mathbf{b}(n) \cdot u(n-1), \forall n > n_0 \\ y(n) &= \mathbf{c}(n)^T \cdot \mathbf{x}(n) + d(n) \cdot u(n), \forall n \geq n_0 \end{aligned} \quad (7.6)$$

where  $\mathbf{A}$  is the following  $q \times q$  matrix:

$$\mathbf{A}(n) = \begin{bmatrix} a_{11}(n) & \dots & a_{1q}(n) \\ \vdots & \ddots & \vdots \\ a_{q1}(n) & \dots & a_{qq}(n) \end{bmatrix}$$

and  $\mathbf{b}$  and  $\mathbf{c}$  are the  $q \times 1$  vectors:



$$\mathbf{b}(n) = [b_1(n) \ \dots \ b_q(n)]^T, \quad \mathbf{c}(n) = [c_1(n) \ \dots \ c_q(n)]^T$$

If the system is shift invariant, the coefficients  $a_{ij}$ ,  $b_i$ ,  $c_i$  and  $d$  do not depend upon the index  $n$ .

Figure 7.7 shows the block diagram representation of a linear shift invariant system as given by equations (7.6).

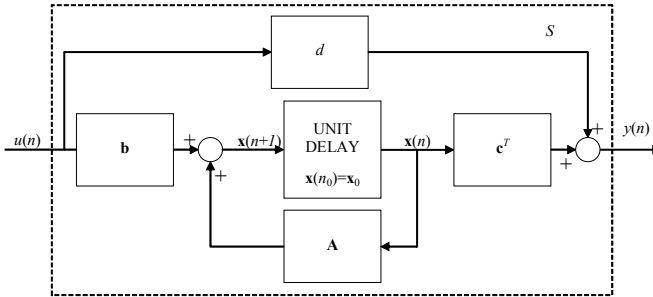


Figure 7.7. Block diagram representing in the  $n$ -domain a linear shift invariant dynamical discrete system, valid  $\forall n \geq n_0$

Using recursively the first equation in (7.6), for a shift invariant linear system we will obtain the vector state sequence as a function of the initial vector state and the input sequence:

$$\begin{aligned} \mathbf{x}(n_0) &= \mathbf{x}(n_0) \\ \mathbf{x}(n_0 + 1) &= \mathbf{A}\mathbf{x}(n_0) + \mathbf{b}u(n_0) \\ \mathbf{x}(n_0 + 2) &= \mathbf{A}\mathbf{x}(n_0 + 1) + \mathbf{b}u(n_0 + 1) = \mathbf{A}^2\mathbf{x}(n_0) + \mathbf{A}\mathbf{b}u(n_0) + \mathbf{b}u(n_0 + 1) \\ \mathbf{x}(n_0 + 3) &= \mathbf{A}^3\mathbf{x}(n_0) + \mathbf{A}^2\mathbf{b}u(n_0) + \mathbf{A}\mathbf{b}u(n_0 + 1) + \mathbf{b}u(n_0 + 2) \\ &\vdots \\ \mathbf{x}(n) &= \underbrace{\mathbf{A}^{n-n_0}\mathbf{x}(n_0)}_{ZIM} + \underbrace{\left[ \sum_{k=n_0}^{n-1} \mathbf{A}^{n-1-k} \mathbf{b}u(k) \right]}_{ZSM}, \quad \forall n > n_0 \end{aligned} \tag{7.7a}$$

From the last equation in (7.7a) we can see that the state vector motion is composed by two component: the first one is due to the initial state vector when the input sequence is zero (*ZIM* – Zero Input Motion). The second component is due to the input sequence when the initial state  $\mathbf{x}(n_0)$  is zero (*ZSM* – Zero State Motion).

The output sequence of the linear shift invariant system results replacing the last expression in (7.7a) into the second equation in (7.6):

$$y(n) = \mathbf{c}^T \mathbf{A}^{n-n_0} \mathbf{x}(n_0) + \left[ \sum_{k=n_0}^{n-1} \mathbf{c}^T \mathbf{A}^{n-1-k} \mathbf{b} u(k) \right] + d \cdot u(n), \forall n > n_0 \quad (7.7b)$$

If we consider now the output sequence  $h(n)$  of the system in response to the unit impulse  $\delta(n)$ , with zero initial state vector, applying (7.6) recursively it is expressed by:

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{0} & h(0) &= \mathbf{c}^T \mathbf{x}(0) + d\delta(0) = d \\ \mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) + \mathbf{b}\delta(0) = \mathbf{b} & h(1) &= \mathbf{c}^T \mathbf{x}(1) + d\delta(1) = \mathbf{c}^T \mathbf{b} \\ \mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) + \mathbf{b}\delta(1) = \mathbf{A}\mathbf{b} & h(2) &= \mathbf{c}^T \mathbf{x}(2) + d\delta(2) = \mathbf{c}^T \mathbf{A}\mathbf{b} \\ \vdots & & \vdots & \\ \mathbf{x}(n) &= \mathbf{A}^{n-1} \mathbf{b} & h(n) &= \mathbf{c}^T \mathbf{A}^{n-1} \mathbf{b} \end{aligned} \quad (7.8)$$

Comparing the expression of the impulse response  $h(n)$  with zero state vector in (7.8) with the output sequence  $y(n)$  in (7.7b), this last equation can be rewritten as:

$$y(n) = \underbrace{\mathbf{c}^T \mathbf{A}^{n-n_0} \mathbf{x}(n_0)}_{ZIR} + \underbrace{h(n) * u(n)}_{ZSR}, \forall n \geq n_0 \quad (7.9)$$

This last relationship states that also the output sequence of a linear shift invariant system can be obtained as the superposition of two terms. The first one on the right side of (7.9) is the system output with no input (*ZIR* - Zero Input Response), while the second term is the system output sequence in response to the input forcing sequence  $u(n)$  when the initial state vector is zero (*ZSR* – Zero State Response).

### STABILITY OF LINEAR SHIFT INVARIANT DISCRETE SYSTEMS.

We have seen that both the state motion and the output sequence of a linear shift invariant system can be split in two contributors: the response of the system to the initial state with zero input and the system response to the input sequence with zero initial state. Nevertheless usually we wish to deal with dynamical systems which response is driven by the input sequence only, independently on the value of the initial state vector which in most cases is either unknown or it is impossible to force to zero.

Such kind of systems are characterized by a *ZIM* part of the state movement with all its vector components converging to zero as  $n$  increases. In these cases, when  $n$  will be large enough, the *ZIM* component will be negligible in comparison with the *ZSM* and consequently the system output will depend on the input sequence only and not on the initial state.

In order to analyze which are the necessary conditions to have such system behavior, let us start considering the simplest linear shift invariant system, i.e. the system of order one. According to (7.7a) the system *ZIM* motion is expressed by:

$$x(n) = a^{n-n_0} x(n_0) \quad (7.10)$$

Figure 7.8 shows the state movements for the following three different values of  $|a|$ :

1.  $|a| = 1$ : the state of the system does not change with  $n$ ; in this case we call the system *stable*;
2.  $|a| < 1$ : the state of the system asymptotically converge to zero as  $n \rightarrow \infty$ ; in this case we call the system *asymptotically stable*;
3.  $|a| > 1$ : the state of the system asymptotically diverge as  $n \rightarrow \infty$ ; in this case we call the system *unstable*.

If the system is asymptotically stable, after a certain value of the index  $n$ , it will be possible to determine the state movement by the input forcing sequence  $u(n)$  only, since the system response to the initial state will be negligible. This will not be possible in case of an unstable system: after a certain value of the index  $n$  the state motion in response to the initial state will be predominant in respect to the system response to the input sequence. The situation of a stable system is in between the two examined before: the *ZIT* contribution to the system state motion will neither diverge nor converge to zero; it will just hold the same amplitude of the initial state.

This result can be extended to higher order systems. In this case the *ZIT* motion is given by:

$$\mathbf{x}(n) = \mathbf{A}^{n-n_0} \mathbf{x}(n_0) \quad (7.11)$$

If we consider the vector norm of the state it can be demonstrated that, according to the absolute value of the  $q$  eigenvalues<sup>3</sup>  $\lambda_i$  of the matrix  $\mathbf{A}$ , we can have the following system behavior:

1. If  $|\lambda_i| < 1$ , for every  $i = 1 \dots q$ , the norm of the system state vector asymptotically converge to zero as  $n \rightarrow \infty$ ; in this case we call the system *asymptotically stable*;
2. If it exists at least one value of  $i = 1 \dots q$  such that  $|\lambda_i| > 1$ , the norm of the system state vector asymptotically diverge as  $n \rightarrow \infty$ ; in this case we call the system *unstable*.

---

<sup>3</sup> The eigenvalues of the matrix  $\mathbf{A}$  are the roots of the characteristic polynomial:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0$$

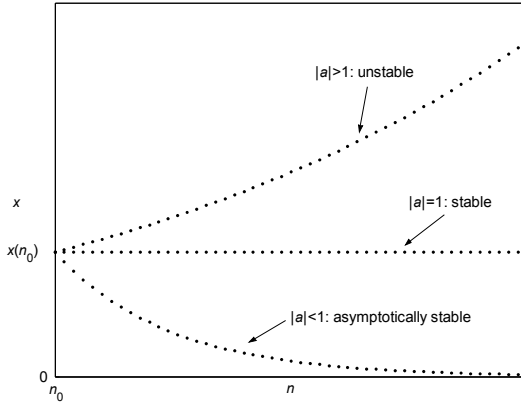


Figure 7.8. The zero input state motions for a unstable, stable and asymptotically stable linear shift invariant first order discrete system

**Example 1:** The system defined by the following state equation:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 1.05 & 0.2 \\ -0.5 & 1.05 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n-1)$$

is unstable because the two eigenvalues associated to the matrix  $\mathbf{A}$  are:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0 \Rightarrow \begin{cases} \lambda_1 = 1.050 + i0.316 \\ \lambda_2 = 1.050 - i0.316 \end{cases}$$

and both are in module greater than 1. Figure 7.9 shows both the unstable motion and trajectory for this system with initial state  $[1 \ 1]^T$ .

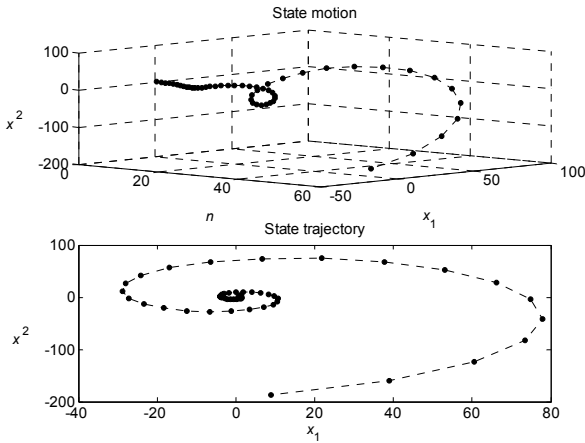


Figure 7.9. State motion and trajectory of the unstable 2<sup>nd</sup> order system of example 1

**Example 2:** The system defined by the following state equation:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ -0.5 & 0.9 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n-1)$$

is asymptotically stable because the two eigenvalues associated to the matrix **A** are:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0 \Rightarrow \begin{cases} \lambda_1 = 0.900 + i0.316 \\ \lambda_2 = 0.900 - i0.316 \end{cases}$$

and both are in module smaller than 1. Figure 7.10 shows both the motion and trajectory for this system with initial state  $[1 \ 1]^T$ .

The above *internal* stability definitions concerned with the system state variable behavior is not the only conceivable. Other kind of *external* stability concepts, concerned with the system input-output sequences behavior, are also of interest. In particular a linear shift invariant system is said to have bounded-input-bounded-output (*BIBO*) stability if every bounded input sequence produces a bounded output sequence for every initial state condition (regardless of the internal state vector norm behavior).

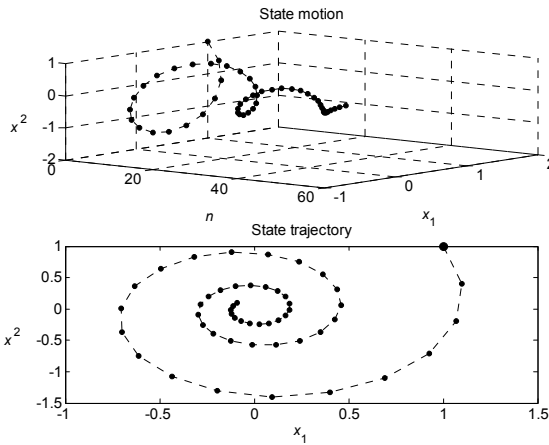


Figure 7.10. State motion and trajectory of the asymptotically stable 2<sup>nd</sup> order system of example 2

As shown in chapter 2 a system is *BIBO* stable if and only if the impulse response  $h(n)$  of the system is absolutely convergent, i.e. iff:

$$\sum_{-\infty}^{+\infty} |h(n)| < \infty \tag{7.12}$$

Condition (7.12) is sufficient because if the input sequence is bounded, exists a constant  $M$  such that  $|u(n)| \leq M < \infty$  for any  $n$  and, if (7.12) holds, the output is bounded either:

$$|y(n)| = \left| \sum_{k=-\infty}^{+\infty} h(k)u(n-k) \right| \leq M \sum_{k=-\infty}^{+\infty} |h(k)| < \infty \tag{7.13}$$

The necessity of the condition (7.12) can be demonstrated if it is possible to find at least one bounded input sequence that gives an unbounded output sequence when (7.12) does not hold. In chapter 2 the following bounded input sequence was considered:

$$u(n) = \begin{cases} +1 & \text{if } h(-n) \geq 0 \\ -1 & \text{if } h(-n) < 0 \end{cases} \tag{7.14}$$

Applying this bounded input sequence the output at  $n=0$  is not bounded:

$$y(0) = \sum_{k=-\infty}^{+\infty} h(k)u(-k) = \sum_{k=-\infty}^{+\infty} |h(k)| = \infty \tag{7.15}$$

The result (7.15) shows that the condition (7.12) is necessary as well.

Since some of the internal state variables may not contribute to determine the output of a linear system, *BIBO* stability does not imply asymptotic stability. Asymptotic stability is a strongest concept and can be demonstrated that it implies *BIBO* stability. When the word stable will be used without any further qualification in the text, it will mean asymptotical stability.

**Example 3:** The system defined by the following equation:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 1.05 & 0.2 \\ -0.5 & 1.05 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n-1), \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix}$$

$$y(n) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + 1 \cdot u(n)$$

is *BIBO* unstable, since its impulse response, shown in figure 7.11, is unbounded. The system of this example has the same state equation as the unstable system described in example 1. Figure 7.11 reports again its unstable zero input state trajectory.

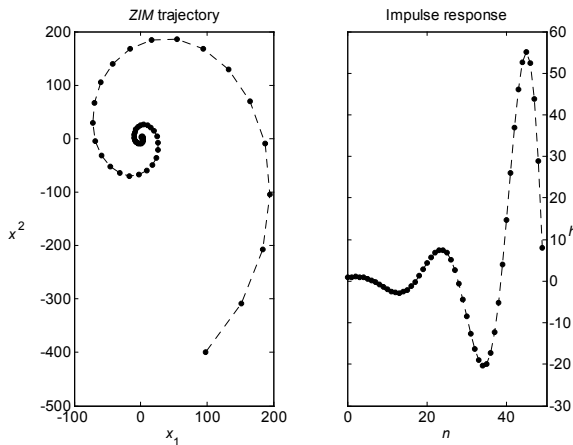


Figure 7.11. ZIM trajectory and impulse response of the unstable and *BIBO* unstable dynamical system of example 3

**Example 4:** The system defined by the following equation:

$$\begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 0.95 & 0.2 \\ 0 & 1.05 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n-1), \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix}$$

$$y(n) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + 1 \cdot u(n)$$

is *BIBO* stable, since its impulse response, shown in figure 7.12, is bounded. If we consider now the eigenvalues of the matrix  $\mathbf{A}$  they are:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0 \Rightarrow \begin{cases} \lambda_1 = 0.95 \\ \lambda_2 = 1.05 \end{cases}$$

Given that one eigenvalue is greater than 1 the system is unstable. Figure 7.12 reports the unstable zero input state trajectory for this system.

This example shows how the *BIBO* stability does not necessarily implies the system stability.

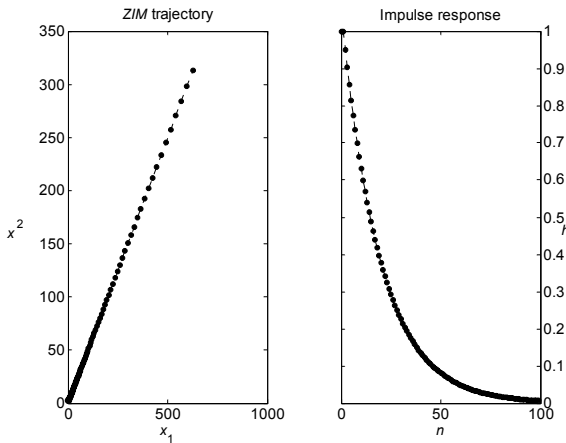


Figure 7.12. ZIM trajectory and impulse response of the unstable and *BIBO* stable dynamical system of example 4

THE *z*-TRANSFORM.

Let us consider now two infinite power series,  $P_1(\xi)$  and  $P_2(\xi)$ , in the complex variable  $\xi$ :



$$P_1(\xi) = \sum_{n=-\infty}^{+\infty} a_n \xi^n \quad (7.16)$$

$$P_2(\xi) = \sum_{n=-\infty}^{+\infty} b_n \xi^n \quad (7.17)$$

and their product  $P(\xi)$ :

$$P(\xi) = P_1(\xi) \cdot P_2(\xi) = \sum_{n=-\infty}^{+\infty} c_n \xi^n \quad (7.18)$$

Analyzing the relationship between the coefficients of  $P(\xi)$  and those of  $P_1(\xi)$  and  $P_2(\xi)$ , it follows:

$$c_n = \sum_{k=-\infty}^{+\infty} a_k b_{n-k} \quad (7.19)$$

Comparing (7.19) with the convolution product (2.22) it results that the coefficients of  $P$  are the convolution of the coefficients of  $P_1$  and  $P_2$ .

This property can be useful if we consider the power series coefficients as element of sequences. In this case we obtain the remarkable result of converting the complicated convolution product between sequences into the simpler ordinary product between complex functions.

We are ready now to introduce the  $z$ -transform  $X(z)$  of a sequence  $\{x(n)\}$ . It is defined as the power series (7.16) where the coefficients  $a_n$  are the elements  $x(n)$  of the sequence to be transformed, and the complex variable  $\xi$  is replaced by  $z^{-1}$ , i.e.:

$$X(z) = Z(\{x\}) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n} \quad (7.20)$$

**Example 1:** given the sequence  $\{x(n)\} = \{\delta(n+1) + \delta(n-1)\}$  its  $z$ -transform is the complex function:

$$X(z) = z + z^{-1} = \frac{z^2 + 1}{z}$$

Figure 7.13 shows the module of the function  $X(z)$  in the  $z$ -plane.

Since the power series (7.20), and therefore the  $z$ -transform, is a special form of a Laurent series<sup>4</sup>, its region of convergence (ROC)  $D$  is always composed by the open annular region in the complex plane  $r < |z| < R$  centered on  $z=0$ . The values  $r$  and  $R$  are the minor and major radius of convergence, respectively. In  $D$  the power series (7.20) converges uniformly (and absolutely), and consequently in  $D$  the function  $X(z)$  is an analytic function of the complex variable  $z$ . Outside  $D$  the power series (4.20) does not converge; on the border  $\partial D$  of  $D$  (7.20) may or may not converge, but at least in one point on  $\partial D$  there will be a singular<sup>5</sup> point for  $X(z)$ .

It is very important to point out that there may be different sequences  $\{x(n)\}$  in (7.20) converging to the same  $z$ -transform  $X(z)$ , but the convergence will be in different ROCs.

**Example 2:** The analytic function:

$$X(z) = \frac{2z + 1}{z^2 + z - 2}$$

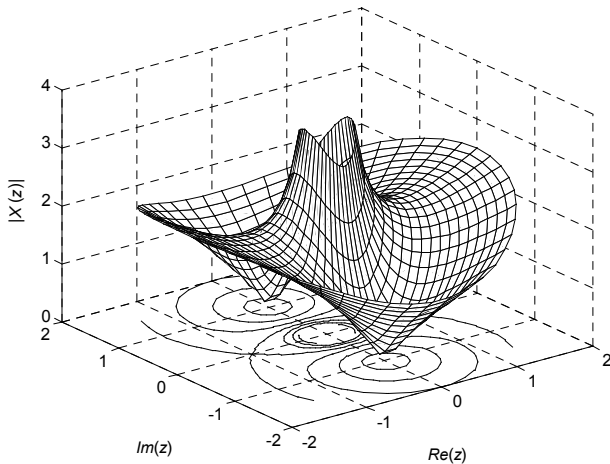


Figure 7.13. Module of the  $z$ -transform of the sequence  $\{\delta(n+1) + \delta(n-1)\}$

<sup>4</sup> A Laurent series of an analytic function  $f(z)$  is the power series:

$$f(z) = \sum_{n=-\infty}^{+\infty} c_n (z - z_0)^{-n}$$

<sup>5</sup> A singular point for an analytic function  $f(z)$  is a point  $z_0$  where does not exist the derivative of  $f(z)$ .

has two singular points in  $z_1=-2$  and  $z_2=1$ . As shown in figure 7.14 there are three annular ROCs in the complex plane:

- the interior of the disc  $D_1 \equiv |z| < 1$ ;
- the annular region  $D_2 \equiv 1 < |z| < 2$ ;
- the external portion of the circle  $|z|=2$   $D_3 \equiv |z| > 2$ .

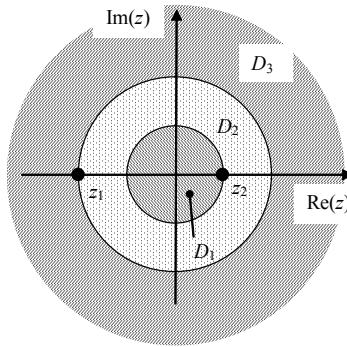


Figure 7.14. The three ROCs  $D_1$ ,  $D_2$  and  $D_3$  of three sequences  $\{x_1\}$ ,  $\{x_2\}$  and  $\{x_3\}$  with the same  $z$ -transform  $X(z)$  in the example 2

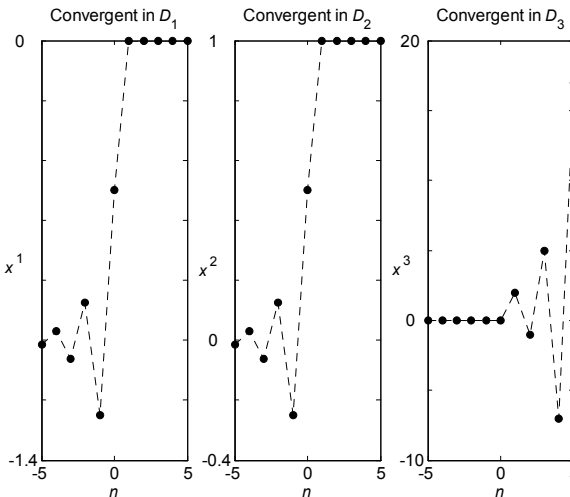


Figure 7.15. Plot of the three sequences  $\{x_1\}$ ,  $\{x_2\}$  and  $\{x_3\}$  convergent to the same  $z$ -transform  $X(z)$  of example 2

It can be demonstrated that the following three different power series, associated to three different sequences  $\{x_1\}$ ,  $\{x_2\}$  and  $\{x_3\}$ , are convergent to the same  $X(z)$  in each  $ROC D_1, D_2$  and  $D_3$ :

$$X(z) = \sum_{n=-\infty}^{+\infty} x_1(n)z^{-n} = \sum_{n=-\infty}^0 -[1+(-2)^{n-1}]z^{-n}, \text{ for each } z \text{ in } D_1$$

$$X(z) = \sum_{n=-\infty}^{+\infty} x_2(n)z^{-n} = \sum_{n=1}^{+\infty} z^{-n} - \sum_{n=-\infty}^0 (-2)^{n-1} z^{-n}, \text{ for each } z \text{ in } D_2$$

$$X(z) = \sum_{n=-\infty}^{+\infty} x_3(n)z^{-n} = \sum_{n=1}^{+\infty} [1+(-2)^{n-1}]z^{-n}, \text{ for each } z \text{ in } D_3$$

The three sequences  $\{x_1\}$ ,  $\{x_2\}$  and  $\{x_3\}$  are shown in figure 7.15.

From the previous example it is possible to see that the sequences  $\{x_1\}$ ,  $\{x_2\}$  and  $\{x_3\}$  have the following properties relating their respective  $ROC$  with the type of sequence:

- $\{x_1\}$ , convergent for  $|z|<1$ , is a sequence with  $x(n)\neq 0$  for  $n\leq 0$ ;
- $\{x_2\}$ , convergent for  $1<|z|<2$ , is a sequence with  $x(n)\neq 0$  for  $-\infty<n<+\infty$ .
- $\{x_3\}$ , convergent for  $|z|>2$ , is a sequence with  $x(n)\neq 0$  for  $n>0$ ;

More generally it is possible to classify sequences in four groups, each of them having a different  $ROC$ :

**1. Finite length sequence:** it is a sequence  $\{x\}$  with  $x(n)\neq 0$  only for  $n_1\leq n\leq n_2$ . For this type of sequences the  $ROC$  is the whole complex plane. If  $n_2\leq 0$  there are no coefficients of  $z^{-n}$  in the  $z$ -transform with  $n>0$ , and the  $ROC$  will include the point  $z=0$ . If  $n_1\geq 0$  in the  $z$ -transform there will be only terms  $z^{-n}$  with  $n>0$  and in this case the  $ROC$  will include the point  $z=\infty$ .

**2. Right monolateral sequence:** it is a sequence  $\{x\}$  with  $x(n)\neq 0$  only for  $n\geq n_0$ . If  $n_0<0$ , the  $z$ -transform of this type of sequence is composed by two terms:

$$X(z) = \underbrace{\sum_{n=n_0}^{-1} x(n)z^{-n}}_{\text{finite length}} + \sum_{n=0}^{+\infty} x(n)z^{-n} \tag{7.21}$$

The first finite length term converge everywhere but at  $z=\infty$ . To see what is the  $ROC$  for the second term let us suppose that it is convergent for a certain  $z=z_0$ . For every  $|z|>|z_0|$  the absolute value of each term in the infinite power series in (7.21) will be smaller than for  $z=z_0$ . As a consequence the second term will be absolutely (and

uniformly) convergent for  $|z|>r$ , with  $r$  equal to the minimum value of  $|z_0|$ . The *ROC* of (7.21), being the intersection of the *ROCs* of the two power series in (7.21), will include all the complex plane  $|z|>r$ , without the point  $z=\infty$ . If  $n_0\geq 0$  the *ROC* will include this point.

**3. Left monolateral sequence:** it is a sequence  $\{x\}$  with  $x(n)\neq 0$  only for  $n\leq n_0$ . If  $n_0>0$ , the  $z$ -transform of this type of sequence is composed by two terms:

$$X(z) = \sum_{n=-\infty}^0 x(n)z^{-n} + \underbrace{\sum_{n=1}^{n_0} x(n)z^{-n}}_{\text{finite length}} \tag{7.22}$$

The second finite length term converge everywhere but at  $z=0$ . To see what is the *ROC* for the first term let us suppose that it is convergent for a certain  $z=z_0$ . For every  $|z|<|z_0|$  the absolute value of each term in the infinite power series in (7.22) will be smaller than for  $z=z_0$ . As a consequence the second term will be absolutely (and uniformly) convergent for  $|z|<R$ , with  $R$  equal to the maximum value of  $|z_0|$ . The *ROC* of (7.22), being the intersection of the *ROCs* of the two power series in (7.22), will include all the complex plane  $|z|<R$ , without the point  $z=0$ . If  $n_0\leq 0$  the *ROC* will include this point.

**4. Bilateral sequence:** it is a sequence  $\{x\}$  with  $x(n)\neq 0$  for  $-\infty\leq n\leq +\infty$ . The  $z$ -transform of this type of sequence is composed by two terms:

$$X(z) = \underbrace{\sum_{n=-\infty}^{-1} x(n)z^{-n}}_{\text{left monolateral}} + \underbrace{\sum_{n=0}^{+\infty} x(n)z^{-n}}_{\text{right monolateral}} \tag{7.23}$$

The two power series in (7.23) are convergent for any  $|z|<R$  and any  $|z|>r$  respectively. If  $R>r$  for this type of sequence the *ROC* of the  $z$ -transform will be the annular ring  $r<|z|<R$ . If it results  $R<r$  the power series (7.23) will be not convergent and there will be not  $z$ -transform for the bilateral sequence.

The computation of the  $z$ -transform from an arbitrary sequence  $\{x(n)\}$  can be quite challenging. In table 7.1 the  $z$ -transforms and their *ROCs* are cataloged for some important elementary sequences.

In digital signal processing very often we have to deal with  $z$ -transforms  $X(z)$  which are rational functions, i.e. they results from the ratio between two polynomial in  $z$ . The roots of the numerator polynomial, where  $X(z)=0$ , are called the *zeros* of the function  $X(z)$ . The roots of the denominator polynomial, which are singular values for  $X(z)$ , are called the *poles* of  $X(z)$ . The poles lay in the  $z$ -plane on the border  $\partial D$  of the *ROC* for the  $z$ -transform. For the graphic representation of the  $z$ -transform of a sequence often it is very useful to plot in the  $z$ -plane the location of the zeros, with the symbol ‘o’, and of the poles, with the symbol ‘x’. This plot is called pole-zero diagram.

Table 7.1. *z*-transforms of elementary functions

Sequence $\{x(n)\}$	<i>z</i> -transform $X(z)$	ROC
$\delta(n)$	1	All $z$
unit step $I(n)$	$\frac{z}{z-1}$	$ z >1$
$n^t I(n)$	$\frac{z(z+1)^{t-1}}{(z-1)^{t+1}}$	$ z >1$
$a^n I(n)$	$\frac{z}{z-a}$	$ z > a $
$n^t a^n I(n)$	$\frac{az(z+a)^{t-1}}{(z-a)^{t+1}}$	$ z > a $
$a^n \sin(\omega_0 n) I(n)$	$\frac{az^{-1} \sin(\omega_0)}{1-2az^{-1} \cos(\omega_0) + a^2 z^{-2}}$	$ z > a $
$a^n \cos(\omega_0 n) I(n)$	$\frac{1-az^{-1} \cos(\omega_0)}{1-2az^{-1} \cos(\omega_0) + a^2 z^{-2}}$	$ z > a $
$a^n (I[n]-I[n-n_0])$	$\frac{1-a^{n_0} z^{-n_0}}{1-az^{-1}}$	All $z$ , except $z=0$ if $n_0>1$ and $z=\infty$ if $n_0<1$

It is very important to underline the close relationship between the *z*-transform and the Fourier transform of a sequence. Writing the complex variable *z* in polar form as:

$$z = \rho e^{j\omega} \tag{7.24}$$

the definition (7.20) of the *z* transformation can be rewritten as:

$$X(z) = Z(x) = \sum_{n=-\infty}^{+\infty} x(n) \rho^{-n} e^{-j\omega n} \tag{7.25}$$

If the ROC of the *z*-transform includes the circle with radius one, on this circle where  $|z|=\rho=1$  (7.25) represents the Fourier transform (2.44) of the sequence  $x(n)$ ; thus the Fourier transform of the sequence can be evaluated by its *z*-transform (7.20) substituting the complex variable *z* with  $e^{j\omega}$ .

$$F(x) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n} = X(e^{j\omega}) \tag{7.26}$$

The periodicity of the Fourier transform of a sequence is now obvious from this geometric interpretation. Figure 7.16 shows the values of the module of the  $z$ -transform of the example 1 evaluated on the circle of radius one in the  $z$ -plane.

Since  $X(e^{j\omega})$  is a complex function of the real argument  $\omega$ , its amplitude and phase can be plotted as function of the angular frequency  $\omega$  as in figure 7.17 where the values of the amplitude and phase of the complex function  $X(z)$  assumed along the unit circle are plotted as function of the phase of the complex number  $z=e^{j\omega}$  (i.e. the angular frequency  $\omega$ ). This function represent the Fourier transform of the original sequence  $X(e^{j\omega})$ .

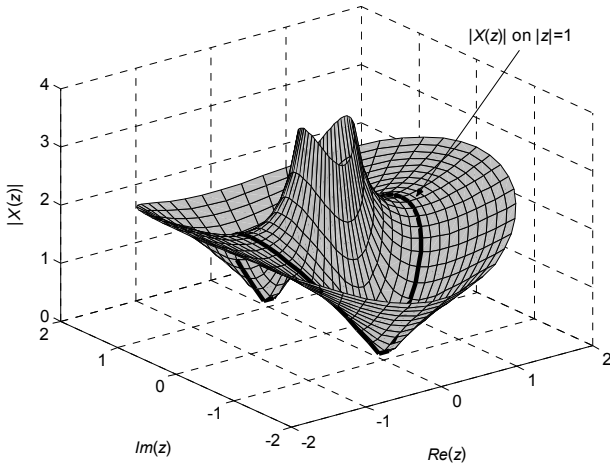


Figure 7.16. The magnitude of the Fourier transform of a sequence is the absolute value of the  $z$ -transform evaluated on the unitary circle in the  $z$ -plane

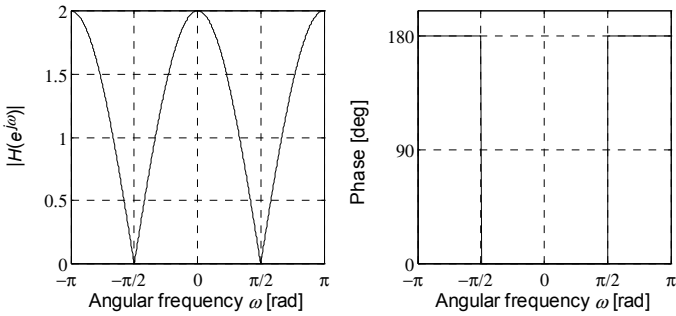


Figure 7.17. The amplitude and phase of the  $z$ -transform computed along the unit circle in the  $z$ -plane, representing the Fourier transform of the sequence of example 1

Table 7.2. Properties of z-transforms

Property	Sequence	z-transform $X(z)$	ROC
Notation	$x(n)$	$X(z)$	$r< z <R$ (ROC)
	$x_1(n)$	$X_1(z)$	$r_1< z <R_1$ (ROC <sub>1</sub> )
	$x_2(n)$	$X_2(z)$	$r_2< z <R_2$ (ROC <sub>2</sub> )
Homogeneity (Scaling)	$\alpha x(n)$	$\alpha X(z)$	ROC
Additivity	$x_1(n)+x_2(n)$	$X_1(z)+X_2(z)$	At least $ROC_1 \cap ROC_2$
Linearity	$\alpha x_1(n)+\beta x_2(n)$	$\alpha X_1(z)+\beta X_2(z)$	At least $ROC_1 \cap ROC_2$
Shift (Delay)	$x(n-n_0)$	$z^{-n_0} X(z)$	ROC, except $z=0$ if $n_0>0$ and $z=\infty$ if $n_0<0$
Reversal	$x(-n)$	$X(1/z)$	$1/R< z <1/r$
Scaling in the z domain	$a^n x(n)$	$X(a^{-1}z)$	$ a r< z < a R$
Complex modulation	$e^{j\theta n} x(n)$	$X(e^{-j\theta}z)$	ROC
Complex conjugate	$x^*(n)$	$X^*(z^*)$	ROC
Differentiation in the z domain (Ramping)	$nx(n)$	$-z \frac{dX(z)}{dz}$	ROC
Reciprocal Decay	$\frac{1}{n} x[n]$	$-\int \frac{X(z)}{z} dz$	ROC
Periodic Extension of Signal of Length N	$\sum_{n=0}^{\infty} x(n+kN)$	$\frac{z^N X(z)}{z^N - 1}$	
Convolution	$x_1(n)*x_2(n)$	$X_1(z) \cdot X_2(z)$	At least $ROC_1 \cap ROC_2$
Multiplication	$x_1(n) \cdot x_2(n)$	$\frac{1}{2\pi j} \oint X_1(\xi) X_2\left(\frac{z}{\xi}\right) \frac{d\xi}{\xi}$	At least $r_1 r_2 <  z  < R_1 R_2$

The z-transform  $Z(\cdot)$  has few remarkable properties, similar to those of the Laplace transform, and summarized in table 7.2. Here some comments are given about how they have been obtained.

**Linearity:** it derives directly from the z-transform definition. The ROC of the linear combination of two sequences  $\{x_1\}$  and  $\{x_2\}$  is *at least* the intersection of their ROCs. It can be wider because some zeros of the combination may cancel out some poles. This happens for example when a finite length sequence, converging everywhere but in  $z=0$  or in  $z=\infty$ , results from the combination of two infinite length sequences. This is the case for the finite length sequence  $\{a^n I(n)-a^n I(n-2)\}$  (see table 1) converging everywhere but at  $z=0$ , while the two sequences  $\{a^n I(n)\}$  and  $\{a^n I(n-2)\}$  are convergent only for  $|z|>|a|$ .



**Shift:** the  $z$ -transform of a shifted (delayed) sequence by  $n_0$  samples is given by

$$Z[x(n - n_0)] = \sum_{k=-\infty}^{+\infty} x(k - n_0)z^{-k} = \sum_{j=-\infty}^{+\infty} x(j)z^{-(j+n_0)} = z^{-n_0} Z[x(n)] \quad (7.22)$$

The  $z^{-n_0}$  term in (7.22) introduces poles in  $z=0$  if  $n_0>0$  and in  $z=\infty$  if  $n_0<0$ .

**Reversal:** reversing a sequence results in the following  $z$ -transform

$$Z[x(-n)] = \sum_{k=-\infty}^{+\infty} x(-k)z^{-k} = \sum_{j=-\infty}^{+\infty} x(j)\left(\frac{1}{z}\right)^{-j} = X\left(\frac{1}{z}\right) \quad (7.23)$$

**Scaling:** multiplying a sequence  $\{x\}$  with an exponential sequence  $\{a^n\}$  results in the following  $z$ -transform:

$$Z[a^n x(n)] = \sum_{k=-\infty}^{+\infty} x(k)a^k z^{-k} = \sum_{k=-\infty}^{+\infty} x(k)\left(\frac{z}{a}\right)^{-k} = X\left(\frac{z}{a}\right) \quad (7.24)$$

If  $a$  is real, with  $0 < a < 1$ , (7.24) is a compression in the  $z$ -plane of the  $z$ -transform of  $x(n)$ . If  $a$  is real with  $a > 1$  (7.24) is instead an expansion of  $X(z)$ . If  $a$  is complex with  $|a|=1$ , i.e.  $a=e^{j\theta}$ , (7.24) produces a rotation for  $X(z)$  of an angle  $\theta$  around the origin in the  $z$ -plane (modulation). Figure 7.18a shows the effect of different scaling on the  $z$ -transform of the sequence  $\{a^n[\delta(n+1)+\delta(n-1)]\}$ .

On the pole-zero location the scaling has the effect to move them along radial cords in the complex plane (see figure 7.18b). The complex modulation instead rotate the position of poles and zeros in the  $z$ -plane (see figure 7.18b).

**Conjugation:** The  $z$ -transform of the complex conjugate is given by:

$$Z(x^*) = \sum_{k=-\infty}^{+\infty} x^*(k)z^{-k} = \left[ \sum_{k=-\infty}^{+\infty} x(k)(z^*)^{-k} \right]^* = X^*(z^*)$$

**Differentiation:** it comes out directly from the derivative of the  $k^{th}$  term  $x(k)z^{-k}$  of the  $z$ -transform of  $\{x(n)\}$  :

$$\frac{d[x(k)z^{-k}]}{dz} = -kx(k)z^{-k-1} = -\frac{kx(k)z^{-k}}{z}$$

$$Z[kx(k)] = \sum_{k=-\infty}^{+\infty} kx(k)z^{-k} = \sum_{k=-\infty}^{+\infty} -z \frac{d[x(k)z^{-k}]}{dz} = -z \frac{d}{dz} \sum_{k=-\infty}^{+\infty} x(k)z^{-k} = -z \frac{d}{dz} X(z)$$

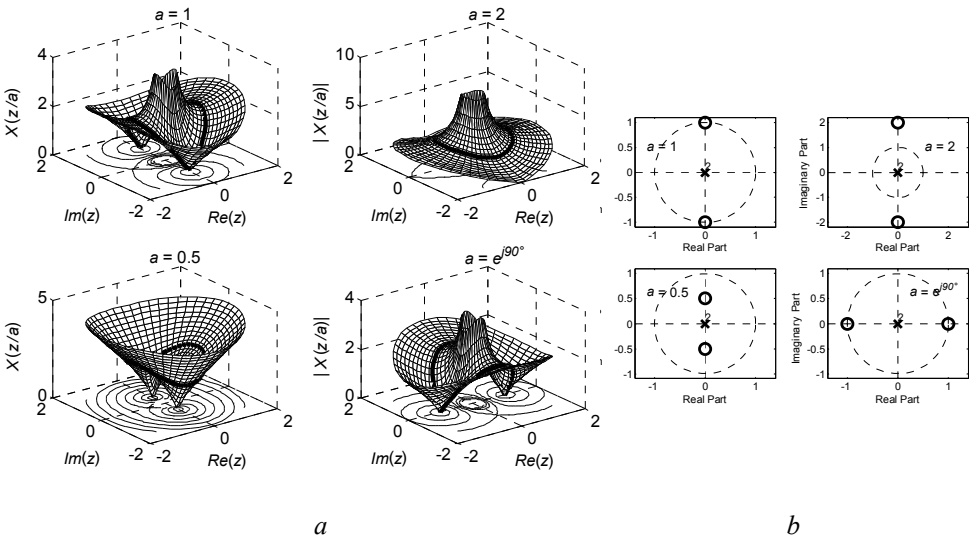


Figure 7.18. a - Scaling and complex modulation effects on the z-transform  
 b - Scaling and complex modulation effects on the zeros and poles location

**Reciprocal decay:** it derives from the integration of the  $k^{th}$  term  $x(k)z^{-k}$  of the z-transform of  $\{x(n)\}$ :

$$\int \frac{x(k)z^{-k}}{z} dz = -\frac{x(k)}{k} z^{-k}$$

$$z \left[ \frac{1}{k} x(k) \right] = \sum_{k=-\infty}^{+\infty} \frac{1}{k} x(k) z^{-k} = \sum_{k=-\infty}^{+\infty} - \int \frac{x(k)z^{-k}}{z} dz = - \int \frac{\sum_{k=-\infty}^{+\infty} x(k)z^{-k}}{z} dz = - \int \frac{X(z)}{z} dz$$

**Convolution:** on the basis of the z-transform definition, the convolution of two sequences is the product of their transformations:

$$z(x_1 * x_2) = X_1(z) \cdot X_2(z) \tag{7.25}$$

The ROC of the product  $X_1(z)X_2(z)$  is at least the intersection between the two ROCs of  $X_1$  and  $X_2$ . If there is a compensation between some zeros and poles of  $X_1(z)$  and  $X_2(z)$  the ROC can be wider than just  $ROC_1 \cap ROC_2$ .

**Multiplication:** More complicate is the transformation of the product of two sequences:

$$z(x_1 \cdot x_2) = \frac{1}{2\pi j} \oint_{\Gamma} X_1(\xi) X_2\left(\frac{z}{\xi}\right) \frac{d\xi}{\xi} \tag{7.26}$$

The integration contour  $\Gamma$  must lie inside the *ROC* of both the  $z$ -transforms  $X_1(\xi)$  and  $X_2(z/\xi)$ .

This property is particularly significant if considered with the Fourier transforms of  $\{x_1\}$  and  $\{x_2\}$ , since we obtain the classical convolution property of continuous functions:

$$\begin{aligned} F(x_1 \cdot x_2) &= \frac{1}{2\pi j} \int_{-\pi}^{+\pi} X_1(e^{j\vartheta}) \cdot X_1\left(\frac{e^{j\omega}}{e^{j\vartheta}}\right) \frac{je^{j\vartheta} d\vartheta}{e^{j\vartheta}} = \\ &= \frac{1}{2\pi} \int_{-\pi}^{+\pi} X_1(e^{j\vartheta}) \cdot X_2(e^{j\omega-\vartheta}) d\vartheta = X_1(e^{j\omega}) * X_2(e^{j\omega}) \end{aligned}$$

### Examples of $z$ -transforms.

**Example 1:**

The *ROC* of the  $z$ -transform of the sequence:

$$\{x(n)\} = \{\delta(n) - 0.8\delta(n-1)\}$$

is the whole complex plane, since  $\{x(n)\}$  is a finite length sequence, but it will not include the point  $z=0$  since the sequence has only terms  $x(n)$  with  $n > 0$ .

$X(z)$  can be computed starting from the  $z$ -transform of the unit pulse sequence, considering the linearity and shift properties, as follows

$$X(z) = Z[\delta(n)] - 0.8Z[\delta(n-1)] = Z[\delta(n)] - 0.8z^{-1}Z[\delta(n)] = \frac{z - 0.8}{z}$$

Figure 7.19 shows the pole-zero diagram for this sequence.

The Fourier transform of the sequence can be computed from  $X(z)$  replacing the  $z$  argument with  $e^{j\omega}$ :

$$X(e^{j\omega}) = 1 - 0.8e^{-j\omega} = 1 - 0.8\cos(\omega) + j0.8\sin(\omega)$$

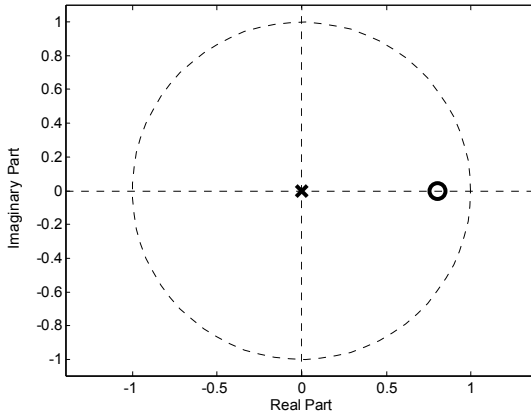


Figure 7.19. Pole-zero diagram for the sequence  $\{\delta(n) - 0.8\delta(n-1)\}$

The amplitude and phase of the Fourier transform are given instead by the following two expressions (see figure 7.20):

$$|X(e^{j\omega})| = \sqrt{[1 - 0.8 \cos(\omega)]^2 + [0.8 \sin(\omega)]^2} = \sqrt{1 + 0.8^2 - 2 \cdot 0.8 \cos(\omega)}$$

$$\angle X(e^{j\omega}) = \text{arctg} \left[ \frac{0.8 \sin(\omega)}{1 - 0.8 \cos(\omega)} \right]$$

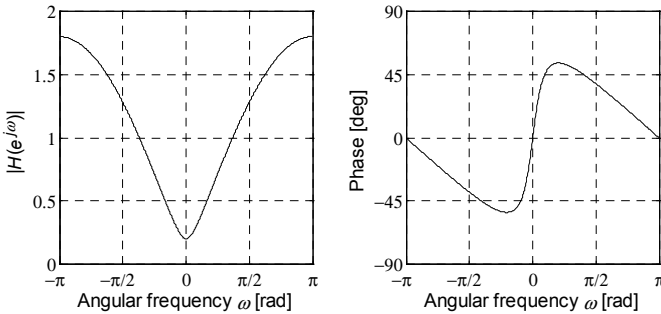


Figure 7.20. Amplitude and phase of the Fourier transform of  $\{\delta(n) - 0.8\delta(n-1)\}$

**Example 2:**

The z-transform of the finite length sequence

$$\{x(n)\} = \{\delta(n+2) - 0.8645\delta(n+1) + 0.64\delta(n)\},$$

since  $x(n) \neq 0$  for  $n \leq 0$ , is convergent for every  $z$  but  $z = \infty$ . To verify this result let us first compute  $X(z)$ :

$$X(z) = z^2 - 0.8645z + 0.64 = (z - j0.8)(z + j0.8)$$

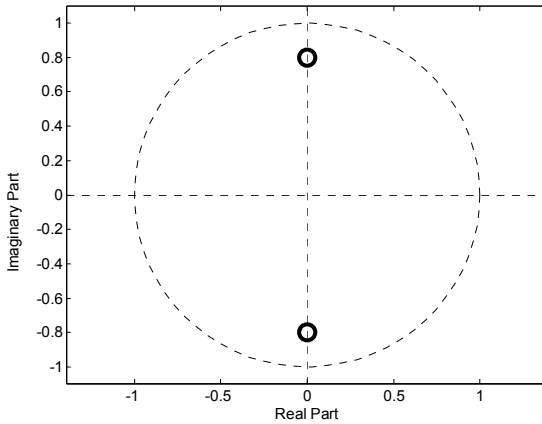


Figure 7.21. Pole-zero diagram for the sequence  $\{\delta(n) - 0.8645\delta(n-1) + 0.64\delta(n-2)\}$

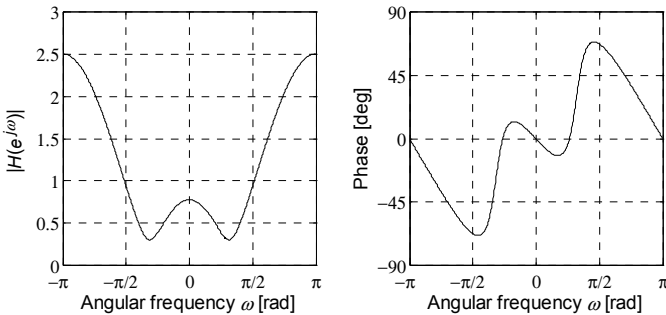


Figure 7.22. Amplitude and phase of the Fourier transform of the sequence  $\{\delta(n) - 0.8645\delta(n-1) + 0.64\delta(n-2)\}$

There are two zeros at  $z = j0.8$  and  $z = -j0.8$ , and  $X(z)$  is singular only in  $z = \infty$ . The zero-pole diagram for this sequence is shown in figure 7.21.

The Fourier transform of  $\{x(n)\}$  can be computed substituting in  $X(z)$  the complex variable  $z$  with  $e^{j\omega}$ :

$$\begin{aligned}
 X(e^{j\omega}) &= e^{j2\omega} - 0.8645e^{j\omega} + 0.64 = \\
 &= [\cos(2\omega) - 0.8645 \cos(\omega) + 0.64] + j[\sin(2\omega) - 0.8645 \sin(\omega)]
 \end{aligned}$$

Figure 7.22 shows both the amplitude and the phase of  $X(e^{j\omega})$ .

THE INVERSE  $z$ -TRANSFORM.

Given a sequence  $\{x(n)\}$ , the  $z$ -transformation formula (7.20) allow us to evaluate its  $z$ -transform  $X(z)$ . We will consider now the opposite operation, i.e. given the complex function  $X(z)$  and its  $ROC$ , we will see how to compute the elements of the sequence  $\{x(n)\}$  which admit  $X(z)$  as  $z$ -transform.

Let us consider first the integration of the generic term  $x(n)z^{-n}$  of the power series (7.20) along a circle  $\sigma$  with radius  $\rho$  placed in the  $ROC$  of  $X(z)$  and centered in the origin (see figure 7.23); on  $\sigma$  we can write  $z = \rho e^{j\omega}$  and we have:

$$\oint_{\sigma} x(n)z^{-n} dz = x(n) \int_0^{2\pi} \rho^{-n} e^{-jn\omega} j\rho e^{j\omega} d\omega = \begin{cases} 0 & \forall n \neq 1 \\ 2\pi j & n = 1 \end{cases} \quad (7.27)$$

Since the power series (7.20) is uniformly convergent in the  $ROC$  of  $X(z)$ , it can be integrated term-by-term. Considering (7.27), it follows:

$$\oint_{\sigma} X(z) dz = \sum_{k=-\infty}^{+\infty} \oint_{\sigma} x(k)z^{-k} dz = 2\pi j x(1) \quad (7.28)$$

Equation (7.28) shows that the first element of the sequence  $\{x(n)\}$ , i.e. the  $z^{-1}$  coefficient in  $X(z)$ , can be obtained by integration along  $\sigma$  of  $X(z)$ . If we consider now the  $z$ -transform  $X(z)z^{n-1}$ , it is given by:

$$X(z)z^{n-1} = \sum_{k=-\infty}^{+\infty} x(k)z^{-k} z^{n-1} = \sum_{k=-\infty}^{+\infty} x(k)z^{-k+n-1} \quad (7.29)$$

Since the  $z^{-1}$  coefficient in (7.29) is  $x(n)$ , according to (7.28) it is given by:

$$x(n) = \frac{1}{2\pi j} \oint_{\sigma} X(z)z^{n-1} dz \quad (7.30)$$

Equation (7.30) represents the inverse  $z$ -transform equation for  $X(z)$ .

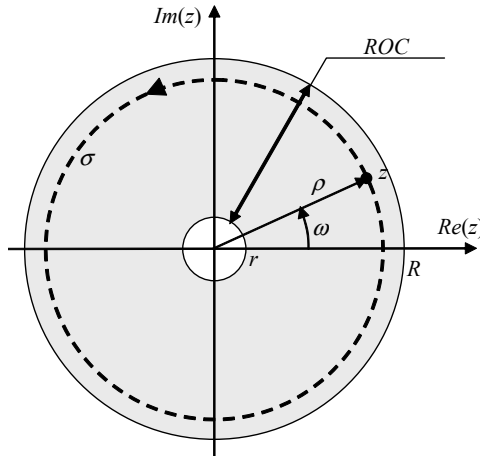


Figure 7.23. The value of  $x(1)$  can be determined by integration of  $X(z)$  along the circle  $\sigma$  laying in the ROC of  $X(z)$

Solving the line integrals (7.35) can be very tedious. The residue theorem offers a way for the computation of the line integral (7.30). This theorem states that  $x(n)$  results by the sum of the residues of the integrand function  $X(z)z^{n-1}$  at its  $N$  poles  $p_k$  inside the circle  $\sigma$ .

$$x(n) = \frac{1}{2\pi j} \oint_{\sigma} X(z)z^{n-1} dz = \sum_{k=1}^N \operatorname{Res}_{p_k} [X(z)z^{n-1}] \quad (7.31)$$

Since the function  $X(z)$  is rational, if  $m$  is the order of a pole  $p_k$ , the residue in  $z=p_k$  can be evaluated with the formula:

$$\operatorname{Res}_{z=p_k} [X(z)z^{n-1}] = \frac{1}{(m-1)!} \left\{ \frac{d^{m-1}}{dz^{m-1}} \left[ (z-p_k)^m X(z)z^{n-1} \right] \right\}_{z=p_k} \quad (7.32)$$

Another very convenient method for solving the inverse  $z$ -transform of a rational function  $X(z)=N(z)/D(z)$ , avoiding the complex line integration (7.30), is by taking its partial fractions. Each partial fraction may be inverted simply by looking up the table 7.1 of elementary  $z$ -transform.

Long division of the polynomials  $N(z)$  and  $D(z)$  is another possibility for computing the sequence  $\{x(n)\}$ .

### Examples of inverse $z$ -transforms.

**Example 1:** The inverse  $z$ -transform of  $X(z)=1/z$  according to (7.30) is given by:

$$x(n) = \frac{1}{2\pi j} \oint_{\sigma} X(z)z^{n-1} dz = \frac{1}{2\pi j} \oint_{\sigma} z^{n-2} dz$$

Since the ROC of  $X(z)^{n-1} = z^{n-2}$  is  $0 < |z| < +\infty$ , as line  $\sigma$  can be chosen any circle with radius  $0 < R < +\infty$  centered on the complex plane origin, where the variable  $z$  can be expressed as  $z = Re^{j\omega}$ . On  $\sigma$ , where results  $dz = jRe^{j\omega}d\omega$ , the previous inverse z-transform integral results in the following sequence:

$$x(n) = \frac{1}{2\pi j} \oint_{\sigma} z^{n-2} dz = \frac{1}{2\pi} \int_{-\pi}^{+\pi} R^{n-1} e^{j(n-1)\omega} d\omega = \begin{cases} 0 & \forall n \neq 1 \\ 1 & n = 1 \end{cases}$$

This result could be obtained directly considering that the inverse z-transform of  $X(z)=1$  is the impulse  $\delta(n)$  and, by the shift property, the inverse z-transform of  $z^{-1}$  is  $\{\delta(n-1)\}$ .

**Example 2:** The inverse z-transform of the function  $X(z)=z/(z-1)$  can be evaluated by the residue theorem. Since the function  $X(z)$  has one pole in  $z=1$  there two ROCs for this z-transform and, consequently, there are two different sequences converging to the same z-transform  $X(z)$ .

The integrand function in (7.30) for this example is given by  $X(z)z^{n-1}=z^n/(z-1)$ .

If the ROC is the region  $|z| < 1$ , the sequence will be left monolateral. As a matter of fact for  $n \geq 0$  there are no poles inside the line  $\sigma$  and the residue of  $X(z)z^{n-1}$  is zero and so is  $x(n)$  in accordance with (7.31). For  $n < 0$  there will be instead  $m = -n$  poles in  $z=0$ . There the residue is given by (7.32), with  $m=-n$ :

$$x(n) = \operatorname{Re}_s \left[ X(z)z^{n-1} \right]_{z=0} = \frac{1}{(-n-1)!} \left\{ \frac{d^{-n-1}}{dz^{-n-1}} \left[ \frac{1}{z-1} \right] \right\}_{z=0} = -1 \quad \forall n < 0$$

If the ROC of  $X(z)$  is the region  $|z| > 1$  the sequence must be right monolateral. In this case the integrand function  $z^n/(z-1)$  in (7.30) has one pole of order  $m=1$  in  $z=1$  when  $n \geq 0$ , where according (7.32) the residue is given by:

$$x(n) = \operatorname{Re}_s \left[ X(z)z^{n-1} \right]_{z=1} = \left\{ z^n \right\}_{z=1} = 1 \quad \forall n \geq 0$$

For  $n < 0$  there will be two poles inside  $\sigma$ , one in  $z=0$  of order  $m = -n$  and another one in  $z=1$  of order  $m=1$ . According to the residue formula (7.32) the sequence  $\{x(n)\}$  for  $n < 0$  is given by:

$$x(n) = \operatorname{Re}_s \left[ X(z)z^{n-1} \right]_{z=0} + \operatorname{Re}_s \left[ X(z)z^{n-1} \right]_{z=1} = -1 + 1 = 0 \quad \forall n < 0$$



**Example 3:** The inverse z-transform of:

$$X(z) = \frac{1}{1 - 0.8z^{-1}} = \frac{z}{z - 0.8}$$

can be computed directly, term by term, using the polynomial long division algorithm as shown below:

Sequence $\{x(n)\} \rightarrow$	1	$0.8z^{-1}$	$0.8^2z^{-2}$	$0.8^3z^{-3}$	...
$z$	$-0.8$	← Numerator of $X(z)$			
↑		$z$	$-0.8$		
Denominator of $X(z)$		$z$	$0.8$		
			$0.8$	$-0.8^2z^{-1}$	
			$0.8$	$0.8^2z^{-1}$	
				$0.8^2z^{-1}$	$-0.8^3z^{-2}$
				$0.8^2z^{-1}$	$0.8^3z^{-2}$
					...

**Example 4:** The inverse z-transform of the rational function  $X(z)$ :

$$X(z) = \frac{N(z)}{D(z)} = \frac{z^3 - 3}{(z-1)(z-2)} = \frac{z^3 - 3}{z^2 - 3z + 2}$$

can be computed first reducing  $X(z)$ , using the long division algorithm, to the sum of the quotient polynomial  $Q(z)$  and the ratio between the remainder  $R(z)$  and the denominator  $D(z)$  polynomials.

$z^2$	Quotient $\rightarrow$	$z$	3		
$-3z$	↑	$z^3$	0	0	-3
$+2$	Denominator of $X(z)$	$z^3$	$-3z^2$	$+2z$	0
			$3z^2$	$-2z$	-3
			$3z^2$	$-9z$	$+6$
				$7z$	$-9$
					← Remainder

$$X(z) = \frac{N(z)}{D(z)} = Q(z) + \frac{R(z)}{D(z)} = \frac{z+3}{z-1} + \frac{7z-9}{(z-1)(z-2)}$$

The inversion of  $Q(z)$  can be obtained directly as sum of shifted impulse sequences. The inversion of  $R(z)/D(z)$  can be obtained instead, after partial factor expansion, by the residue theorem applied to each factor. The partial fraction expansion of  $R(z)/D(z)$  is:

$$\frac{R(z)}{D(z)} = \frac{A}{z-1} + \frac{B}{z-2}$$

with A and B given by

$$\begin{cases} A+B=7 \\ 2A+B=9 \end{cases} \quad \begin{cases} A=2 \\ B=5 \end{cases}$$

giving the following expression for  $X(z)$ :

$$X(z) = z + 3 + \frac{2}{z-1} + \frac{5}{z-2}$$

The inverse  $z$ -transform of  $X(z)$  is then:

$$x(n) = \delta(n+1) + 3\delta(n) + \sum_k \operatorname{Re} s \left[ \frac{2}{z-1} z^{n-1} \right] + \sum_j \operatorname{Re} s \left[ \frac{5}{z-2} z^{n-1} \right]$$

The poles  $p_k$  and  $p_j$  in the residues depends on the ROC where to make the inversion. In this example the function  $X(z)$  admits three ROCs, each associated to different poles and thus to different sequences.

In the region  $D_1$ , where  $0 \leq |z| < 1$ , for  $n \geq 1$  there are no poles and results

$$\sum_k \operatorname{Re} s \left[ \frac{2}{z-1} z^{n-1} \right] = 0$$

$$\sum_j \operatorname{Re} s \left[ \frac{5}{z-2} z^{n-1} \right] = 0$$

For  $n < 1$  there are  $m = -n+1$  poles  $p_k$  and  $p_j$  in  $z=0$ , giving the following residues:

$$\operatorname{Re} s \left[ \frac{2z^{n-1}}{z-1} \right]_{z=0} = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 2(z-1)^{-1} \right] \right\}_{z=0} = -2$$

$$\operatorname{Re} s \left[ \frac{5z^{n-1}}{z-2} \right]_{z=0} = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 5(z-2)^{-1} \right] \right\}_{z=0} = -5 \cdot 2^{n-1}$$

In  $D_1$  the left monolateral inverse  $z$ -transform is:

$$x(n) = \begin{cases} \delta(n+1) + \delta(n) = 0 & \forall n \geq 1 \\ \delta(n+1) + \delta(n) - 2 - 5 \cdot 2^{n-1} & \forall n < 1 \end{cases}$$

In the region  $D_2$ , where  $1 < |z| < 2$ , for  $n \geq 1$  there is only a simple pole  $p_k$  in  $z=1$ , giving the following residue:

$$\operatorname{Re} s \left[ \frac{2}{z-1} z^{n-1} \right]_{z=1} = \left\{ \left[ 2z^{n-1} \right] \right\}_{z=1} = 2$$

For  $n < 1$  there is one pole  $p_k$  in  $z=1$ , of order  $m=1$ , and in  $z=0$  one pole  $p_k$  and one pole  $p_j$  of order  $m = -n+1$ . The three residues are:

$$\operatorname{Re} s \left[ \frac{2}{z-1} z^{n-1} \right]_{z=1} = \left\{ \left[ 2z^{n-1} \right] \right\}_{z=1} = 2$$

$$\operatorname{Re} s \left[ \frac{2z^{n-1}}{z-1} \right]_{z=0} = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 2(z-1)^{-1} \right] \right\}_{z=0} = -2$$

$$\operatorname{Re} s \left[ \frac{5z^{n-1}}{z-2} \right]_{z=0} = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 5(z-2)^{-1} \right] \right\}_{z=0} = -5 \cdot 2^{n-1}$$

In  $D_2$  the bilateral inverse  $z$ -transform is:

$$x(n) = \begin{cases} \delta(n+1) + \delta(n) + 2 = 2 & \forall n \geq 1 \\ \delta(n+1) + \delta(n) + 2 - 2 - 5 \cdot 2^{n-1} = \delta(n+1) + \delta(n) - 5 \cdot 2^{n-1} & \forall n < 1 \end{cases}$$

In the region  $D_3$ , where  $2 < |z| < +\infty$ , for  $n \geq 1$  there are a simple pole  $p_k$  in  $z=1$  and  $p_j$  in  $z=2$ , giving the following residues:

$$\operatorname{Re} s \left[ \frac{2}{z-1} z^{n-1} \right]_{z=1} = \left\{ \left[ 2z^{n-1} \right] \right\}_{z=1} = 2$$

$$\operatorname{Res}_{z=2} \left[ \frac{5z^{n-1}}{z-2} \right] = \left\{ \left\{ 5z^{n-1} \right\} \right\}_{z=2} = 5 \cdot 2^{n-1}$$

For  $n < 1$  the poles  $p_k$  are in  $z=1$  of order  $m=1$  and in  $z=0$  of order  $m=-n+1$ . The poles  $p_j$  are in  $z=2$  of order  $m=1$  and in  $z=0$  of order  $m=-n+1$ :

$$\operatorname{Res}_{z=1} \left[ \frac{2}{z-1} z^{n-1} \right] = \left\{ \left\{ 2z^{n-1} \right\} \right\}_{z=1} = 2$$

$$\operatorname{Res}_{z=0} \left[ \frac{2z^{n-1}}{z-1} \right] = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 2(z-1)^{-1} \right] \right\}_{z=0} = -2$$

$$\operatorname{Res}_{z=2} \left[ \frac{5z^{n-1}}{z-2} \right] = \left\{ \left\{ 5z^{n-1} \right\} \right\}_{z=2} = 5 \cdot 2^{n-1}$$

$$\operatorname{Res}_{z=0} \left[ \frac{5z^{n-1}}{z-2} \right] = \frac{1}{(-n)!} \left\{ \frac{d^{-n}}{dz^{-n}} \left[ 5(z-2)^{-1} \right] \right\}_{z=0} = -5 \cdot 2^{n-1}$$

In  $D_3$  the right monolateral inverse  $z$ -transform is:

$$x(n) = \begin{cases} \delta(n+1) + \delta(n) + 2 + 5 \cdot 2^{n-1} = 2 + 5 \cdot 2^{n-1} & \forall n \geq 1 \\ \delta(n+1) + \delta(n) + 2 - 2 + 5 \cdot 2^{n-1} - 5 \cdot 2^{n-1} = \delta(n+1) + \delta(n) & \forall n < 1 \end{cases}$$

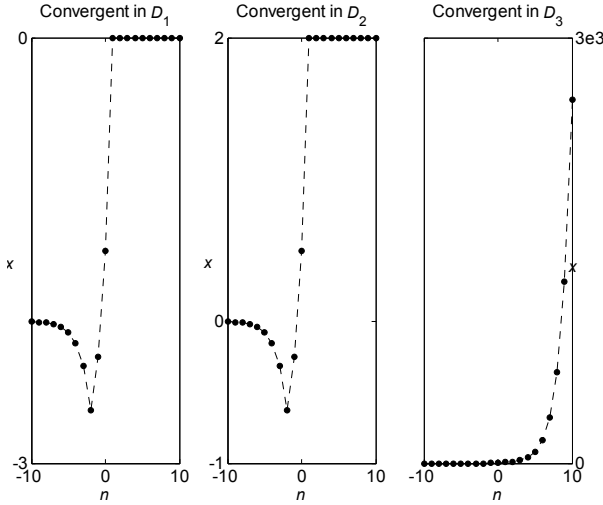


Figure 7.24. Inverse  $z$ -transform of  $X(z)=(z^3-3)/(z^2-3z+2)$  for three different ROCs of  $X(z)$

**$z$ -DOMAIN REPRESENTATION OF CAUSAL DISCRETE LINEAR SHIFT INVARIANT SYSTEMS.**

Considering the  $z$ -transform of the state equations and of the output transformation for a linear shift invariant causal discrete system (7.6) it is possible to extend the well known concept of transfer function of a linear time invariant continuous systems to a linear shift invariant discrete systems. Applying the  $z$ -transform to the state equation (7.6), and considering that this equation is valid only for  $n > n_0$  and that  $x(n)=0, \forall n < n_0$ , it results:

$$\begin{aligned} \mathbf{X}(z) &= \mathcal{Z}[\mathbf{x}(n)] = \sum_{k=-\infty}^{+\infty} \mathbf{x}(k)z^{-k} = \mathbf{x}(n_0)z^{-n_0} + \sum_{k=n_0+1}^{+\infty} \mathbf{x}(k)z^{-k} = \\ &= \mathbf{x}(n_0)z^{-n_0} + z^{-1}[\mathbf{A} \cdot \mathbf{X}(z) + \mathbf{b} \cdot U(z)] \end{aligned} \tag{7.33}$$

Applying (7.33) to the output transformation (7.6) it is possible to establish the following relationship between the  $z$ -transform of the input and output sequences:

$$\begin{aligned} Y(z) &= \mathbf{c}^T \mathcal{Z}(\mathbf{x}) + d \cdot \mathcal{Z}(u) = \\ &= \underbrace{z^{-n_0} \mathbf{c}^T (z\mathbf{I} - \mathbf{A})^{-1} \mathbf{x}(n_0)}_{ZIR} + \underbrace{[\mathbf{c}^T (z\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} + d]}_{ZSR} U(z) \end{aligned} \tag{7.34}$$

As for the  $n$ -domain analysis also in the  $z$ -domain the output is composed by two terms. The first is the system response to the initial state with zero input; the second one

is the system response to the input sequence with zero vector state. Figure 7.25 shows in a block diagram the  $z$ -domain representation (7.34) of a linear discrete system.

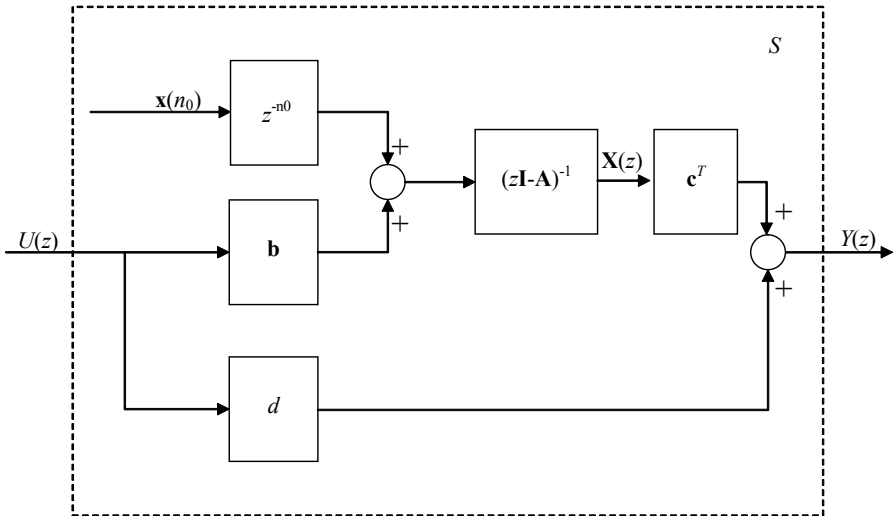


Figure 7.25. Block diagram representing in the  $z$ -domain a linear shift invariant dynamical discrete system

The complex function in squared brackets is defined as the *transfer function* of the discrete linear shift invariant system. Comparing (7.34) with the convolution product (7.9) it results that the transfer function of a discrete system is the  $z$ -transform of its impulse response  $h(n)$  with zero vector state:

$$H(z) \stackrel{\Delta}{=} \mathbf{c}^T (\mathbf{zI} - \mathbf{A})^{-1} \mathbf{b} + d = z(h) = \frac{z(y)}{z(u)} \quad (7.35)$$

The transfer function  $H(z)$  in (7.35) is a rational function. The roots of the numerator are called the zeros of the transfer function, while the roots of the denominator are the poles of the system. The poles of the transfer function thus are the values of  $z$  for which the matrix  $(\mathbf{zI} - \mathbf{A})$  is singular; they are the roots of the characteristic polynomial of the matrix  $\mathbf{A}$ .

Since system stability requires that all the system poles must have module smaller than one, it means that they all must lay inside the unit circle. The *BIBO* system stability instead requires that the *ROC* of  $H(z)$  includes the unit circle. As a matter of fact from this condition it follows that

$$|H(e^{j\omega})| = \left| \sum_{k=-\infty}^{+\infty} h(k)e^{-j\omega k} \right| \leq \sum_{k=-\infty}^{+\infty} |h(k)| \cdot |e^{-j\omega k}| = \sum_{k=-\infty}^{+\infty} |h(k)| < \infty \quad (7.36)$$

The converse is also true for rational  $H(z)$ .

System causality is another important property, completely unrelated to *BIBO* stability. It implies that the *ROC* of  $H(z)$  is outside the outermost pole.

As already mentioned before the *BIBO* stability is a weaker concept than asymptotic stability: a system must be causal and *BIBO* stable for having all poles inside the unit circle and consequently being asymptotically stable.

**Example 1:** The system with the following transfer function:

$$H(z) = \frac{1}{(z-1.5)(z-0.5)}$$

is

- causal and *BIBO* unstable if the *ROC* is  $|z| > 1.5$ ;
- anti-causal and *BIBO* unstable if the *ROC* is  $|z| < 0.5$ ;
- bilateral and *BIBO* stable if the *ROC* is  $0.5 < |z| < 1.5$ .

Since one pole is outside the unit circle it is not asymptotically stable.

**Example 2:** The system with transfer function:

$$H(z) = \frac{1}{(z-0.8)(z-0.5)}$$

is

- causal and *BIBO* stable (thus asymptotically stable as well) if the *ROC* is  $|z| > 0.8$ ;
- anti-causal and *BIBO* unstable (thus unstable) if the *ROC* is  $|z| < 0.5$ ;
- bilateral and *BIBO* stable (but not asymptotically stable) if the *ROC* is  $0.5 < |z| < 0.8$ .

Another important interpretation of the transfer function  $H(z)$  derives directly from the analysis of the response of a linear system with zero state to an exponential input sequence  $\{\tilde{z}^n\}$ . This is given by the convolution product between the system impulse response  $h(n)$  and the input sequence  $\tilde{z}^n$ , i.e.:

$$y(n) = h(n) * \tilde{z}^n = \sum_{k=-\infty}^{+\infty} h(k) \tilde{z}^{n-k} = \tilde{z}^n \sum_{k=-\infty}^{+\infty} h(k) \tilde{z}^{-k} \quad (7.37)$$

Since  $H(z)$  is the  $z$ -transform of  $h(n)$ , (7.37) can be rewritten as:

$$y(n) = H(\tilde{z}) \cdot \tilde{z}^n \tag{7.38}$$

Equation (7.38) shows that for linear shift invariant discrete dynamical systems the response  $\{y(n)\}$  to an exponential input  $\{\tilde{z}^n\}$  is an exponential sequence as well, weighed by a complex numerical factor given by the transfer function  $H(z)$  evaluated at  $z = \tilde{z}$ , i.e. for linear shift invariant discrete dynamical systems the system's eigenfunctions are the exponential sequences.

In chapter 2 it was shown how to compute the impulse response of a system composed by a connection of various subsystems and represented by a block diagram in the  $n$ -domain. This computation is generally quite cumbersome. Even in the simple case of two subsystems connected in cascade the impulse response is given by the convolution product of the impulse responses of the two subsystems. Much easier is the elaboration of the block diagram represented in the  $z$ -domain. In this case all the convolution products are replaced by simple products. In figure 7.26 are given some simple examples of block interconnection and their transfer function  $H(z)$ .

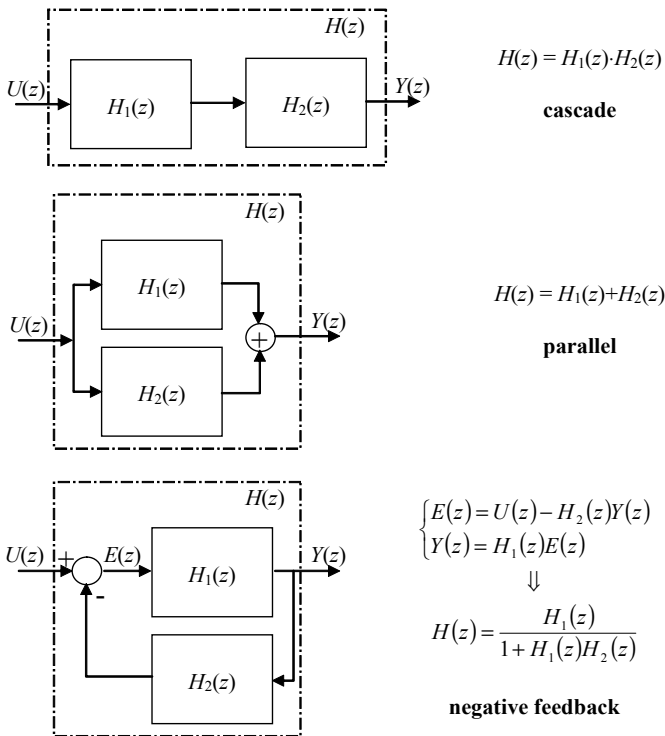


Figure 7.26. Examples of discrete systems interconnections in the  $z$ -domain



LINEAR SHIFT INVARIANT DISCRETE FILTERS.

In digital signal processing the most interesting class of linear shift invariant systems are the discrete filters of order  $q$ , for which the state variables at step  $n$  are the values of the system output at the  $N$  previous steps and of the input at the  $M$  previous steps:

$$\mathbf{x}(n) = [y(n-1) \quad \dots \quad y(n-N) \quad u(n-1) \quad \dots \quad u(n-M)]^T \quad (7.39)$$

Considering the output transformation in (7.6), the output of a discrete filter can be written as:

$$y(n) = \mathbf{c}^T \mathbf{x}(n) + du(n) = \sum_{k=1}^N c_k y(n-k) + \sum_{k=1}^M c_{k+N} u(n-k) + du(n) \quad (7.40)$$

The previous difference equation (7.40) can also be written as:

$$y(n) = \sum_{k=0}^M \alpha_k u(n-k) - \sum_{k=1}^N \beta_k y(n-k) \quad (7.41)$$

with  $\alpha_k = c_{k+N}$ ,  $\alpha_0 = d$  and  $\beta_k = -c_k$

The transfer function for this type of system can be evaluated by applying the  $z$ -transform to (7.41):

$$Z(y) = \sum_{k=0}^M \alpha_k z^{-k} Z(u) - \sum_{k=1}^N \beta_k z^{-k} Z(y) \quad (7.42)$$

Hence factorizing (7.42) and applying relationship (7.35) it follows:

$$H(z) = \frac{Z(y)}{Z(u)} = \frac{\sum_{k=0}^M \alpha_k z^{-k}}{\sum_{k=0}^N \beta_k z^{-k}} \quad (7.43)$$

In (7.43) at the coefficient  $\beta_0$  is given the value 1.

If  $Z_k$  and  $P_k$  are respectively zeros and poles of the transfer function, the polynomials at numerator and denominator of (7.43) can be factorized giving a different expression of the transfer function  $H(z)$ :

$$H(z) = \alpha_0 \frac{\prod_{k=1}^M (1 - Z_k z^{-1})}{\prod_{k=1}^N (1 - P_k z^{-1})} = \alpha_0 z^{N-M} \frac{\prod_{k=1}^M (z - Z_k)}{\prod_{k=1}^N (z - P_k)} \quad (7.44)$$

The order  $q$  of a discrete system is the minimum number of state variables necessary for the computation of the output sequence and is equal to the number of the system's poles. For discrete filters this is less than  $M+N$ . As a matter of fact from (7.44) results that the discrete filter has  $N$  poles in  $P_k$  plus other  $M-N$  poles in the origin only if  $M>N$ . The entire number of poles, i.e. the system's order, is thus the greater number between  $M$  and  $N$ .

The complex function (7.44) can be split in two equations, one for the amplitude of  $H(z)$  and another one for the phase:

$$|H(z)| = |\alpha_0| \cdot |z^{N-M}| \cdot \frac{\prod_{k=1}^M |z - Z_k|}{\prod_{k=1}^N |z - P_k|} \quad (7.45a)$$

$$\angle[H(z)] = \angle\alpha_0 + \angle(z^{N-M}) + \sum_{k=1}^M \angle(z - Z_k) - \sum_{k=1}^N \angle(z - P_k) \quad (7.45b)$$

The frequency response of the system can be derived from the polar expressions of  $H(z)$  (7.45) simply by substitution of the complex variable  $z$  with  $e^{j\omega}$ .

$$|H(e^{j\omega})| = |\alpha_0| \cdot \frac{\prod_{k=1}^M |e^{j\omega} - Z_k|}{\prod_{k=1}^N |e^{j\omega} - P_k|} \quad (7.46a)$$

$$\angle[H(e^{j\omega})] = \angle\alpha_0 + (N - M)\omega + \sum_{k=1}^M \angle(e^{j\omega} - Z_k) - \sum_{k=1}^N \angle(e^{j\omega} - P_k) \quad (7.46b)$$

If in (7.46a) the generic pole and zero is expressed in polar form as  $\gamma e^{j\theta}$ , the single factor in (7.46a) is given by:

$$|e^{j\omega} - \gamma e^{j\theta}| = |1 - \gamma e^{j\theta} e^{-j\omega}| = \sqrt{1 + \gamma^2 - 2\gamma \cos(\theta - \omega)} \quad (7.47)$$

Since  $\cos(\theta-\omega)$  is maximum for  $\theta=\omega$  (7.47) will be minimum at that frequency. This means that, if (7.47) represents a zero in (7.46a), there will be a dip in the frequency response at that frequency. If (7.47) represents a pole of (7.46a) the magnitude of the frequency response will increase. Figure 7.27 shows an example of this behavior for a transfer function with two poles at  $z=0.8e^{\pm j\pi/4}$  and a zero at  $z=1.2e^{j\pi}$ .

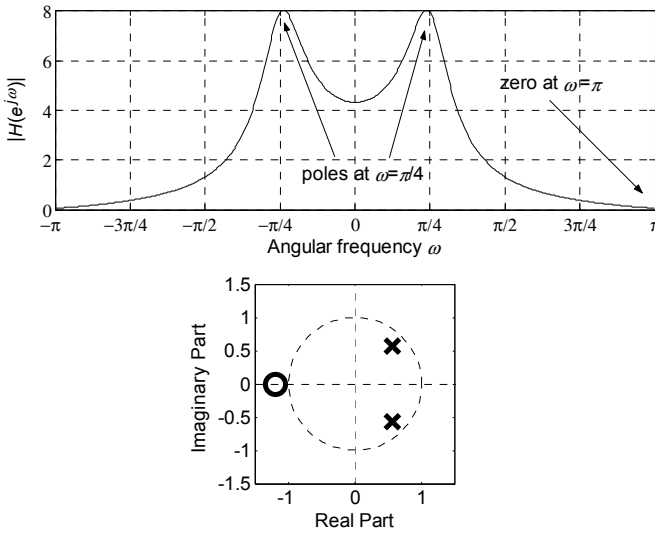


Figure 7.27. Frequency magnitude response and zero-pole diagram for the transfer

$$\text{function } H(z) = \frac{z + 1.2}{(z - 0.8e^{j\pi/4})(z - 0.8e^{-j\pi/4})}$$

The geometric interpretation of (7.47) is given in figure 7.28, where the magnitude response of (7.47) is given by traveling along the unit circle and evaluating the length of the vector  $r$ .

For a specific frequency  $\omega$  the magnitude of  $H(e^{j\omega})$  in (7.46a) can be obtained by multiplying the length of all vectors connecting all the zeros to the point on the unit circle at angle  $\omega$ , and dividing by the length of all vectors connecting all the poles to the same point on the unit circle and multiplying the result by  $|\alpha_0|$ . If there is a zero or a pole on the unit circle the magnitude response goes either to zero or infinity, respectively, at that frequency. Poles and zeros at the origin and at infinity do not affect the magnitude response because they are equidistant from every point of the unit circle.

The phase response of the generic pole or zero term in (7.46) is:

$$\angle(e^{j\omega} - \gamma e^{j\theta}) = \angle r = \phi \tag{7.48}$$

The geometric interpretation of (7.48) and (7.46b) is that for a specific frequency  $\omega$  the phase of  $H(e^{j\omega})$  in (7.46b) can be obtained by summing together the angles of all vectors connecting all the zeros to the point on the unit circle at angle  $\omega$ , and subtracting the angles of all vectors connecting all the poles to the same point on the unit circle and adding  $\angle\alpha_0+(N-M)\omega$  to the result. If there is a zero or a pole on the unit circle the phase flips  $\pi$  radians at that frequency. Poles and zeros at infinity do not affect the phase response, while poles and zeros in the origin add the linear phase term  $(N-M)\omega$ .

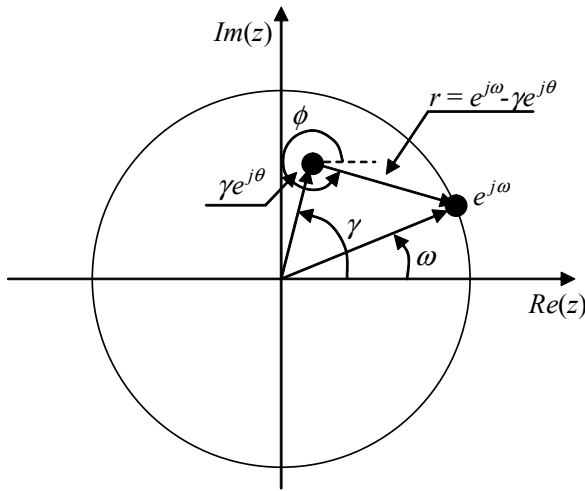


Figure 7.28. Geometric view for the computation of amplitude and phase of the generic term  $r = z - \gamma e^{j\theta}$  evaluated on the unit circle  $z = e^{j\omega}$

RELATIONSHIPS BETWEEN SEQUENCES AND SAMPLED FUNCTIONS IN THE  $z$  AND  $s$  DOMAINS.

As sequences can be analyzed by the  $z$ -transform so the main tool for the analysis of continuous functions is the Laplace transform. The relationships between the two transformation tools, when the discrete sequences derive from sampling continuous functions, can be established by the shift theorem. Let us first consider a continuous function  $g(t)$  of the variable  $t$  and its Laplace transform:

$$G(s) = L[g(t)] \tag{7.49}$$

The shift theorem states that:

$$L[g(t - kT)] = e^{-kTs} \cdot G(s) \tag{7.50}$$

If the discrete sequence  $\{x(n)\}$  is the result of samples taken from the continuous function  $g(t)$ , with sampling period  $T$ , it can be represented as:

$$\{x(n)\} = \sum_{n=-\infty}^{+\infty} g(nT)\delta(t-nT) \quad (7.51)$$

Equation (7.51) establishing a link between the sequence  $\{x(n)\}$  and the function  $g(t)$ , allows the interpretation of the series of samples either as a sequence (left-hand side of (7.51)) or a function (right-hand side of (7.51)). The  $z$ -transform and the Laplace transform of the sequence/function (7.51) are expressed by:

$$X_D(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \quad (7.52a)$$

$$X_C(s) = \sum_{n=-\infty}^{+\infty} g(nT)e^{-nTs} = \sum_{k=-\infty}^{+\infty} x(n)e^{-nTs} \quad (7.52b)$$

In (7.52b) the shift theorem (7.50) has been applied to the Laplace transform of the impulse function  $\delta(t-nT)$ . By forcing the two transforms (7.52) *a* and *b* to be equal the following relationship between the complex variables  $z$  and  $s$  must be satisfied:

$$z = e^{Ts} \quad (7.53a)$$

$$s = \frac{1}{T} \ln z \quad (7.53b)$$

The maps (7.53) *a* and *b* are not bijective. The mapping (7.53a) maps several points in the  $s$ -plane to the same point in the  $z$ -plane. Vice versa the mapping (7.53b) maps one single point in the  $z$ -plane to several points in the  $s$ -plane. This is another interpretation of the aliasing effect described in chapter 3. To better understand this behavior let us write the complex variable  $s$  as

$$s = \sigma + j\Omega \quad (7.54)$$

Replacing (7.54) into relationship (7.53a) it is equivalent to

$$z = e^{T\sigma} e^{jT\Omega} = e^{T\sigma} [\cos(T\Omega) + j \sin(T\Omega)] \quad (7.55)$$

From (7.55) it results that every point in the  $s$ -plane  $\bar{s} = \bar{\sigma} + j(\bar{\Omega} + k2\pi/T)$  is mapped in the  $z$ -plane to the same point  $\bar{z} = e^{j\bar{\sigma}T} [\cos(\bar{\Omega}T) + j \sin(\bar{\Omega}T)]$  and, inversely each point  $\bar{z}$  is mapped to infinite points in the  $s$ -plane aligned on a vertical line and separated by  $2\pi/T$ , as shown in figure 7.29. This property of the conformal map  $z=e^{Ts}$  means that given the  $z$ -transform  $X_D(z)$  of the samples sequence  $\{x(n)\}$  the related Laplace transform  $X_C(s)$  must be periodic in each strip of the  $s$ -plane as shown in figure 7.29.

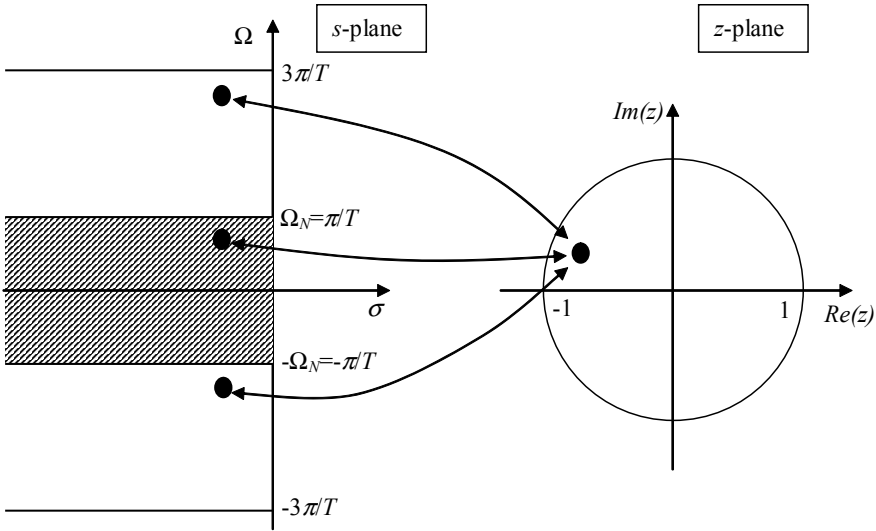


Figure 7.29. Graphical representation of the  $2\pi/T$  periodicity in the  $s$ -domain along lines parallel to the imaginary axis due to the conformal mapping  $z=e^{Ts}$

From (7.55) the following properties of the  $s \rightarrow z$  mapping can be easily derived (see figure 7.29 and 7.30):

- the vertical lines  $\text{Re}(s) = \bar{\sigma} = \text{const.}$ , parallel to the imaginary axis in the  $s$ -plane, maps onto circle of radius  $e^{T\bar{\sigma}}$  in the  $z$ -plane;
- the imaginary axis  $s=j\Omega$  of the  $s$ -plane maps to the unit circle  $z=e^{jT\Omega}$  in the  $z$ -plane;
- the horizontal lines  $\text{Im}(s) = \bar{\Omega} = \text{const.}$ , parallel to the real axis in the  $s$ -plane, maps onto radial lines with slope  $T\bar{\Omega}$  in the  $z$ -plane;
- the DC point  $s=0$  maps to  $z=e^{j0}=1$ ;
- the Nyquist angular frequency  $\Omega_N = \pm\pi/T$  maps to  $z=e^{\pm j\pi}=-1$ ;
- the stable left-hand half plane  $\sigma < 0$  in the  $s$  domain is mapped to the interior of the unit circle in the  $z$ -plane.

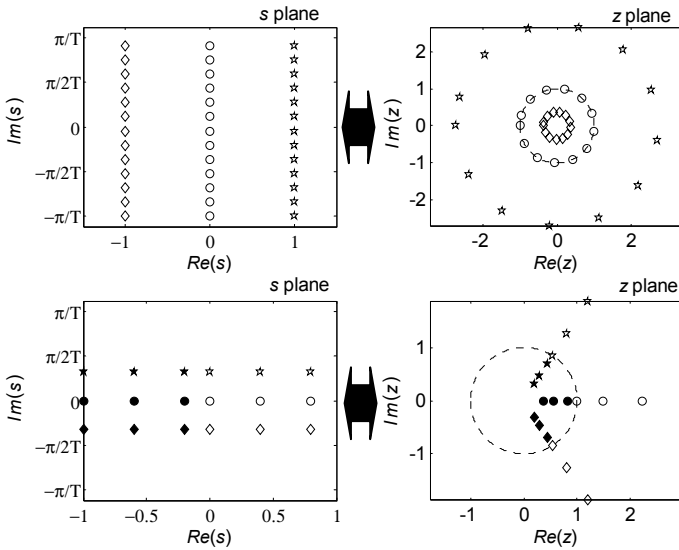


Figure 7.30. Properties of the conformal map  $z = e^{Ts}$

Expression (7.51) can also be Fourier transformed. If  $\omega \in [-\pi, \pi)$  represents the angular frequency in the discrete domain and  $\Omega \in (-\infty, \infty)$  the angular frequency in the continuous domain, the two transformations can be obtained replacing  $z$  with  $e^{j\omega}$  and  $s$  with  $j\Omega$  in (7.52) *a* and *b*, giving the following expressions for the two Fourier transforms:

$$X_D(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-jn\omega} \tag{7.56a}$$

$$X_C(j\Omega) = \sum_{k=-\infty}^{+\infty} x(k)e^{-jnT\Omega} \tag{7.56b}$$

Forcing the two Fourier transforms (7.56) *a* and *b* to be the same the following relationship between the angular frequencies  $\omega$  and  $\Omega$  must be satisfied:

$$\Omega = \frac{\omega}{T} + k \frac{2\pi}{T}, \quad k = 0, \pm 1, \pm 2, \dots \tag{7.57}$$

Relationship (7.57) is very important because it shows that the two Fourier transforms  $X_D(e^{j\omega})$  and  $X_C(j\Omega)$  in the discrete and continuous domain are both periodic, the former with period  $2\pi$  and the latter with period  $2\pi/T$ . One period of Fourier transform  $X_C(j\Omega)$

can be obtained from one period of the Fourier transform  $X_D(e^{j\omega})$  simply by scaling the frequency axis by  $T$ :

$$X_C(j\Omega) = X_D(e^{j\Omega T}) \tag{7.58}$$

with  $\Omega \in [-\pi/T, \pi/T)$ .

More complex instead is the relationship between the Fourier transform  $G(j\Omega)$  and  $X_C(j\Omega)$ . It can be derived directly from the sampling theory developed in chapter 3 applying equation (3.12):

$$X_C(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} G\left(j\Omega + j\frac{2\pi k}{T}\right) \tag{7.59}$$

This results can be also derived rewriting (7.51) as:

$$\{x(n)\} = \sum_{n=-\infty}^{+\infty} g(t)\delta(t-nT) = g(t) \sum_{n=-\infty}^{+\infty} \delta(t-nT) \tag{7.60}$$

and considering the Fourier series of the periodic function  $f(t) = \sum_{n=-\infty}^{+\infty} \delta(t-nT)$ , with period  $T$ :

$$\begin{aligned} f(t) &= \sum_{n=-\infty}^{+\infty} \delta(t-nT) \\ c_k &= \frac{1}{T} \int_{-T/2}^{+T/2} f(t) e^{-jk\frac{2\pi}{T}t} dt = \frac{1}{T} \int_{-T/2}^{+T/2} \delta(t) e^{-jk\frac{2\pi}{T}t} dt = \frac{1}{T} \\ f(t) &= \sum_{k=-\infty}^{+\infty} c_k e^{jk\frac{2\pi}{T}t} = \frac{1}{T} \sum_{k=-\infty}^{+\infty} e^{jk\frac{2\pi}{T}t} \end{aligned} \tag{7.61}$$

Replacing (7.61) into (7.60) and taking the Fourier transform of the resulting function, it follows:

$$\begin{aligned} X_C(j\Omega) &= G(j\Omega) * \frac{1}{T} \sum_{n=-\infty}^{+\infty} \delta\left(j\Omega + jn\frac{2\pi}{T}\right) = \\ &= \frac{1}{T} \sum_{n=-\infty}^{+\infty} G\left(j\Omega + jn\frac{2\pi}{T}\right) \end{aligned} \tag{7.62}$$

in agreement with the sampling theory result (7.59).



**Example:** Given the function

$$g(t) = \begin{cases} 0 & \forall t < 0 \\ e^{2t} & \forall t \geq 0 \end{cases}$$

its Laplace transform is

$$G(s) = \frac{1}{s-2}$$

The Fourier transform of  $g(t)$  can be derived from  $G(s)$ :

$$G(j\Omega) = \frac{1}{j\Omega - 2}$$

If the function  $g(t)$  is sampled with a sampling frequency  $T$ , the sequence obtained with these samples is expressed as

$$\{x(n)\} = \begin{cases} 0 & \forall n < 0 \\ e^{2nT} & \forall n \geq 0 \end{cases}$$

The z-transform of  $\{x(n)\}$  is (see z-transform table 7.1)

$$X_D(z) = \frac{z}{z - e^{2T}}$$

The Fourier transform of  $\{x(n)\}$  can be derived from  $X_D(z)$ :

$$X_D(e^{j\omega}) = \frac{e^{j\omega}}{e^{j\omega} - e^{2T}}$$

The Fourier transform as a function of the analog angular frequency  $\Omega$  can be obtained using the scaling (7.58):

$$X_C(j\Omega) = \frac{e^{j0.1\Omega}}{e^{j0.1\Omega} - e^{2T}}$$

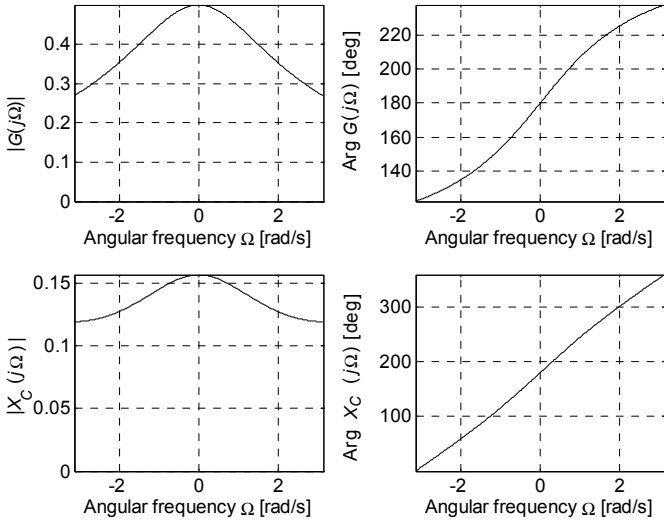


Figure 7.31. Magnitude and phase of the Fourier transforms  $G(j\Omega)$  and  $X_c(j\Omega)$  of the function  $g(t)=e^{-2t}$  and of the sequence  $x(n) = g(nT)$ , respectively, when the sampling period is  $T=1$  s

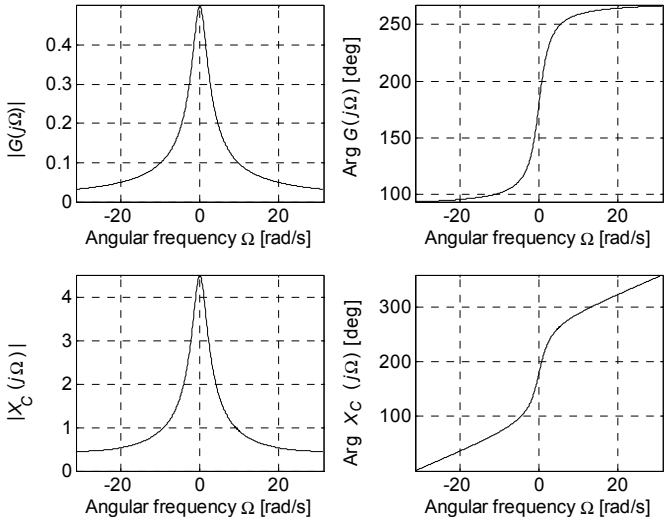


Figure 7.32. Magnitude and phase of the Fourier transforms  $G(j\Omega)$  and  $X_c(j\Omega)$  of the function  $g(t)=e^{-2t}$  and of the sequence  $x(n) = g(nT)$ , respectively, when the sampling period is  $T=0.1$  s

In figure 7.31 and 7.32 are shown the Fourier transforms  $G(j\Omega)$  of the continuous function and  $X_C(j\Omega)$  of the samples sequence with sampling period equal to 1 s and 0.1 s respectively. The two Fourier transforms are not equal because of the aliasing effect. In particular in figure 7.31 the great difference between  $X_C(j\Omega)$  and  $G(j\Omega)$  is due to the long sampling period ( $T = 1$  s) compared to the exponential time constant in  $g(t)$  ( $\tau = 1/2$  s). This difference, apart for a scaling factor, is greatly reduced in figure 7.32 since the sampling period has been reduced to 0.1 s.

In figure 7.33 is shown instead the Fourier transform  $X_D(e^{j\omega})$  obtained with a sampling period of 0.1 s. Comparing this picture with figure 7.32 it is possible to see the equivalence with  $X_C(j\Omega)$  apart for the rescaling of the angular frequency axis. It is noteworthy also the change in the measurement unit for the angular frequencies. In the discrete domain the angular frequency is an angle in the  $z$ -domain and thus it is measured in radians or degrees. In the continuous domain instead the measurement unit of the angular frequency is the ratio between the measurement units of an angle and of the sampling period. In this example since the sampling is in the time domain the angular frequency  $\Omega$  is measured in rad/s.

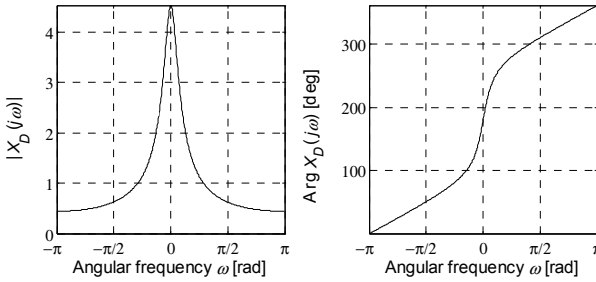


Figure 7.33. Magnitude and phase of the Fourier transform  $X_D(e^{j\omega})$  of the sequence  $x(n)$  obtained sampling the function  $g(t) = e^{-2t}$  with a sampling period  $T = 0.1$  s

The intuitive result obtained in the previous example that the Fourier transform  $X_C(j\Omega)$  converge to  $G(j\Omega)$  when the sampling period  $T \rightarrow 0$ , can be formalized and generalized to the Laplace transforms  $X_C(s)$  and  $G(s)$  considering the definition of the Laplace transform of a function:

$$G(s) = \int_{-\infty}^{+\infty} g(t)e^{-st} dt \tag{7.59}$$

By definition the integral in (7.59) is the limit:

$$G(s) = \lim_{T \rightarrow 0} \sum_{n=-\infty}^{+\infty} g(nT)e^{-snT} T \tag{7.60}$$

Comparing the limit (7.60) with the definition (7.52b) it results the following relationship between the Laplace transforms  $X_C(s)$  and  $G(s)$ :

$$G(s) = \lim_{T \rightarrow 0} TX_C(s) \tag{7.61}$$

i.e. as the sampling period  $T$  decreases the Laplace transform of the samples sequence  $X_C(s)$  converge the Laplace transform  $G(s)$  of the sampled function  $g(t)$  scaled by the sample period  $T$ . Obviously the result (7.61) can be considered for the Fourier transforms as well:

$$G(j\Omega) = \lim_{T \rightarrow 0} TX_C(j\Omega) \tag{7.62}$$

Figure 7.34 shows the convergence (7.62) for the function of example 1, sampled with a sampling period  $T$  ranging from 0.1 s to 0.9 s. The periodicity of the Fourier transforms  $X_C(j\Omega)$  of the samples sequences is evident in this figure. The period of each transformation  $X_C(j\Omega)$  is  $2\pi/T$ .

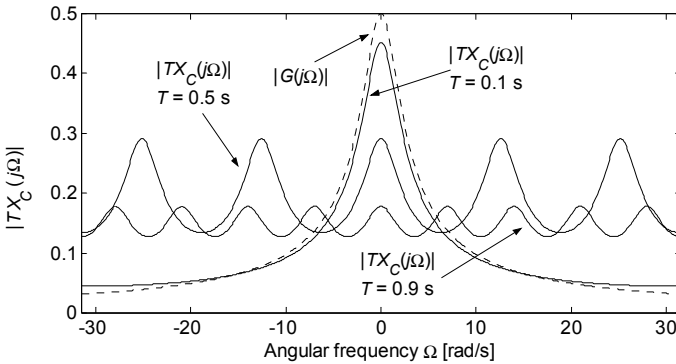


Figure 7.34. Comparison between the magnitude of the Fourier transform  $TX_C(j\Omega)$  and  $G(j\Omega)$  with sampling periods  $T$  of 0.1 s, 0.5 s and 0.9 s

## FINITE IMPULSE RESPONSE (FIR) FILTERS

*N*-DOMAIN REPRESENTATION OF FIR FILTERS.

As the name says a finite impulse response (FIR) filter is a linear shift invariant filter with an impulse response sequence  $h(n)$  of finite length  $q+1 < +\infty$ :

$$\{h(n)\} = \{h_0\delta(n) + h_1\delta(n-1) + \dots + h_q\delta(n-q)\} \quad (8.1)$$

The output sequence  $\{y(n)\}$  to an arbitrary input sequence  $\{u(n)\}$  is given by the convolution product between the impulse response and the input sequence:

$$y(n) = h(n) * u(n) = \sum_{k=0}^q h(k)u(n-k) \quad (8.2)$$

The block diagram in figure 8.1 shows the  $n$ -domain implementation of the FIR filter resulting directly from equations (8.2).

By direct comparison of equation (8.2) with the difference equation (7.41) defining a general linear discrete filter, it result that a FIR filter is a particular discrete filter with  $M=q$  and  $N=0$ . Being  $M>N$ , the filter order is  $q$ .

The state vector for this type of filter, according to (7.39), is given by the  $q$  values of the input sequence  $\{u\}$  preceding the current step  $n$ :

$$\mathbf{x}(n) = [u(n-1) \quad u(n-1) \quad \dots \quad u(n-q)]^T \quad (8.3)$$

**Example 1:** A first example of a FIR filter is the moving average filter, which just compute the average of the last  $q+1$  input samples:

$$y(n) = \frac{1}{q+1} \sum_{k=0}^q u(n-k)$$

Comparing this equation with (8.2) it follows that the impulse response for the moving average filter of order  $q$  is:

$$\{h(n)\} = \left\{ \frac{1}{q+1} \delta(n) + \frac{1}{q+1} \delta(n-1) + \dots + \frac{1}{q+1} \delta(n-q) \right\}$$

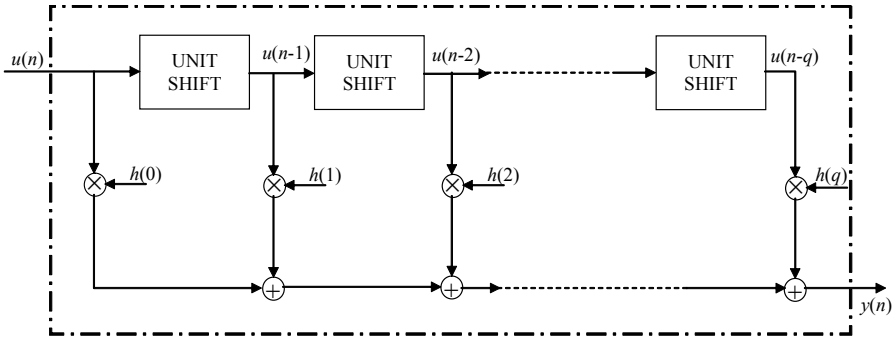


Figure 8.1.  $n$ -domain block diagram of a FIR filter

Z-DOMAIN REPRESENTATION OF FIR FILTERS.

The transfer function of a  $q^{th}$  order FIR filter is given by the  $z$ -transform of its impulse response (8.1):

$$H(z) = \sum_{k=0}^q h(k)z^{-k} = \frac{h(0)z^q + h(1)z^{q-1} + \dots + h(q)}{z^q} \tag{8.4}$$

The  $q$  poles of a  $q^{th}$  order FIR filter are in the origin of the  $z$ -plane. The filter has also  $q$  zeros given by the roots of the polynomial at numerator of (8.4).

The frequency response of the filter can be derived from the transfer function expression for  $z=e^{j\omega}$ .

$$H(e^{j\omega}) = \sum_{k=0}^q h(k)e^{-jk\omega} = \frac{h(0)e^{jq\omega} + h(1)e^{j(q-1)\omega} + \dots + h(q)}{e^{jq\omega}} \tag{8.5}$$

Since the frequency response  $H(e^{j\omega})$  is a periodic function, with period  $2\pi$ , the  $q+1$  filter coefficients  $h(k)$ , according to (8.5), form the Fourier series expansion of  $H(e^{j\omega})$ .

An important property of FIR filters is that they can approximate any stable and causal linear discrete filter. Considering the frequency response of a general linear discrete filter, being a  $2\pi$  period function of  $\omega$ , can always be expressed as an infinite Fourier series:

$$H(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} \alpha_k e^{jk\omega} \tag{8.6}$$

with

$$\alpha_k = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(e^{j\omega}) e^{-jk\omega} d\omega \tag{8.7}$$

Since the transfer function  $H(e^{j\omega})$  is the Fourier transform of the discrete filter impulse response  $h(n)$ :

$$H(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} h(k) e^{-jk\omega} \tag{8.8}$$

comparing, term-by-term, (8.6) with (8.8) it results that the coefficients  $\alpha_k$  of the series (8.6) form the filter impulse response sequence  $h(n)$ :

$$\{h(n)\} = \{ \dots \alpha_1 \delta(n+1) \quad h_0 \delta(n) \quad \alpha_{-1} \delta(n-1) \dots \} \tag{8.9}$$

If the discrete filter is causal the coefficients  $\alpha_k$  in (8.9) are zero for  $k > 0$ , reducing (8.6) to:

$$H(e^{j\omega}) = \sum_{k=-\infty}^0 \alpha_k e^{jk\omega} = \sum_{k=0}^{+\infty} h(k) e^{-jk\omega} \tag{8.10}$$

If the filter is stable  $h(k)$  tends to zero as  $k$  tends to infinity and, consequently, the function  $H(e^{j\omega})$  can be approximated by the series (8.10) with a finite number  $q$  of terms:

$$H(e^{j\omega}) \cong \sum_{k=0}^q h(k) e^{-jk\omega} \tag{8.11}$$

By direct comparison of (8.10) with (8.5) it is immediate to see that the generic discrete filter transfer function  $H(e^{j\omega})$  can be approximated by a FIR filter of order  $q$ , as far as  $q$  is large enough for the approximation (8.11).

**Example 1:** The transfer function of the moving average FIR filter is given by

$$\begin{aligned}
 H(z) &= \frac{1}{q+1} \sum_{k=0}^q z^{-k} = \frac{1}{q+1} \left( \sum_{k=0}^{\infty} z^{-k} - \sum_{k=q+1}^{\infty} z^{-k} \right) = \\
 &= \frac{1}{q+1} \left( \frac{1}{1-z^{-1}} - \frac{z^{-(q+1)}}{1-z^{-1}} \right) = \frac{1}{q+1} \frac{1-z^{-(q+1)}}{1-z^{-1}} = \frac{1}{q+1} \frac{z^{q+1}-1}{z^q(z-1)}
 \end{aligned}$$

The moving average filter has  $q$  poles in the origin and one pole at  $z=1$ . The  $q+1$  zeros of the filter are on the unit circle at  $z=e^{jk2\pi/(q+1)}$  with  $k=0,1,\dots,q$ . The pole and zero at  $z=1$ , for  $k=0$ , compensate with each other and the *ROC* of  $H(z)$  is the whole complex plane  $|z|>0$ .

The magnitude response and the zero-pole diagram of a 5<sup>th</sup> order moving average FIR filter is shown in figure 8.2.

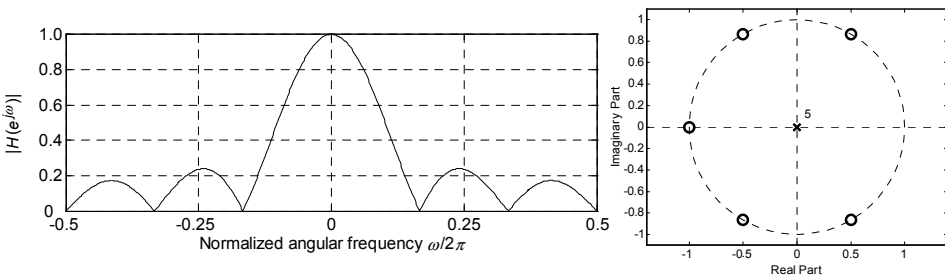


Figure 8.2. Magnitude response and zero-pole diagram of a 5<sup>th</sup> order moving average filter

**Example 2:** Other simple FIR filters are the comb filters, which output at step  $n$  is simply either the sum or the difference between  $u(n)$  and  $u(n-N)$ . The transfer functions for these filters are given by:

$$H(z) = 1 \pm z^{-N} = \frac{z^N \pm 1}{z^N}$$

The comb filters, as the moving average filter, have  $N$  poles in the origin. The comb filter with transfer function  $H(z)=1+z^{-N}$  has  $N$  zeros at  $z=e^{jk2\pi/N}$  with  $k=1,2,\dots,N$ . Its magnitude frequency response and zero-pole diagram are shown in figure 8.3.

The zeros of the comb filter with transfer function  $H(z)=1-z^{-N}$  are at  $z=e^{jk2\pi/N}$  with  $k=0,2,\dots,N-1$ . Its magnitude frequency response and zero-pole diagram are shown in figure 8.4.



The comb filter with  $H(z)=1-z^{-N}$ , having one zero at  $z=1$ , has a zero DC gain, while the comb filter with  $H(z)=1+z^{-N}$ , having no zero at  $z=1$ , has finite DC gain.

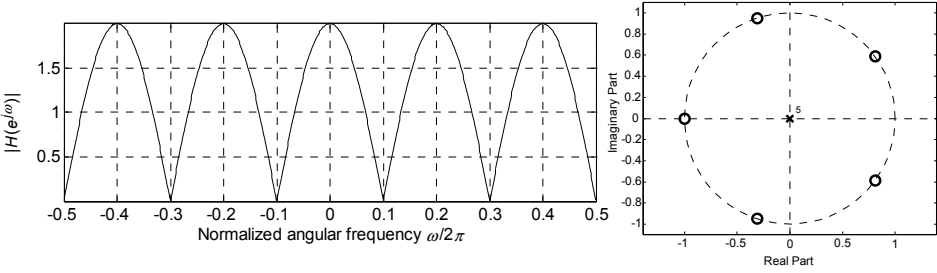


Figure 8.3. Magnitude frequency response and zero-pole diagram for a comb filter with transfer function  $H(z)=1+z^{-N}$  and  $N=5$

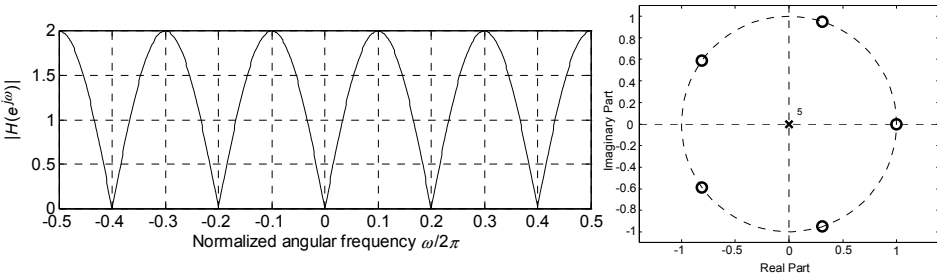


Figure 8.4. Magnitude frequency response and zero-pole diagram for a comb filter with transfer function  $H(z)=1-z^{-N}$  and  $N=5$

**Example 3:** A particular comb FIR filter is the cosine filter defined by the following difference equation:

$$y(n) = \frac{1}{2} [u(n) + u(n-1)]$$

The transfer function for the cosine filter is:

$$H(z) = \frac{1}{2} (1 + z^{-1}) = \frac{z+1}{2z}$$

with a pole in  $z=0$  and a zero at  $z=j$ , as shown in figure 8.5 together with the filter impulse response  $h(n)$ .

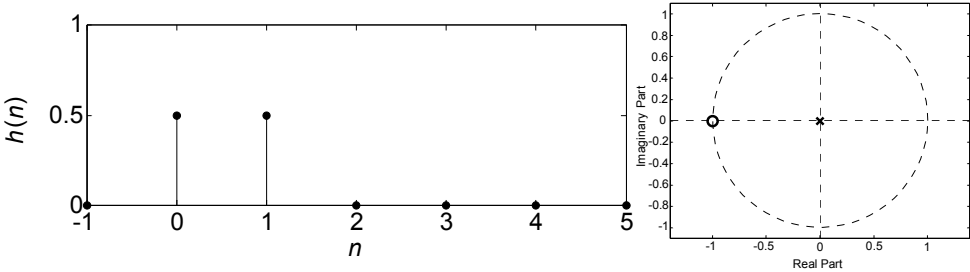


Figure 8.5. Impulse response and pole-zero diagram of a cosine filter

Its Fourier transform is given by

$$H(e^{j\omega}) = \frac{1}{2}(1 + e^{-j\omega}) = e^{-j\omega/2} \frac{e^{j\omega/2} + e^{-j\omega/2}}{2} = e^{-j\omega/2} \cos(\omega/2)$$

which amplitude  $|H(e^{j\omega})| = \cos(\omega/2)$  gives the name to this filter.

The phase frequency response of the cosine filter is linear versus  $\omega$  and expressed by  $\angle H(e^{j\omega}) = -\omega/2$ . Figure 8.6 shows both the amplitude and phase response of the filter. The cosine filter behaves as a low pass filter; the effect of the zero at  $\omega = \pi$  is to select components with angular frequency  $\omega = \pi$  and allows to pass low angular frequencies up to DC.

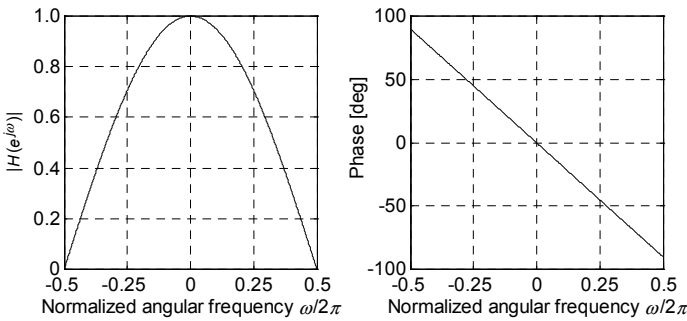


Figure 8.6. Amplitude and phase response of a cosine filter

We have seen that the cosine filter has got one zero and one pole and that the zero in  $z=-1$  is responsible for the its low pass filter behavior. If we consider now a FIR filter with a transfer function with two zeros in  $z=-1$  and two poles in  $z=0$ , i.e. of the type

$$H(z) = \frac{(z+1)^2}{4z^2} = \frac{1}{4} (1 + 2z^{-1} + z^{-2})$$

we should obtain a more selective low pass filter at  $\omega=\pi$ . The Fourier transform for this filter is expressed by:

$$H(e^{j\omega}) = \frac{1}{4} (1 + 2e^{-j\omega} + e^{-2j\omega}) = \frac{1}{2} e^{-j\omega} \left( 1 + \frac{e^{j\omega} + e^{-j\omega}}{2} \right) = \frac{1}{2} e^{-j\omega} [1 + \cos(\omega)]$$

Because of the amplitude response this type of FIR filter is called raised cosine filter. Figure 8.7 shows the amplitude and the linear phase frequency responses of the raised cosine filter. Comparing this picture with figure 8.6, reporting the frequency response of a cosine filter, it is evident the increased selective low pass filtering action of the raised cosine filter obtained just adding a further zero at  $z=-1$ , i.e. increasing the filter length.

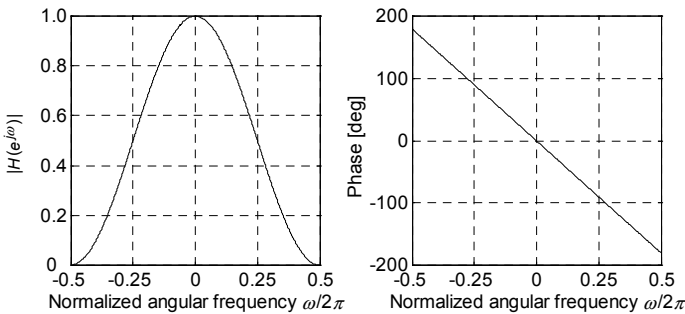


Figure 8.7. Frequency response of a raised cosine FIR filter

From the transfer function it is simple to derive the impulse response of this filter and its pole-zero diagram, as reported in figure 8.8. The transfer function  $H(z)$  define also the difference equation of this filter:

$$y(n) = \frac{1}{4} [u(n) + 2u(n-1) + u(n-2)]$$

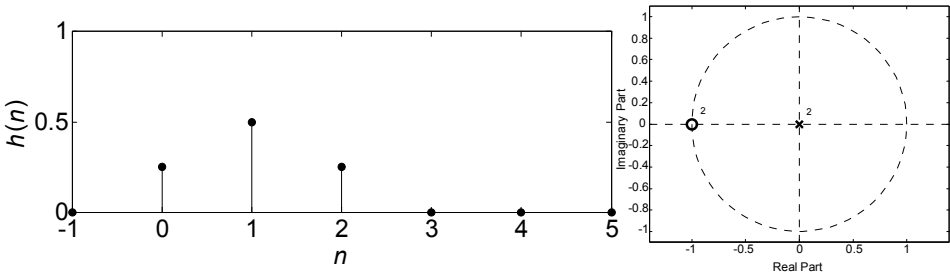


Figure 8.8. Impulse response and pole-zero diagram of a raised cosine FIR filter

LINEAR PHASE FIR FILTERS

In digital signal processing a distortionless filter is often required. This implies that, when the spectrum of the filter input sequence  $\{u(n)\}$  lays inside the filter pass-band, the filter output sequence  $\{y(n)\}$  is just proportional to a translated version of  $\{u(n)\}$ :

$$y(n) = G \cdot u(n-N) \tag{8.12}$$

where  $G$  is the filter gain in the pass-band and  $N$  is the translation step shift.

The corresponding requirements in the frequency domain can be gained tacking the Fourier transform of (8.12):

$$Y(e^{j\omega}) = G \cdot e^{-jN\omega} U(e^{j\omega}) \tag{8.13}$$

Thus, for distorsionless filters, the transfer function in the pass-band must be expressed by:

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{U(e^{j\omega})} = G e^{-jN\omega} \tag{8.14}$$

More generally  $|H(e^{j\omega})|$  can be expressed as a positive real function  $G(e^{j\omega})$  with:

- $G(e^{j\omega}) = 1$ , in the filter pass-band;
- $G(e^{j\omega}) = 0$ , in the filter stop-band;
- $0 < G(e^{j\omega}) < 1$ , in the filter transition-band.

The filter transfer function (8.14) becomes then:

$$H(e^{j\omega}) = G(e^{j\omega}) \cdot e^{-jN\omega} \tag{8.15}$$

This implies that for no distortion at the output of the linear shift invariant filter two requirements must be satisfied in the pass-band:

1. the amplitude frequency response is flat, i.e.:

$$|H(e^{j\omega})| = 1, \text{ in the pass-band} \quad (8.16a)$$

2. the phase frequency response is a linear function of the angular frequency  $\omega$ , that is:

$$\theta(e^{j\omega}) = \angle H(e^{j\omega}) = -N\omega, \text{ in the pass-band} \quad (8.16b)$$

The impulse response of a distortionless filter is given by the inverse Fourier transform of  $H(e^{j\omega})$  in (8.15):

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} G(e^{j\omega}) e^{-j\phi(e^{j\omega})} e^{jn\omega} d\omega = \frac{1}{2\pi} \int_{-\pi}^{+\pi} G(e^{j\omega}) e^{j(n-N)\omega} d\omega \quad (8.17)$$

If the real  $2\pi$  periodic function  $G(e^{j\omega})$  is split into its even  $G_E(e^{j\omega})$  and odd  $G_O(e^{j\omega})$  parts:

$$\begin{aligned} G_E(e^{j\omega}) &= \frac{G(e^{j\omega}) + G(e^{-j\omega})}{2} \\ G_O(e^{j\omega}) &= \frac{G(e^{j\omega}) - G(e^{-j\omega})}{2} \end{aligned} \quad (8.18)$$

the impulse response becomes:

$$h(n) = \frac{1}{\pi} \left[ \int_0^{+\pi} G_E(e^{j\omega}) \cos[(n-N)\omega] d\omega + j \int_0^{+\pi} G_O(e^{j\omega}) \sin[(n-N)\omega] d\omega \right] \quad (8.19)$$

If  $h(n)$  must be real  $G(e^{j\omega})$  must be an even function, i.e.:

$$G(e^{j\omega}) = G_E(e^{j\omega}), G_O(e^{j\omega}) = 0, \quad (8.20)$$

resulting in the following expression for the impulse response:

$$h(n) = \frac{1}{\pi} \int_0^{+\pi} G(e^{j\omega}) \cos[(n-N)\omega] d\omega \quad (8.21)$$

Hence from relationship (8.20) we have learnt that in order to design a discrete linear phase filter we must specify an even amplitude frequency response; once defined the function  $G(e^{j\omega})$  the filter coefficients can be computed using (8.21).

For an arbitrary amplitude response the number of the resulting filter coefficients  $h_n$  will not be finite. However in the previous paragraph it was shown with equation (8.11) that any stable and causal linear discrete filter can be approximated by a FIR filter just considering the first  $q+1$  filter coefficient  $h(n)$  in (8.21).

For example figure 8.9 shows the amplitude and phase response of a 10<sup>th</sup> order FIR filter with shift  $N=1$ , approximating an ideal low pass filter with a cut off angular frequency of  $\pi/2$ . The effect of having limited the number of filter coefficients (8.21) in the FIR approximation appears clearly in the plot of figure 8.9: both the amplitude and phase responses are far from the ideal one, shown in figure as dotted line.

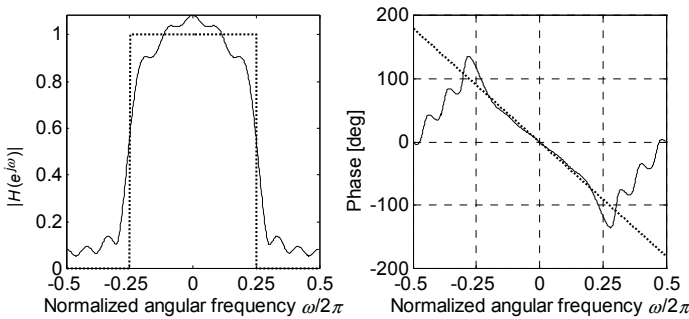


Figure 8.9. Frequency response for 10<sup>th</sup> order FIR filter, with a step shift of  $N=1$ , approximating an ideal low pass linear phase filter

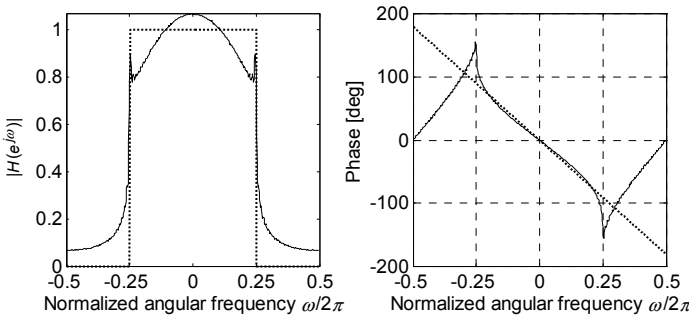


Figure 8.10. Frequency response for a 100<sup>th</sup> order FIR filter, with a step shift of  $N=1$ , approximating an ideal low pass linear phase filter

In figure 8.10 is shown the result obtained increasing the order of the FIR filter. The figure has been obtained rising the number of coefficients in the FIR approximation from 11 to 101. Although the number of coefficients in the FIR filter approximation of (8.11) is largely increased, in comparison with the previous approximation shown in figure 8.9, the situation is not improved much. Both the amplitude and phase responses are not converging towards the ideal low-pass linear phase frequency response.

The reasons for this behaviour can be understood considering that the finite length approximation (8.11) is valid only for stable and causal linear filters, but from expression (8.21) results that the impulse response for the ideal distortionless filter must be an even sequence of  $n$  and therefore cannot be causal. Looking at the impulse responses  $\{h(n)\}$  for the two FIR filters shown in figure 8.11, since the FIR approximation is causal, both the impulse responses are forced to zero for  $n < 0$ , removing all those coefficients  $\{h(n)\}$  in (8.21) with  $n < 0$ . Thus if the step shift  $N$  is undersized, coefficients with large values can be removed as well from the FIR expansion limiting the effectiveness of the finite length FIR approximation.

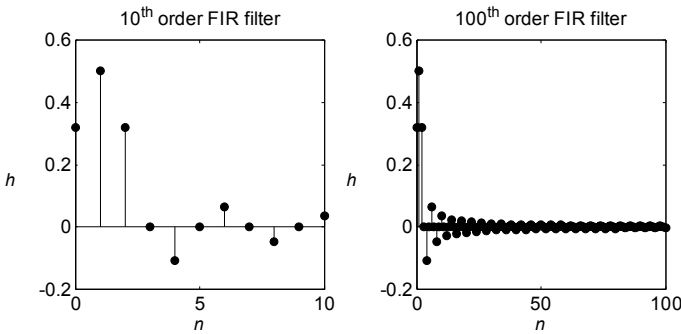


Figure 8.11. Impulse responses of the 10<sup>th</sup> and 100<sup>th</sup> order FIR filters whose frequency responses are shown in figure 8.9 and 8.10

In order to preserve in the finite length expansion (8.11) the coefficients with largest absolute value  $|h_k|$ , the shift  $N$  cannot be arbitrary. In particular if the even structure of the sequence  $\{h(n)\}$  is to be maintained besides the FIR truncation, the shift in (8.21) must be  $N=q/2$ . This shift assumes an integer value only when the filter order  $q$  is even. Figure 8.12 shows the two impulse responses for both the 10<sup>th</sup> and 100<sup>th</sup> order FIR filters, when  $N$  is equal to 5 and 50 respectively. In this cases from the picture it can be seen that the impulse response, although of finite length, retain the even property of  $\{h(n)\}$  around the shift  $N$ , i.e. it results:

$$h(n) = h(q-n) \tag{8.22}$$

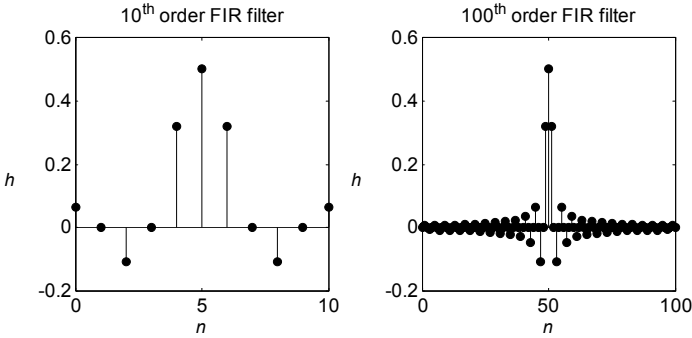


Figure 8.12. Impulse responses for two FIR filter of order  $q$  equal to 10 and 100, when the step shift  $N$  is  $q/2$

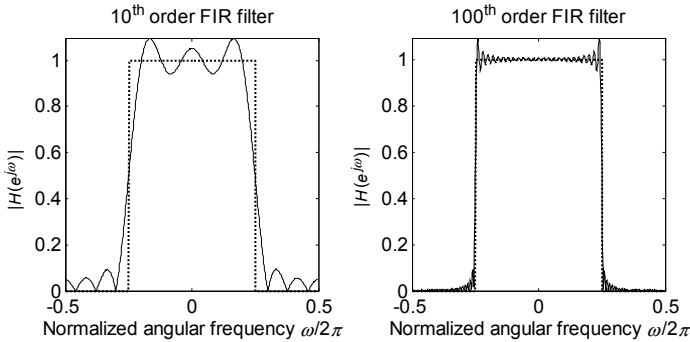


Figure 8.13. Amplitude frequency responses for the two FIR filters whose impulse responses are shown in figure 8.12

Figure 8.13 shows the amplitude frequency response for the two FIR filters. Comparing this picture with the amplitude frequency responses shown in figure 8.9 and 8.10 it is clear the better convergence behaviour towards the ideal low-pass filter. Furthermore the increased order of the filter produces a frequency response convergent towards the ideal one.

Even better is the phase frequency response since this is not approximating a linear behaviour, but if condition (8.22) is sustained the phase response is exactly linear. This can be verified just replacing (8.22) into (8.11) and factorizing. The factorization is slightly different for  $q$  even and odd. In case of  $q$  even it results:



$$\begin{aligned}
 H(e^{j\omega}) &\cong \sum_{k=0}^q h(k)e^{-jk\omega} = \sum_{k=0}^{\frac{q-1}{2}} h(k)e^{-jk\omega} + \sum_{k=q/2+1}^q h(k)e^{-jk\omega} + h\left(\frac{q}{2}\right)e^{-jN\omega} = \\
 &= \sum_{k=0}^{\frac{q-1}{2}} [h(k)e^{-jk\omega} + h(q-k)e^{-j(q-k)\omega}] + h\left(\frac{q}{2}\right)e^{-jN\omega} = \\
 &= e^{-j\frac{q}{2}\omega} \left\{ \sum_{k=0}^{\frac{q-1}{2}} 2h(k)\cos\left[\left(\frac{q}{2}-k\right)\omega\right] + h\left(\frac{q}{2}\right) \right\}
 \end{aligned} \tag{8.23a}$$

If  $q$  is odd instead the factorisation gives:

$$\begin{aligned}
 H(e^{j\omega}) &\cong \sum_{k=0}^q h(k)e^{-jk\omega} = \sum_{k=0}^{\frac{q-1}{2}} h(k)e^{-jk\omega} + \sum_{k=\frac{q+1}{2}}^q h(k)e^{-jk\omega} = \\
 &= \sum_{k=0}^{\frac{q-1}{2}} [h(k)e^{-jk\omega} + h(q-k)e^{-j(q-k)\omega}] = \\
 &= e^{-j\frac{q}{2}\omega} \left\{ \sum_{k=0}^{\frac{q-1}{2}} 2h(k)\cos\left[\left(\frac{q}{2}-k\right)\omega\right] \right\}
 \end{aligned} \tag{8.23b}$$

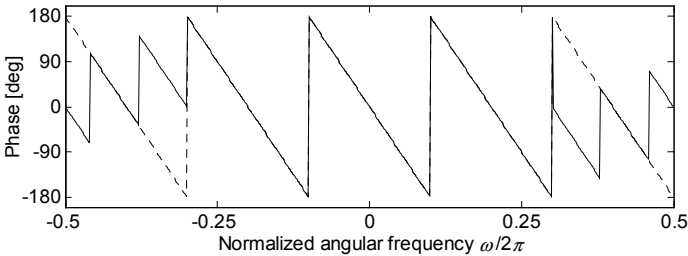


Figure 8.14. Phase response of a 10<sup>th</sup> order linear phase FIR filter

Since the coefficients  $h(k)$  are real, the terms in braces are real as well and the phase of the frequency response (8.23) is exactly linear. The step shift of the FIR filter as said before is  $q/2$  and it is integer only when the filter order  $q$  is even. Figure 8.14 reports the phase response for the 10<sup>th</sup> order FIR approximation whose impulse and amplitude

responses are shown in the previous figures 8.12 and 8.13. It can be seen that in the pass-band the FIR approximation follows exactly the dotted line, representing the phase response of the ideal low-pass filter.

FIR FILTER SYNTHESIS BY FOURIER SERIES EXPANSION.

The previous paragraph suggests a simple way for the synthesis of a  $q^{\text{th}}$  order FIR filter. Firstly the theoretical amplitude frequency response  $|H_{SP}(e^{j\omega})|$  is defined. Secondly the theoretical impulse response  $h_{SP}(n)$  of the filter is specified by the inverse Fourier transform of  $H_{SP}(e^{j\omega})$ , assuming linear phase response  $-q\omega/2$ , accordingly with (8.21):

$$h_{SP}(n) = \frac{1}{\pi} \int_0^{+\pi} |H_{SP}(e^{j\omega})| \cos\left[\left(n - \frac{q}{2}\right)\omega\right] d\omega \tag{8.24}$$

Thirdly, since generally the impulse response  $h_{SP}(n)$  will not be of finite length and causal, the impulse response  $h(n)$  of the  $q^{\text{th}}$  order FIR filter, approximating the amplitude response  $|H_{SP}|$ , is derived simply limiting  $h(n)$  only to the first  $q+1$  terms of  $h_{SP}(n)$ :

$$\begin{aligned} h(n) &= 0 \quad \forall n \notin 0 \dots q \\ h(n) &= h_{SP}(n) \quad \forall n \in 0 \dots q \end{aligned} \tag{8.25}$$

This last operation is equivalent to the product of the infinite length sequence  $h_{SP}(n)$  with a rectangular weighting window sequence  $w(n)$ :

$$h(n) = h_{SP}(n) \cdot w(n) \tag{8.26}$$

where the rectangular window  $w(n)$  is defined as:

$$w(n) = \begin{cases} 1 & \forall n \in 0 \dots q \\ 0 & \forall n \notin 0 \dots q \end{cases} \tag{8.27}$$

Equation (8.26) is equivalent, in the frequency domain, to the complex convolution between  $H_{SP}(e^{j\omega})$  and the Fourier transform  $W(e^{j\omega})$  of the sequence  $w(n)$ :

$$H(e^{j\omega}) = H_{SP}(e^{j\omega}) * W(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H_{SP}(e^{j\xi}) \cdot W(e^{j(\omega-\xi)}) d\xi \tag{8.28}$$

The Fourier transform of the rectangular windowing sequence  $w(n)$  can be determined from its  $z$ -transform  $W(z)$ :

$$\begin{aligned}
 W(z) &= \sum_{k=0}^q w(k) \cdot z^{-k} = \sum_{k=0}^{\infty} z^{-k} - \sum_{k=q+1}^{\infty} z^{-k} = \\
 &= \frac{1}{1-z^{-1}} - \frac{z^{-(q+1)}}{1-z^{-1}} = \frac{1-z^{-(q+1)}}{1-z^{-1}}
 \end{aligned}
 \tag{8.29}$$

by replacing in (8.29) the complex variable  $z$  with  $e^{j\omega}$ .

$$W(e^{j\omega}) = \frac{1-e^{-j(q+1)\omega}}{1-e^{-j\omega}} = e^{-j\frac{q}{2}\omega} \frac{\sin\left(\frac{q+1}{2}\omega\right)}{\sin\left(\frac{1}{2}\omega\right)}
 \tag{8.30}$$

Figure 8.15 shows the amplitude of (8.30) setting  $q$  to 5 and 50 respectively. The function  $|W(e^{j\omega})|$  is composed by one *main lobe* in the region  $-\frac{2\pi}{q+1} \leq \omega \leq \frac{2\pi}{q+1}$  and  $q-1$  decreasing oscillatory *side lobes* for higher frequencies. Increasing the order  $q$  of the filter will increase the amplitudes of both the main and the side lobes, while decreasing their width and leaving unchanged the subtended area.

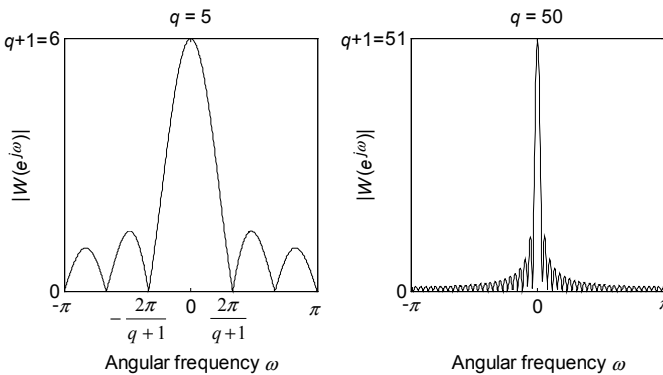


Figure 8.15. Amplitude of  $W(e^{j\omega})$  for two values (5 and 50) of the FIR filter order  $q$

The convolution product (8.28) will certainly change the shape of the actual FIR transfer function  $H(e^{j\omega})$  with respect to the specified transfer function  $H_{SP}(e^{j\omega})$ . The distortion effect due to the sliding window  $W(e^{j\omega})$  in (8.28) can be understood considering for example the convolution product with the low-pass response  $H_{SP}(e^{j\omega})$  shown in figure 8.16. The result is a smoothed amplitude response  $H(e^{j\omega})$ , where the

effect of the main lobe in  $W(e^{j\omega})$  is to limit the steepness of the FIR filter response at the cut-off frequency, while the side lobes introduce ripples in  $H(e^{j\omega})$ . Increasing the filter order will decrease the main lobe width, allowing a sharper transition at the cut-off frequency. The ripple amplitude instead will not decrease since the area underneath each side lobe remain constant as the filter order increases (Gibb's phenomenon).

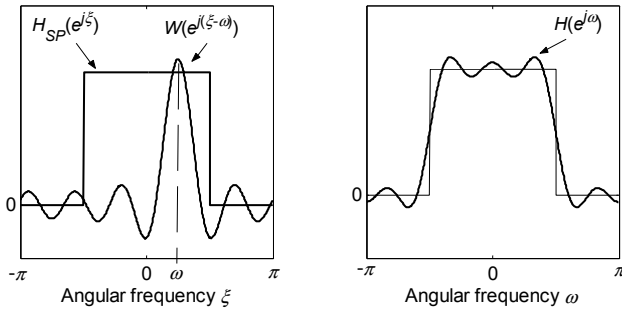


Figure 8.16. Effect of the convolution between  $H_{SP}(e^{j\omega})$  with  $W(e^{j\omega})$

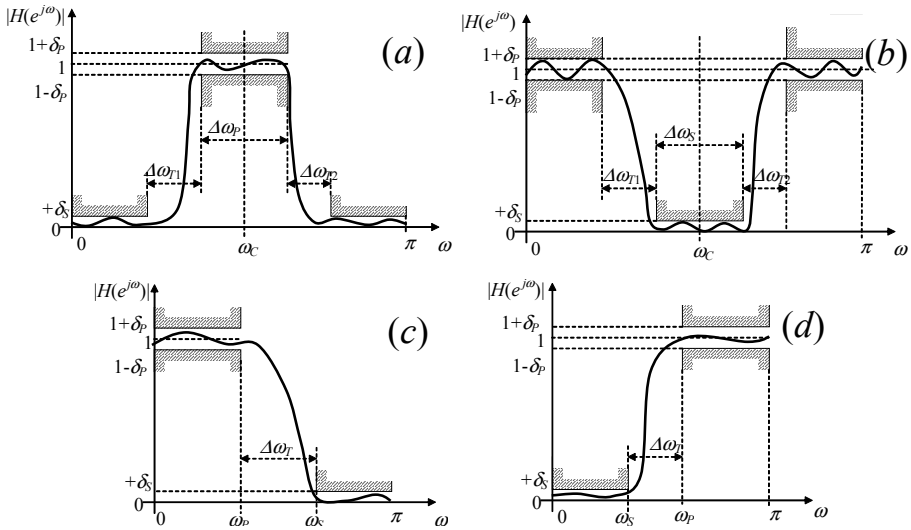


Figure 8.17. Typical design masks in the frequency domain for (a) pass-band, (b) stop-band, (c) low-pass and (d) high-pass filters

The specification in the frequency domain of the actual FIR filter response is usually given considering one of the masks shown in figure 8.17 for band-pass, stop-band, low-pass and high-pass filters with center frequency  $\omega_c$ . Each mask defines one or more stop bands  $\Delta\omega_s$ , pass bands  $\Delta\omega_p$  and transition bands  $\Delta\omega_t$ . In the pass and stop bands some ripple,  $\delta_p$  and  $\delta_s$  respectively, is allowed. The filter design phase consists in determining the coefficients of the shortest FIR filter meeting the filter specifications.

Using a rectangular window, and increasing the filter order, the filter specifications can be met only in the transition-band. As far as the ripple in the pass and stop bands is concerned the filter order will leave its level unchanged, because of the Gibb's phenomenon; one possibility for reducing the ripple is thus by changing the window shape with another one whose spectra introduces less ripple than that of the rectangular window. Figure 8.18 shows a comparison between the rectangular window with the triangular window, which avoids the abrupt truncation caused by the rectangular window of the specified infinite length impulse response  $h_{SP}(n)$ . In this case the side lobes are 24 dB down the main lobe, much less than the 13 dB of the rectangular window. The consequence of a reduced amplitude of the side lobes is an increase in the transition-band due to a double width for the main lobe, as shown in figure 8.18.

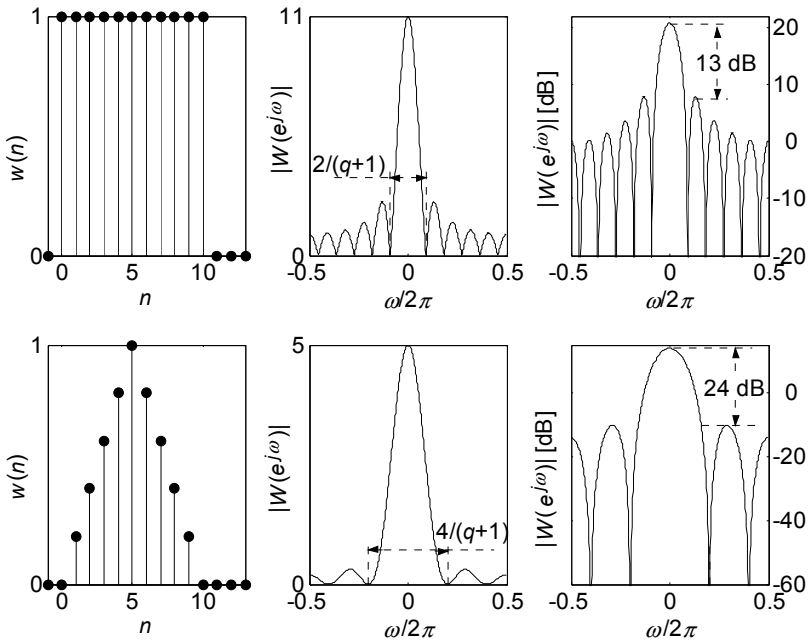


Figure 8.18. The main lobe width for triangular windows is double than for rectangular windows, but the side lobe amplitude is lower (24 dB vs. 13 dB)

Windows shapes, other than the rectangular and triangular, have been proposed by various authors. They are characterized by different values of the main lobe width and side lobes amplitude. The most important and frequently used are:

- The *rectangular window*:

$$w(n) = \begin{cases} 1 & 0 \leq n \leq q \\ 0 & \text{otherwise} \end{cases} \quad (8.31)$$

- The *Bartlett (triangular) window*:

$$w(n) = \begin{cases} \frac{2n}{q} & 0 \leq n \leq \frac{q}{2} \\ 2 - \frac{2n}{q} & \frac{q}{2} < n \leq q \end{cases} \quad (8.32)$$

- The *Hanning window*:

$$w(n) = \begin{cases} 0.5 - 0.5 \cos\left(\frac{2\pi n}{q}\right) & 0 \leq n \leq q \\ 0 & \text{otherwise} \end{cases} \quad (8.33)$$

- The *Hamming window*:

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{q}\right) & 0 \leq n \leq q \\ 0 & \text{otherwise} \end{cases} \quad (8.34)$$

- The *Blackman window*:

$$w(n) = \begin{cases} 0.42 - 0.5 \cos\left(\frac{2\pi n}{q}\right) + 0.08 \cos\left(\frac{4\pi n}{q}\right) & 0 \leq n \leq q \\ 0 & \text{otherwise} \end{cases} \quad (8.35)$$

A further window used for FIR filter design is the Kaiser window:

$$w(n) = \begin{cases} \frac{I_0\left(\beta\sqrt{1-\left[1-\frac{n}{q/2}\right]^2}\right)}{I_0(\beta)} & 0 \leq n \leq q \\ 0 & \text{otherwise} \end{cases} \quad (8.36)$$

where  $I_0(\cdot)$  is the 0<sup>th</sup> order modified Bessel function of the first kind, and  $\beta$  a parameter optimizing the threshold between the main lobe width and the side lobes amplitudes. Typical values of  $\beta$  are between 4 and 9.

Figure 8.19 shows the  $n$ -domain shape and  $\omega$ -domain spectra for these windows, while table 8.1 reports their main characteristics.

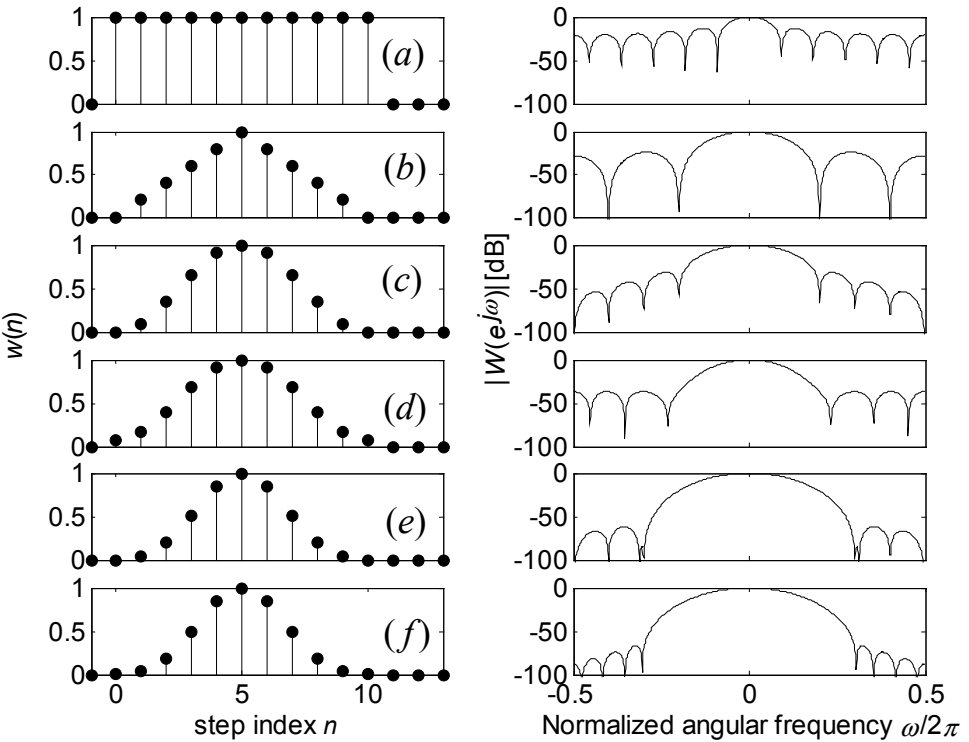


Figure 8.19. Comparison between the (a) uniform, (b) Bartlett, (c) Hanning, (d) Hamming, (e) Blackman and (f) Kaiser ( $\beta=9$ ) windows with length  $q=10$ , in both the  $n$ -domain shape and the  $\omega$ -domain spectra

Table 8.1

Window	Main lobe width	Side lobe amplitude
Rectangular	$4\pi/(q+1)$	-13 dB
Bartlett	$8\pi/(q+1)$	-24 dB
Hanning	$8\pi/(q+1)$	-32 dB
Hamming	$8\pi/(q+1)$	-37 dB
Blackman	$12\pi/(q+1)$	-62 dB

**Example:** For a low-pass FIR filter the following specifications in the frequency domain are given:

- Cut-off frequency:  $\omega_c = \pi/6$  rad
- Ripple:  $\delta_s = \delta_p = 0.02$
- Transition band:  $\Delta\omega_T = \pi/30$  rad

From (8.24) is simple to derive the infinite length impulse response for the specified filter with linear phase  $\angle H_{SP}(e^{j\omega}) = -q\omega/2$ :

$$h_{SP}(n) = \frac{1}{\pi} \int_0^{+\pi} |H_{SP}(e^{j\omega})| \cos\left[\left(n - \frac{q}{2}\right)\omega\right] d\omega = \frac{1}{\pi} \int_0^{+\frac{\pi}{6}} \cos\left[\left(n - \frac{q}{2}\right)\omega\right] d\omega = \frac{1}{6} \frac{\sin\left[\left(n - \frac{q}{2}\right)\frac{\pi}{6}\right]}{\left(n - \frac{q}{2}\right)\frac{\pi}{6}}$$

Figure 8.20 shows, together with the low-pass filter specification mask, the amplitude frequency response of a 20<sup>th</sup> order FIR filter obtained with a rectangular window and approximating the specified low-pass filter with a cut-off frequency at  $\pi/6$ . The FIR filter frequency response does not meet the specification because both the transition band and the ripple in the stop and pass bands are too wide.

Figure 8.21 shows the result obtained increasing the filter order from 20 to 35, still using a rectangular window. In this case the transition band meet the requirements, but since the amplitude of the side lobes does not decrease with filter order, the ripple amplitude is not decreased. In figure 8.22 is reported the amplitude frequency response computed with a 35<sup>th</sup> order FIR filter but designed using the Hamming window. The ripple this time is lower than  $\delta_p$  and  $\delta_s$ , but the transition band is wider than  $\Delta\omega_T$  because the wider main lobe of the Hamming window than the rectangular window.

By increasing the filter order from 35 to 80, still using the Hamming window, the filter amplitude response finally meets the specifications, as shown in figure 8.23. Figure 8.24 instead shows the response of a 80<sup>th</sup> order FIR filter designed using the rectangular window; again it is possible to see that the ripple has the same level as the 20<sup>th</sup> order FIR filter whose amplitude frequency response is shown in figure 8.20.



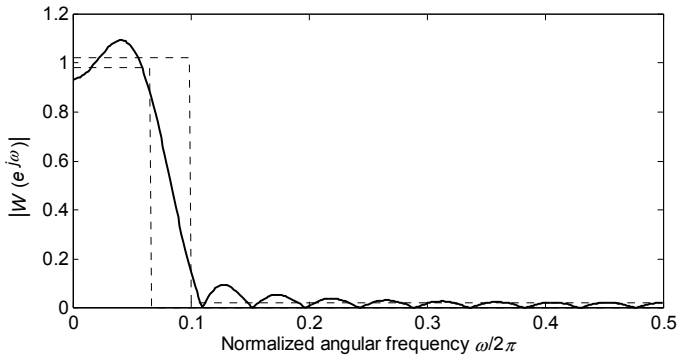


Figure 8.20. Amplitude response of a 20<sup>th</sup> order FIR filter with rectangular window

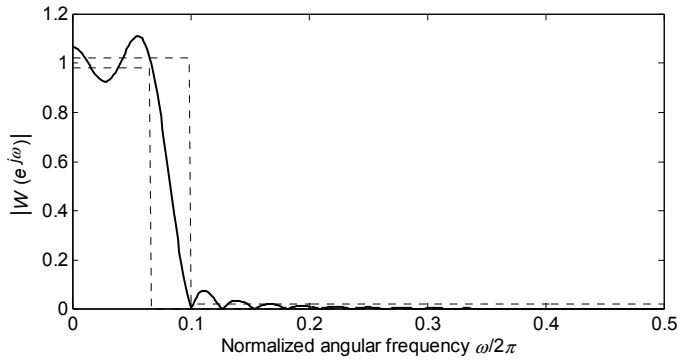


Figure 8.21. Amplitude response of a 35<sup>th</sup> order FIR filter with rectangular window

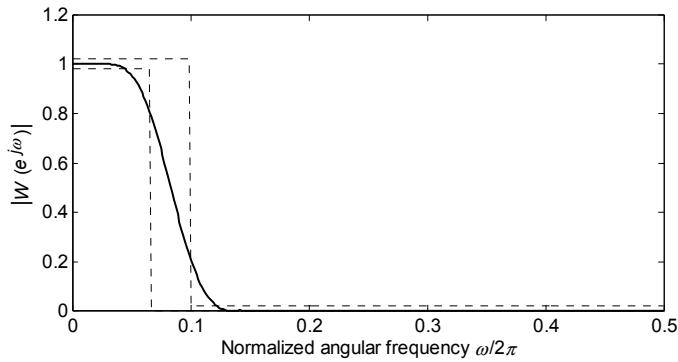


Figure 8.22. Amplitude response of a 35<sup>th</sup> order FIR filter with Hamming window

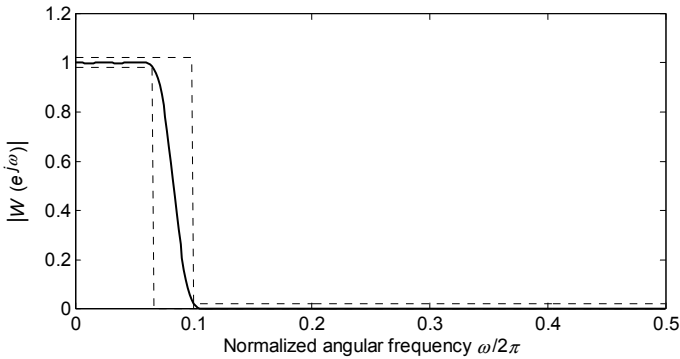


Figure 8.23. Amplitude response of a 80<sup>th</sup> order FIR filter with Hamming window

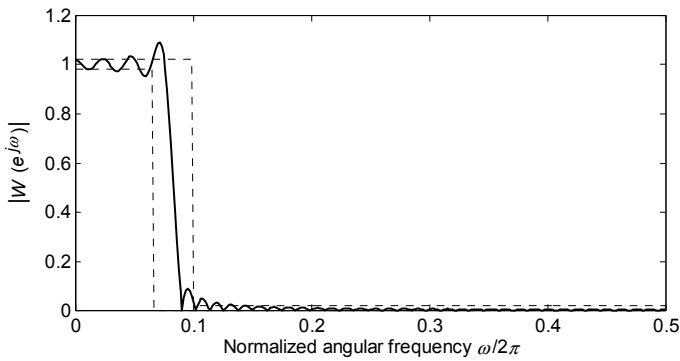


Figure 8.24. Amplitude response of a 80<sup>th</sup> order FIR filter with rectangular window

FIR FILTER SYNTHESIS BY THE FREQUENCY SAMPLING METHOD.

Another possibility for the design of FIR filters derives from the consideration that the discrete filter frequency response is a periodic function of  $\omega$  with period  $2\pi$ . According to the sampling theory of periodic functions, developed in chapter 3, the specified frequency response  $H_{SP}(e^{j\omega})$  can be defined by  $q+1$  samples, with  $q=2N$ , taken at frequency multiple of  $\omega_s = 2\pi/(q+1)$ . The frequency response  $H_{SP}(e^{j\omega})$  can be reconstructed from its  $q+1$  samples

$$\hat{H}_{SP}(k) = H_{SP}(e^{jk\omega_s}) \tag{8.37}$$

using the Shannon interpolation formula for periodic functions (3.47):

$$H_{SP}(e^{j\omega}) = \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \frac{\sin\left[\frac{2N+1}{2}\left(\omega - k\frac{2\pi}{2N+1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - k\frac{2\pi}{2N+1}\right)\right]} \tag{8.38}$$

The interpolation formula (8.38) will give the specified frequency response  $H_{SP}(e^{j\omega})$  only if its Fourier transform satisfies the Shannon theorem for periodic synchronous sampling. This is usually not true since the specified frequency response is a theoretical one and as such it is very sharp at the cut-off frequencies, resulting in a Fourier transform, which is the impulse response of the specified filter, very long lasting. The interpolation formula (8.38) will give then only an approximation  $H(e^{j\omega})$  of the function  $H_{SP}(e^{j\omega})$ :

$$H(e^{j\omega}) = \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \frac{\sin\left[\frac{2N+1}{2}\left(\omega - k\frac{2\pi}{2N+1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - k\frac{2\pi}{2N+1}\right)\right]} \tag{8.39}$$

The interpolation (8.39) has the property that at the sample frequency  $k\omega_s$  the specified frequency response assumes the same values of  $H(e^{j\omega})$ :

$$H_{SP}(e^{jk\omega_s}) = H(e^{jk\omega_s}) \quad \text{with } -N \leq k \leq N \tag{8.40}$$

Figure 8.25 shows an example of the application of (8.40) to an ideal low-pass filter with cut-off frequency at  $\omega_C = \pi/2$  and  $q=10$ . From this picture it results clearly how  $H(e^{j\omega})$  does not match  $H_{SP}(e^{j\omega})$  but at the sampling frequencies  $k\omega_s$ .

Taking into account that:

$$\frac{\sin\left(\frac{2N+1}{2}\omega\right)}{\sin\left(\frac{1}{2}\omega\right)} = \sum_{m=-N}^N e^{jm\omega} \tag{8.41}$$

equation (8.39) can be rewritten as:

$$H(e^{j\omega}) = \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \sum_{m=-N}^N e^{jm\left(\omega - k \frac{2\pi}{2N+1}\right)} \quad (8.42)$$

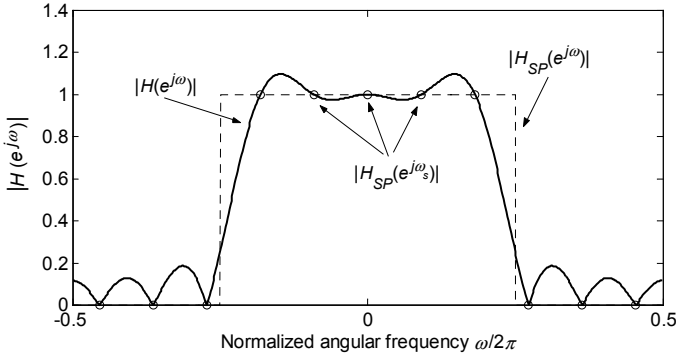


Figure 8.25. Amplitude response  $|H(e^{j\omega})|$  obtained sampling the specified frequency response  $H_{SP}(e^{j\omega})$  in 11 points

It is trivial now to compute the impulse response of the FIR filter, just taking the inverse Fourier transform of (8.42):

$$\begin{aligned} h(n) &= \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(e^{j\omega}) e^{jn\omega} d\omega = \\ &= \frac{1}{2\pi} \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \sum_{m=-N}^N e^{-jkm \frac{2\pi}{2N+1}} \int_{-\pi}^{+\pi} e^{j(n+m)\omega} d\omega \end{aligned} \quad (8.43)$$

The integral in (8.43) is equal to:

$$\int_{-\pi}^{+\pi} e^{j(n+m)\omega} d\omega = \frac{e^{j(n+m)\pi} - e^{-j(n+m)\pi}}{j(n+m)} = 2\pi \frac{\sin[(n+m)\pi]}{(n+m)\pi} = \begin{cases} 0 & \forall m \neq -n \\ 2\pi & \forall m = -n \end{cases} \quad (8.44)$$

By substitution of (8.44) into (8.43) results in the following expression for the coefficients of the FIR filter:

$$h(n) = \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) e^{jkn \frac{2\pi}{2N+1}} \quad (8.45)$$

Figure 8.26 shows the impulse response of length  $q+1=11$  obtained using (8.45) for the low-pass FIR filter of figure 8.25. The impulse response is not causal, since it is centered on  $n=0$ . In order to obtain a casual FIR filter the impulse response should be shifted by  $N$  steps. In the frequency domain it results in a multiplication of  $H(e^{j\omega})$  by the complex function  $e^{jN\omega}$ . Therefore the causal FIR filter interpolation and impulse response formulas as derived from (8.39) and (8.45) are given by:

$$H(e^{j\omega}) = \frac{e^{-j\omega N}}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \frac{\sin\left[\frac{2N+1}{2}\left(\omega - k \frac{2\pi}{2N+1}\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - k \frac{2\pi}{2N+1}\right)\right]} \tag{8.46}$$

$$h(n) = \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) e^{jk(n-N)\frac{2\pi}{2N+1}} \tag{8.47}$$

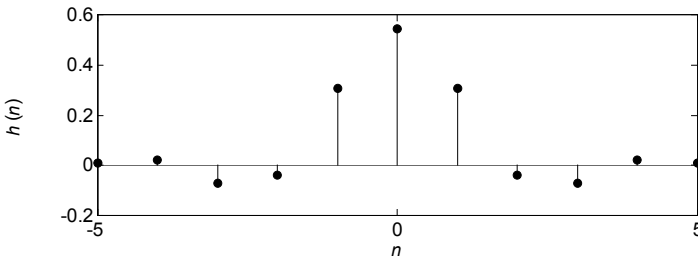


Figure 8.26. Anti-causal impulse response obtained taking 11 samples from the ideal low-pass filter response shown in figure 8.26

A design limitation of the frequency sampling method can occur when the specified amplitude frequency response  $|H_{SP}(e^{j\omega})|$  undergoes an abrupt change in amplitude across the transition band. The result is a strong ripple in the FIR filter frequency response  $H(e^{j\omega})$  owing to the Gibb’s phenomenon, as shown in figure 8.25. This difficulty can be alleviate relaxing the transition band requirements and allowing that some frequency samples around the cut-off frequency take values different from the ideal filter response. For example in figure 8.27 is shown the result obtained with the sampling frequency method when at the two samples around the cut-off frequencies  $\omega_C = \pm\pi/2$  are assigned the values  $\hat{H}_{SP} = \alpha = 0.5$  instead than the theoretical nil value. In the same figure with

dashed line is shown the FIR filter transfer function obtained without forcing these two samples to take values different from the ideal low-pass filter frequency response. It is clear that the ripple is greatly reduced, although the transition band is now wider.

Another example of a similar approach to the design of a FIR filter is shown in figure 8.28, where at four samples around the cut-off frequencies  $\omega_c = \pm\pi/2$  are assigned values different than the theoretical low-pass filter frequency response. This time the values assigned to the two samples around the cut-off frequency are  $\hat{H}_{SP} = 1 - \alpha$  and  $\hat{H}_{SP} = \alpha$ , with  $\alpha > 0$  ( $\alpha = 0.2$  in the example of figure 8.28). Again the result is a reduced ripple with an increased transition width.

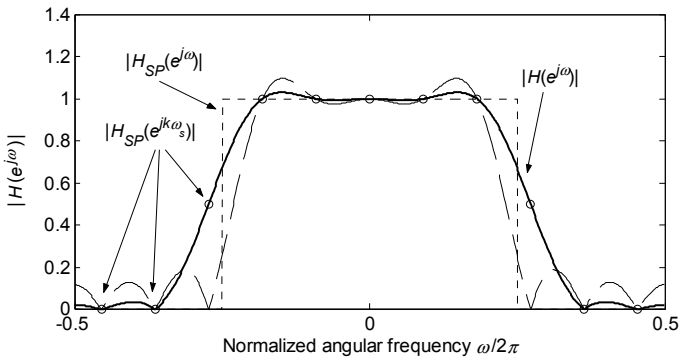


Figure 8.27. Filter design with the frequency sampling method obtained with one frequency sample specified in the transition band

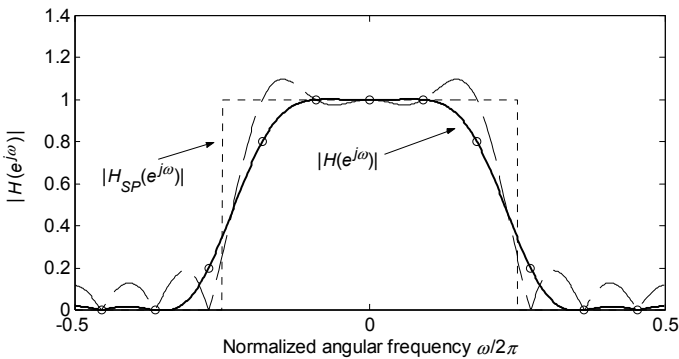


Figure 8.28. Filter design with the frequency sampling method obtained with two frequency samples specified in the transition band

The abovementioned criteria for designing FIR filter using the sampling frequency method can lead to an automatic procedure for the computation of the filter coefficients. The methodology is based on the definition of the design objective in terms of a design error  $\varepsilon(e^{j\omega})$  specified in the frequency domain and defined as the difference of the actual FIR filter frequency response  $H(e^{j\omega})$  and the specified ideal filter frequency response  $H_{SP}(e^{j\omega})$ :

$$\varepsilon(e^{j\omega}) = H(e^{j\omega}) - H_{SP}(e^{j\omega}) \quad (8.48)$$

The design criterion takes normally the form of an optimisation algorithm based on a mathematical metrics applied to the design error  $\varepsilon(e^{j\omega})$ . The metrics used are the  $L_2$  (mean square) and  $L_\infty$  (maximum) norm, leading to the following optimisation design criteria:

- Minimum Square Error (MSE)

$$\hat{H}_{SP}(k) = \min_{\hat{H}_{SP}(k)} \left[ \frac{1}{2\pi} \int_{-\pi}^{+\pi} |\varepsilon(e^{j\omega})|^2 d\omega \right] \quad (8.49)$$

- Minimax:

$$\hat{H}_{SP}(k) = \min_{\hat{H}_{SP}(k)} \left[ \max |\varepsilon(e^{j\omega})| \right] \quad (8.50)$$

Since relationship (8.46) is linear in the parameters  $\hat{H}_{SP}(k)$ , the optimization problems (8.49) and (8.50) can be solved numerically using linear optimization algorithms.

The two metrics in (8.49) and (8.50) have different advantages and disadvantages. The MSE criterion generally leads to simpler algorithms, but the solution could have large local errors although the mean square error was small. On the other way the minimax criterion is more complicate computationally but it limits the worst case error.

A weakness of the sampling frequency design method is that the samples  $\hat{H}_{SP}(k)$  are uniformly distributed along the frequency axis and this results in a lack of flexibility on specifying the cut-off frequencies of the filter. Increasing the number of frequency samples will improve the ability to specify the frequency response transition at the cut-off frequencies but this will be highly inefficient.

The transfer function of the FIR filter is given by the z-transform of the impulse transform (8.47):

$$\begin{aligned}
 H(z) &= \sum_{n=0}^{2N} h(n)z^{-n} = \frac{1}{2N+1} \sum_{n=0}^{2N} \sum_{k=-N}^N \hat{H}_{SP}(k) e^{jk(n-N)\frac{2\pi}{2N+1}} z^{-n} = \\
 &= \frac{1}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) e^{-jkN\frac{2\pi}{2N+1}} \sum_{n=0}^{2N} e^{jkn\frac{2\pi}{2N+1}} z^{-n}
 \end{aligned} \tag{8.51}$$

The second summation in (8.51) is the  $z$ -transform of the sequence:

$$\left\{ e^{jkn\frac{2\pi}{2N+1}} [I(n) - I(n - 2N - 1)] \right\} \tag{8.52}$$

which according to table 7.1 is equal to:

$$\sum_{n=0}^{2N} e^{jkn\frac{2\pi}{2N+1}} z^{-n} = \sum_{n=-\infty}^{+\infty} e^{jkn\frac{2\pi}{2N+1}} [I(n) - I(n - 2N - 1)] z^{-n} = \frac{1 - z^{-(2N+1)}}{1 - e^{jk\frac{2\pi}{2N+1}} z^{-1}} \tag{8.52}$$

Replacing the relationship (8.52) into (8.50) it results the following expression for the transfer function of the FIR filter obtained with the frequency sampling method:

$$H(z) = \frac{1 - z^{-(2N+1)}}{2N+1} \sum_{k=-N}^N \hat{H}_{SP}(k) \frac{e^{-jkN\frac{2\pi}{2N+1}}}{1 - e^{jk\frac{2\pi}{2N+1}} z^{-1}} \tag{8.53}$$

### EFFECTS OF FILTER COEFFICIENTS QUANTIZATION

Limitation in the number of bits available for storing the filter coefficients and the internal data results in a change in its frequency response. The quantization effect on the frequency response can be analyzed in the frequency domain in terms of an error function  $\Delta H(e^{j\omega})$  which is the difference between the actual frequency response  $H(e^{j\omega})$ , resulting from the coefficients and data quantization, and the theoretical frequency response  $H_{TH}(e^{j\omega})$ :

$$\Delta H(e^{j\omega}) = H(e^{j\omega}) - H_{TH}(e^{j\omega}) \tag{8.54}$$

Let us start considering first the effect due to the quantization of the filter coefficients only. If  $\Delta$  represents the quantization step, the rounding error  $\Delta h(k)$  on each coefficients  $h(k)$  of the filter will be:



$$|\Delta h(k)| \leq \Delta/2, \forall k \in 0 \dots q \quad (8.55)$$

Taking into account the FIR filter transfer function (8.5) the maximum value of  $\Delta H(e^{j\omega})$  is given by:

$$\max |\Delta H(e^{j\omega})| = \max \left| \sum_{k=0}^q \frac{\Delta}{2} e^{-jk\omega} \right| \leq (q+1) \frac{\Delta}{2} \quad (8.56)$$

In many cases the worst case estimation (8.56) of the coefficients quantization effect is too rough. A better estimation can be obtained modeling the quantization error  $\Delta h(k)$  of each filter coefficient as a random noise uniformly distributed between  $-\Delta/2$  and  $\Delta/2$ . The filter coefficients variance is in this case:

$$\sigma^2 [\Delta h(k)] = \frac{\Delta^2}{12} \quad (8.57)$$

Considering statistically independent the quantization noise on each filter coefficient, the standard deviation of the (8.54) is given by the uncertainty propagation law applied to (8.5):

$$\sigma^2 [\Delta H(e^{j\omega})] = \sum_{k=0}^q \left| \frac{\partial H}{\partial h(k)} \right|^2 \sigma^2 [\Delta h(k)] = \sum_{k=0}^q |e^{jk\omega}|^2 \sigma^2 [\Delta h(k)] \quad (8.58)$$

Since  $|e^{jk\omega}| = 1$  expression (8.58) simplifies to:

$$\sigma [\Delta H(e^{j\omega})] = \sqrt{\sum_{k=0}^q \sigma^2 [\Delta h(k)]} = \sqrt{\frac{q+1}{3}} \frac{\Delta}{2} \quad (8.59)$$

The estimation (8.59) of the quantization effect on the filter frequency response is smaller than (8.56) but it is more significant since the worst case error (8.56) is very unlikely.

If  $\delta_{SP} = \min(\delta_P, \delta_S)$  is the minimum ripple allowed by the filter specifications and  $\delta_0$  is the ripple obtained without quantizing the filter coefficients, the variance (8.59) must satisfy the following limit:

$$\sigma [\Delta H] = \sqrt{\frac{q+1}{3}} \frac{\Delta}{2} \leq \frac{\delta_{SP} - \delta_0}{2} \quad (8.60)$$

If  $N_C$  is the number of bit used for representing the each filter coefficient (and its sign), the quantization step is expressed by:

$$\Delta = \frac{\max|h(k)|}{2^{N_c-1}} \tag{8.61}$$

A useful formula for estimating the number of bits required for representing the filter coefficients can be obtained from (8.61) together with the inequality (8.60):

$$N_c = 1 + \log_2 \frac{\max|h(k)|}{\Delta} \geq 1 + \log_2 \frac{\sqrt{q+1} \max|h(k)|}{(\delta_{SP} - \delta_0)\sqrt{3}} \tag{8.62}$$

The consequences of filter coefficients quantization can be studied also in terms of its effects on the zeros location. Starting from the expression of the FIR filter transfer function (8.4), which can be factorized as:

$$H(z) = h(0) \frac{\prod_{j=1}^q (z - z_j)}{z_i^q} \tag{8.63}$$

and using the derivative chain rule to calculate the transfer function derivative with respect to the filter coefficients:

$$\left[ \frac{\partial H}{\partial h(k)} \right]_{z=z_i} = \left[ \frac{\partial H}{\partial z_i} \right]_{z=z_i} \frac{\partial z_i}{\partial h(k)} \tag{8.64}$$

the sensitivity of the  $i^{\text{th}}$  zero location to the quantization of the  $k^{\text{th}}$  coefficient  $h(k)$  is expressed by the derivative  $\partial z_i / \partial h(k)$  in (8.64). The derivative  $\partial H / \partial h(k)$  in (8.64) derives directly from (8.4):

$$\left[ \frac{\partial H}{\partial h(k)} \right]_{z=z_i} = z_i^{-k} \tag{8.65}$$

while the derivative  $\partial H / \partial z_i$  results from (8.63):

$$\left[ \frac{\partial H}{\partial z_i} \right]_{z=z_i} = -h(0) \frac{\prod_{\substack{j=1 \\ j \neq i}}^q (z_i - z_j)}{z_i^q} \tag{8.66}$$

Replacing (8.66) and (8.65) in (8.64) and solving for  $\partial z_i / \partial h(k)$  the  $i^{\text{th}}$  zero sensitivity to the  $k^{\text{th}}$  filter coefficient quantization is given by the following formula:

$$\frac{\partial z_i}{\partial h(k)} = - \frac{z_i^{q-k}}{h(0) \prod_{\substack{j=1 \\ j \neq i}}^q (z_i - z_j)} \tag{8.67}$$

The importance of formula (8.67) is that it shows how the sensitivity of zero location due to the coefficients quantization could be very high if the filter poles were very close to each other, since the denominator of (8.67) tends to zero. Figure 8.29 shows this effect for a 8<sup>th</sup> order FIR filter with linear phase with transfer function:

$$H(z) = 1 + az^{-1} + bz^{-2} + cz^{-3} + dz^{-4} + cz^{-5} + bz^{-6} + az^{-7} + z^{-8} \tag{8.68}$$

In figure 8.29 are shown four of the eight zeros with circles and with asterisks the zeros when the four coefficients  $a, b, c$  and  $d$  in (8.68) are affected by a 0.5% random perturbation. In figure 8.29b the eight zeros of (8.68) are more far apart from each other than in figure 8.29a. It is evident from the picture that the change in location of the zeros, due to the coefficient quantization, is smaller in figure 8.29b than in figure 8.29a as predicted by equation (8.67).

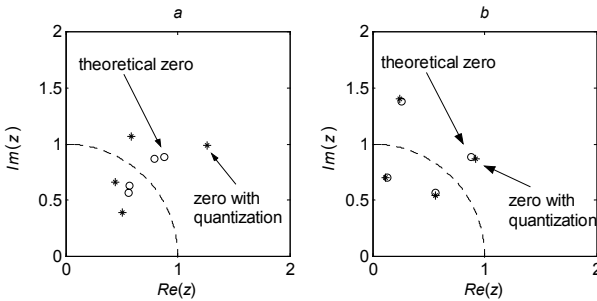


Figure 8.29. Effect on the zeros location of 0.5% random perturbation on the coefficients of a 8<sup>th</sup> order linear phase FIR filter: (a) close zeros - (b) distant zeros

FIR FILTER ARCHITECTURES.

The structure of a linear discrete filter is composed by a discrete dynamical system implementing digitally the operations involved in the difference state equations (7.4a)

and the output transformation (7.4b). The operations involved are storage, multiplication and addition. Furthermore the complexity of the system is determined by the number of these three types of operations needed and by their precision in terms of the number of bits required for the data and coefficients quantization.

The simplest architecture for implementing a  $q^{\text{th}}$  order FIR filter can be derived directly from equation (8.2) as shown in the block diagram of figure 8.1 and reported here in figure 8.30 for convenience. This architecture, named *direct* structure, requires:

- $2q+1$  memory registers (for storing the filter coefficients and state),
- $q+1$  multipliers,
- $q$  adders.

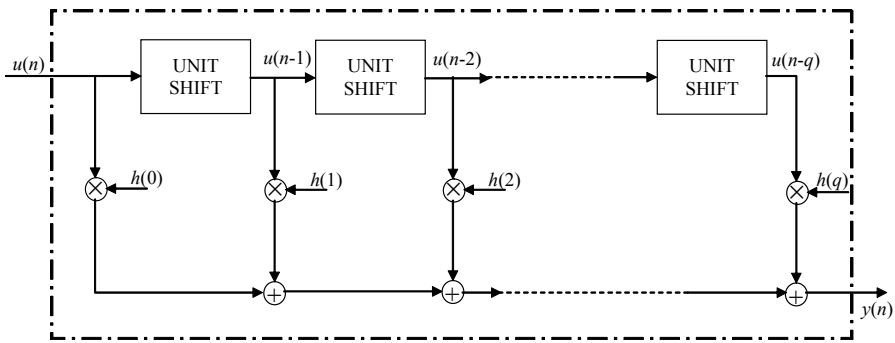


Figure 8.30. Direct structure of a  $q^{\text{th}}$  order FIR filter

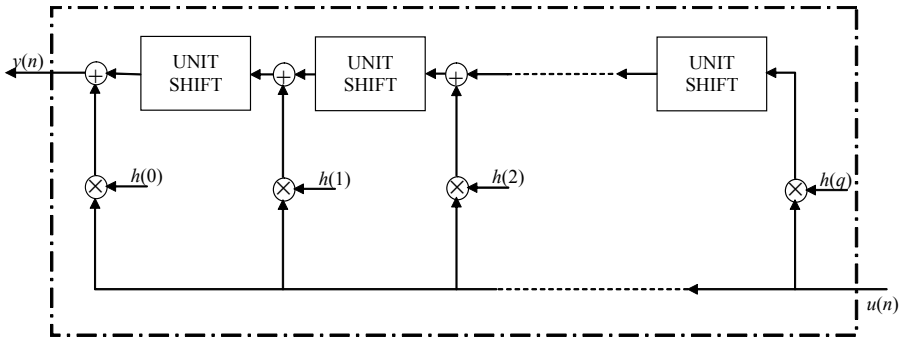


Figure 8.31. Transposed structure of a  $q^{\text{th}}$  order FIR filter

The architecture of figure 8.30 is not the only way the arrange the calculations required for implementing a FIR filter. Another structure, named *transposed*, can be

derived from the direct realization of figure 8.30 by inverting all the signal flows: figure 8.31 shows the resulting numerical network. The complexity of the new structure is the same as for the direct implementation:

- $2q+1$  memory registers (for storing the filter coefficients the state),
- $q+1$  multipliers,
- $q$  adders.

If the FIR filter has linear phase the filter coefficients are real and have to satisfy the symmetry condition (8.22). This condition can be exploited to halve the number of multiplication and sums and the number of registers required for storing the filter coefficients. The architecture for  $q^{\text{th}}$  order linear phase FIR filter is given in figure 8.32 for  $q$  even and in figure 8.33 for  $q$  odd. The complexity of these architecture are summarized, respectively, by the following figures:

$q$  even:

- $(3q+2)/2$  memory registers (for storing the filter coefficients and state),
- $q/2+1$  multipliers,
- $q$  adders.

$q$  odd:

- $(3q+1)/2$  memory registers (for storing the filter coefficients and state),
- $(q+1)/2$  multipliers,
- $q$  adders.

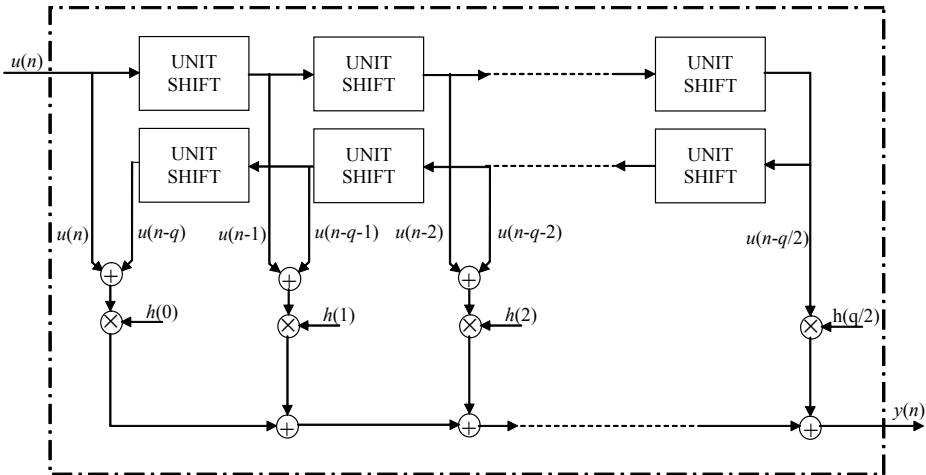


Figure 8.32. Reduced multiplication  $q^{\text{th}}$  order linear phase FIR architecture, with  $q$  even

From the above figures it results that the numbers of multipliers necessary for implementing a linear phase filter are about half the number required for a direct structure while the number of memory registers are about  $q/2$  less.

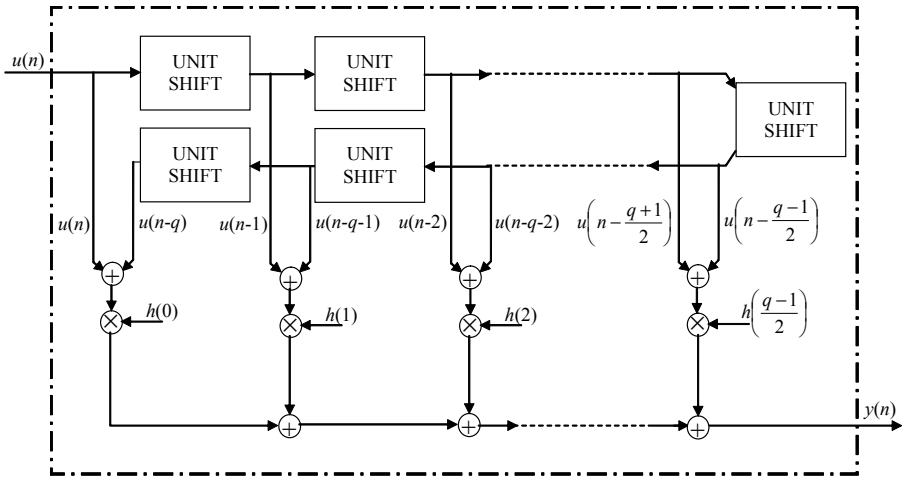


Figure 8.33. Reduced multiplication  $q^{\text{th}}$  order linear phase FIR architecture, with  $q$  odd

An important limitation concerning the FIR filter structures previously discussed regards the effect of coefficients quantization. This effect is particularly serious when the filter transfer function has zeros in the  $z$ -plane closed to each other, for the reasons explained in the previous paragraph.

In order to limit the effect of filter coefficients quantization on the zeros location let us consider the following factorization of the FIR filter transfer function (8.4) in  $q$  products of first degree polynomials:

$$H(z) = h(0) \prod_{k=1}^q (1 - z_k z^{-1}) = h(0) \prod_{k=1}^q H_k(z) \quad (8.69)$$

where  $z_k$  are the  $q$  zeros of the filter. The product in (8.69) of the  $q$  terms:

$$H_k(z) = 1 - z_k z^{-1} \quad (8.70)$$

means that the impulse response of a  $q^{\text{th}}$  order FIR filter can be expressed as  $q$  convolutions of the inverse  $z$ -transform of  $H_k(z)$ . According to (2.27) this is equivalent to think at a  $q^{\text{th}}$  order FIR filter as a realization of the cascade of  $q$   $1^{\text{st}}$  order FIR filters which transfer function is expressed by (8.70). The effect of the filter coefficients quantization

is limited now because each of the  $q$  stages in cascade realize only one of the  $q$  zero of the FIR filter.

The difference equation which corresponds to (8.70), reminding that the term  $z^{-1}$  is equivalent to a unit shift, is:

$$y_k(n) = u_k(n) - z_k \cdot u_k(n-1) \tag{8.71}$$

where  $u_k(n)$  and  $y_k(n)$  represent the  $k^{\text{th}}$  section input and output sequences, respectively.

The block diagram of the numerical network implementing one of the  $q$  first order section (8.71) is shown figure 8.34. Each section requires:

- 2 memory registers (for storing the filter coefficients and state),
- 1 multiplier,
- 1 adder.

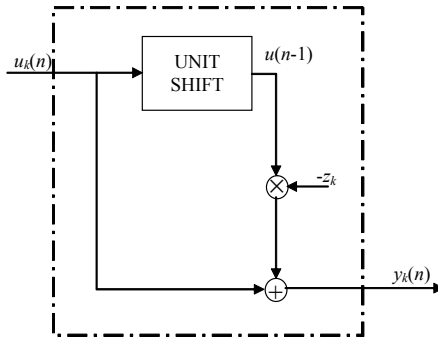


Figure 8.34. First order stage implementing the zeros  $z_k$  of a FIR filter using complex arithmetic

Since the zeros  $z_k$  are generally complex, the structure shown in figure 8.34 requires the implementation of complex arithmetic operations. If the FIR filter has real coefficients, i.e. impulse response is real, the zeros of its transfer function take place in pairs complex conjugate:

$$\begin{cases} z_k^{(1)} = r_k e^{j\theta_k} \\ z_k^{(2)} = r_k e^{-j\theta_k} \end{cases} \tag{8.72}$$

In this case it is better to factorize the filter transfer function (8.69) with  $q/2$  second order polynomial in  $z^{-1}$ :

$$H(z) = h(0) \prod_{k=1}^{\frac{q}{2}} (1 + a_k z^{-1} + b_k z^{-2}) \tag{8.73}$$

with:

$$\begin{cases} a_k = -2 \operatorname{Re}(z_k) \\ b_k = |z_k|^2 \end{cases} \tag{8.74}$$

If  $q$  is odd the factors in (8.71) are  $(q+1)/2$ , with  $b_{(q+1)/2} = 0$ .

The advantage of the factorization (8.73) is that the coefficients  $a_{(q+1)/2}$ ,  $a_k$  and  $b_k$  are real. The difference equation which corresponds to the second order polynomials in (8.73) is:

$$y_k(n) = u_k(n) + a_k \cdot u_k(n-1) + b_k \cdot u_k(n-2) \tag{8.75}$$

Figure 8.35 shows the digital network implementing (8.75). The complexity of this section is given by:

- 4 memory registers (for storing the filter coefficients the filter state),
- 2 multipliers,
- 2 adders.

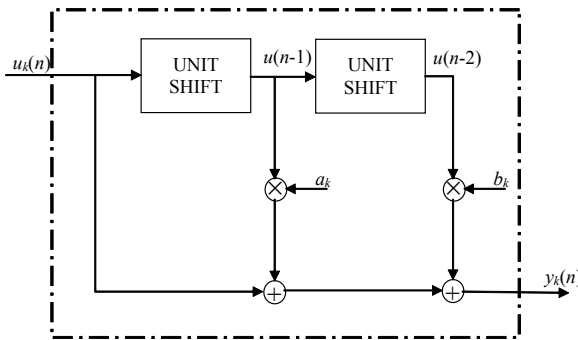


Figure 8.35. Second order stage implementing a FIR filter with real impulse response, without using complex arithmetic

Equation (8.74) shows that the effect of quantization on the coefficients  $a_k$  and  $b_k$  results in the quantization of the squared absolute value and of the real part of the zero  $z_k$  and its complex conjugate. Figure 8.36a shows this effect when the filter coefficients



(8.74) are quantized with 4 bits. The zeros cannot lay anywhere in the complex plane, but they must stay on the grid defined by the intersection of the continuous and dotted lines. For example in figure 8.36b with dots are shown the zeros that can be obtained with the 2<sup>nd</sup> order stage of figure 8.35 when the coefficients quantization is made using 6 bits.

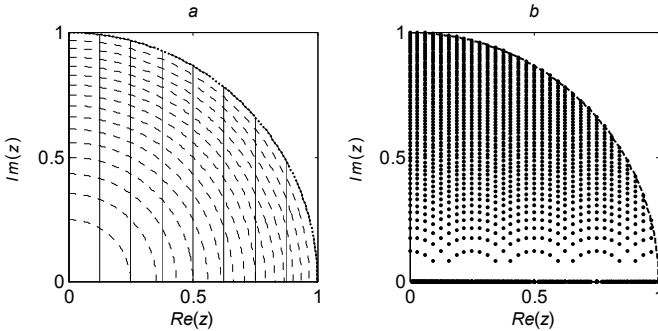


Figure 8.36. Zeros location for a 2<sup>nd</sup> order FIR filter stage with quantized coefficients; (a) grid for 4 bits quantization; (b) zeros location with 6 bits quantization

If the FIR filter has linear phase its complex zeros happens to be grouped in reciprocal pair, i.e. if  $z_k$  is a zero so is  $1/z_k$  as well. Moreover if the filter has real coefficients the zeros must be in complex conjugate pairs, so that the zeros must occur in groups of four complex numbers such that:

$$\begin{cases} z_k^{(1)} = r_k e^{j\theta_k} \\ z_k^{(2)} = r_k e^{-j\theta_k} \end{cases} \quad \begin{cases} z_k^{(3)} = \frac{1}{r_k} e^{j\theta_k} \\ z_k^{(4)} = \frac{1}{r_k} e^{-j\theta_k} \end{cases} \quad (8.76)$$

This particular distribution of the roots is shown in figure 8.37 where the four zeros (8.76) are plotted together with the unit circle in the  $z$ -plane.

Adopting two 2<sup>nd</sup> order stages for realizing the 4 zeros, because of the quantization of their coefficients, the actual roots will only approximate (8.76). If one 2<sup>nd</sup> order stage realizes the complex conjugate pair  $z_k^{(1)}$  and  $z_k^{(2)}$  while the other stage the pair  $z_k^{(3)}$  and  $z_k^{(4)}$  the actual zeros will be two complex conjugate pairs that generally will not satisfy relation (8.76) with each other, i.e. the resulting filter will not have liner phase.

To force the filter to have linear phase, even though the coefficients quantization, it is necessary to factorize the filter transfer function (8.4) into fourth order factors with real coefficients having the symmetry property (8.22) of the FIR filter coefficients:

$$H(z) = h(0) \prod_{k=1}^{\frac{q}{4}} (1 + a_k z^{-1} + b_k z^{-2} + a_k z^{-3} + z^{-4}) \tag{8.77}$$

with:

$$\begin{cases} a_k = -2 \operatorname{Re} \left( z_k + \frac{1}{z_k} \right) \\ b_k = |z_k|^2 + \frac{1}{|z_k|^2} + 4 \operatorname{Re}(z_k) \operatorname{Re} \left( \frac{1}{z_k} \right) \end{cases} \tag{8.78}$$

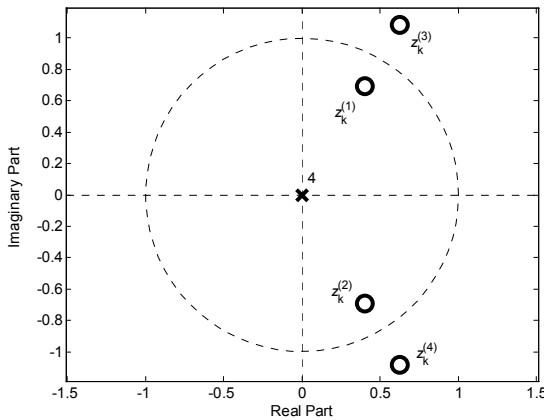


Figure 8.37. Location of complex zeros in the  $z$ -plane for a linear phase FIR filter

In this way the linear phase FIR filter is realized as the cascade of  $q/4$  stages such that each of them is a linear phase 4<sup>th</sup> order FIR filter with transfer function given by (8.77).

To the factor (8.77) corresponds the following difference equation in the  $n$ -domain and the numerical network shown in figure 8.38:

$$y_k(n) = [u_k(n) + u_k(n - 4)] + a_k \cdot [u_k(n - 1) + u_k(n - 3)] + b_k \cdot u_k(n - 2) \tag{8.79}$$

The complexity of this structure is reduced because of the coefficients symmetry and given by:

- 6 memory registers (for storing the filter coefficients and state),
- 2 multipliers,
- 4 adders.

Compared to a pair of 2<sup>nd</sup> order stages the 4<sup>th</sup> order linear phase structure requires two less memory registers and multipliers.

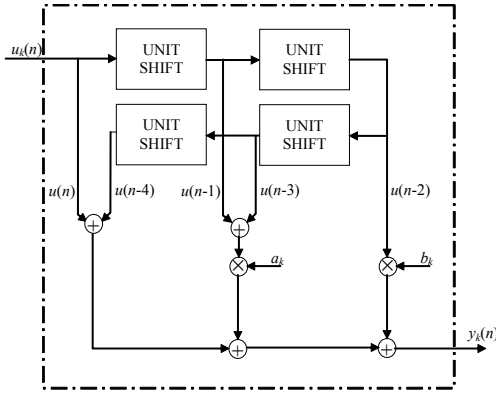


Figure 8.38. Fourth order stage implementing a FIR filter with real impulse response and linear phase

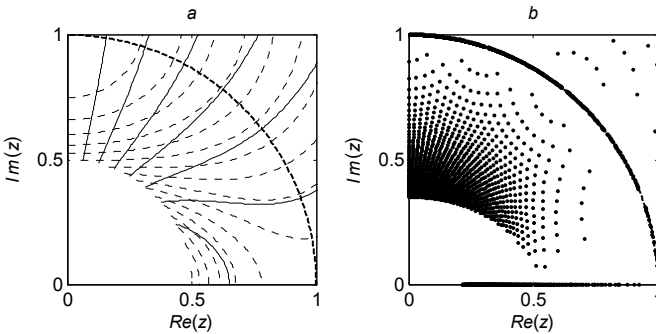


Figure 8.39. Zeros location for a 4<sup>th</sup> order linear phase FIR filter stage with quantized coefficients and  $0.5 \leq |z_k| \leq 0.5^{-1}$ ; (a) grid for 4 bits quantization; (b) zeros location with 6 bits quantization

A far as the filter coefficients quantization is concerned figure 8.39a shows the grid for the zeros positions in the complex  $z$ -plane considering 4 bits for the quantization of the coefficients (8.78) when  $0.5 \leq |z_k| \leq 0.5^{-1}$ . Figure 8.39b shows instead the zeros positions when the quantization is obtained using 6 bits. The two figures illustrate the bad coverage of the  $z$ -plane due to the quantization of the coefficients (8.78) of the fourth order structure. The situation is even worse under the hypothesis that  $0.05 \leq |z_k| \leq 0.05^{-1}$ , as shown

in figure 8.40, where a resolution of 5 bits has been considered for the coefficients quantization. The reason for that are the term  $z_k$  and  $1/z_k$  in (8.78); when  $|z_k| \rightarrow 0$  the dynamics of the coefficients grows to infinity and the quantization became coarser. The effect on the zeros location grid is shown in figure 8.40 for  $|z_k| \leq 1$ ; the grid results very dense only around the circle  $|z_k|=0.05$  (and the circle  $|z_k|=0.05^{-1}$  not shown in figure 8.40) and very sparse elsewhere.

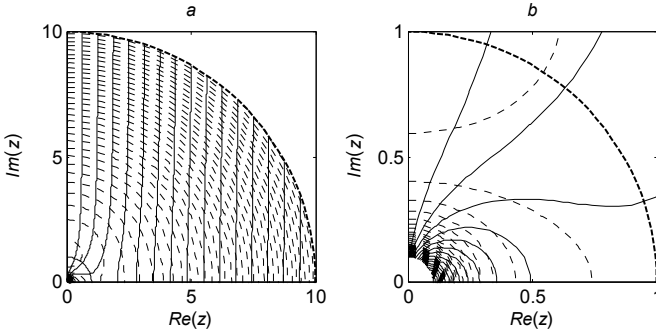


Figure 8.40. Zeros location grid for a 4<sup>th</sup> order linear phase FIR filter stage with quantized coefficients on 5 bits and  $0.1 \leq |z_k| \leq 0.1^{-1}$

A better coverage of the z-plane after coefficient quantization can be gained forcing the 4<sup>th</sup> order stage zeros to stand a quantization as uniform as possible, for example with the following quantization of the zeros positioning:

$$\begin{cases} \alpha_k = -2\text{Re}(z_k) \\ \beta_k = |z_k|^2 \end{cases} \quad (8.80)$$

In this case the transfer function of a single stage assume the following form:

$$\begin{aligned} H_k(z) &= (1 + a_k z^{-1} + b_k z^{-2} + a_k z^{-3} + z^{-4}) = \\ &= \frac{1}{\beta_k} (1 + \alpha_k z^{-1} + \beta_k z^{-2}) (\beta_k + \alpha_k z^{-1} + z^{-2}) \end{aligned} \quad (8.81)$$

Figure 8.41 shows the resulting numerical network, characterized by:

- 7 memory registers (for storing the filter coefficients and state),
- 5 multipliers,
- 4 adders.

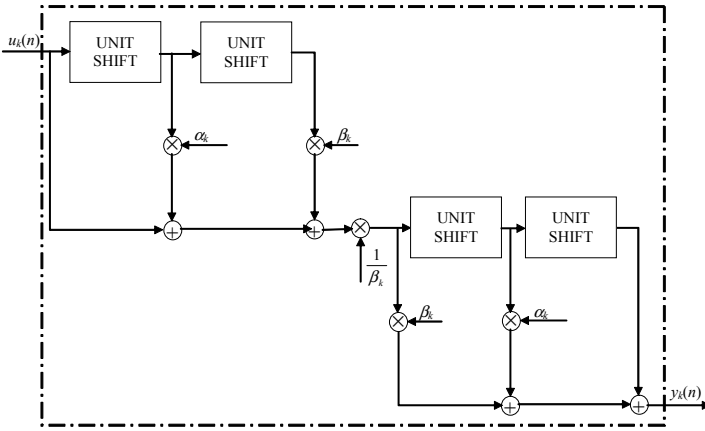


Figure 8.41. Numerical network for one stage of a linear phase 4<sup>th</sup> order FIR filter

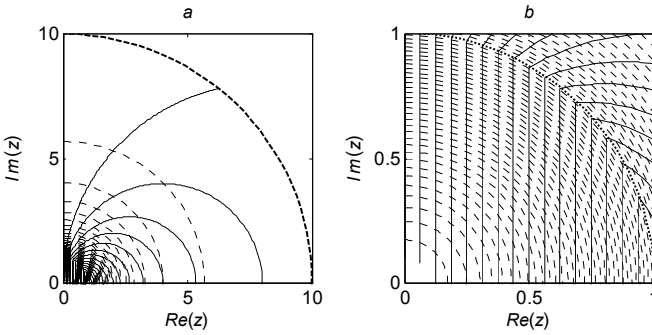


Figure 8.42. Zeros location grid for a 4<sup>th</sup> order linear phase FIR filter stage with quantized coefficients on 5 bits and  $0.1 \leq |z_k| \leq 0.1^{-1}$

Figure 8.43 shows the numerical network for the synthesis of a FIR filter with the frequency sampling method (8.53). It is composed by a comb FIR filter in cascade with the parallel of  $N+1$  stages characterized by a feedback loop. This stages are filter with an infinite impulse response and will be described in the next chapter.

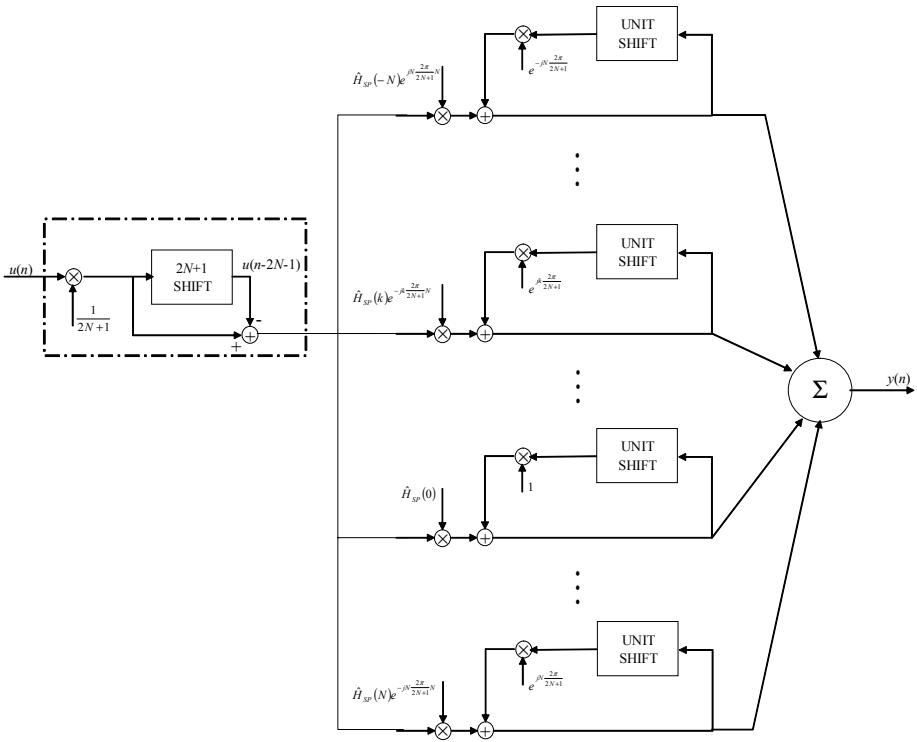


Figure 8.43. Numerical structure of a FIR filter designed with the sampling frequency method

## INFINITE IMPULSE RESPONSE (IIR) FILTERS

 $z$ -DOMAIN REPRESENTATION OF IIR FILTERS.

An infinite impulse response (IIR) filter of order  $q$  is linear shift invariant causal discrete systems having as the  $q$  state variables at step  $n$  the values of the system output at the  $N$  previous steps and the system input at the  $M$  previous steps, with  $q = \max(N, M)$ . As the name suggests the impulse response of an IIR filter is a sequence  $h(n)$  persisting forever:

$$\{h(n)\} = \{h(0)\delta(n) + h(1)\delta(n-1) + \dots\} \quad (9.1)$$

In the  $z$ -domain the IIR filter can be analyzed in terms of its transfer function:

$$H(z) = \sum_{k=0}^{+\infty} h(k)z^{-k} \quad (9.2)$$

which, as explained in chapter 7, can be written in the format (7.43) here reported for convenience:

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{k=0}^M \alpha_k z^{-k}}{1 + \sum_{k=1}^N \beta_k z^{-k}} \quad (9.3)$$

If  $M \geq N$  the ratio (9.3), between the numerator polynomial  $N(z)$  of degree  $M$  and the denominator polynomial  $D(z)$  of degree  $N$ , can be split as the sum of a quotient polynomial  $Q(z)$  of degree  $M-N$  and the ratio between the remainder polynomial  $R(z)$  of degree  $N-1$  and the denominator  $D(z)$ :

$$H(z) = Q(z) + \frac{R(z)}{D(z)} = \sum_{k=0}^{M-N} \gamma_k z^{-k} + \frac{\sum_{k=0}^{N-1} \alpha_k z^{-k}}{1 + \sum_{k=1}^N \beta_k z^{-k}} \tag{9.4}$$

In (9.4) the transfer function  $Q(z)$  corresponds to a FIR filter of order  $M-N$ .

In If  $M < N$  the quotient  $Q(z)$  is zero.

As shown in chapter 7 the transfer function (9.3) admit the factorization (7.44), reported below:

$$H(z) = \alpha_0 \frac{\prod_{k=1}^M (1 - Z_k z^{-1})}{\prod_{k=1}^N (1 - P_k z^{-1})} = \alpha_0 z^{N-M} \frac{\prod_{k=1}^M (z - Z_k)}{\prod_{k=1}^N (z - P_k)} \tag{9.5}$$

An IIR filter is characterized by  $M$  zeros in  $Z_k$ ,  $N$  poles in  $P_k$  and in the origin, if  $N > M$  there are  $N-M$  zeros otherwise if  $N < M$  there  $M-N$  poles.

The frequency response of an IIR filter can be determined from either (9.4) or (9.5) by replacing  $z$  with  $e^{j\omega}$ :

$$\begin{aligned} H(e^{j\omega}) &= \sum_{k=0}^{M-N} \gamma_k e^{-jk\omega} + \frac{\sum_{k=0}^{N-1} \alpha_k e^{-jk\omega}}{1 + \sum_{k=1}^N \beta_k e^{-jk\omega}} = \\ &= \alpha_0 e^{j(N-M)\omega} \frac{\prod_{k=1}^M (e^{j\omega} - Z_k)}{\prod_{k=1}^N (e^{j\omega} - P_k)} \end{aligned} \tag{9.6}$$

***n*-DOMAIN REPRESENTATION OF IIR FILTERS.**

The difference equation in the  $n$ -domain representing the relationship between the input sequence  $\{u(n)\}$  and the output sequence  $\{y(n)\}$  for an IIR filter can be derived directly from the expression of the transfer function (9.3):



$$Y(z) = \frac{\sum_{k=0}^M \alpha_k z^{-k}}{1 + \sum_{k=1}^N \beta_k z^{-k}} U(z) \tag{9.7}$$

⇓

$$y(n) = \alpha_0 u(n) + \sum_{k=1}^M \alpha_k u(n-k) - \sum_{k=1}^N \beta_k y(n-k)$$

The block diagram in figure 9.1 shows graphically this relationship.

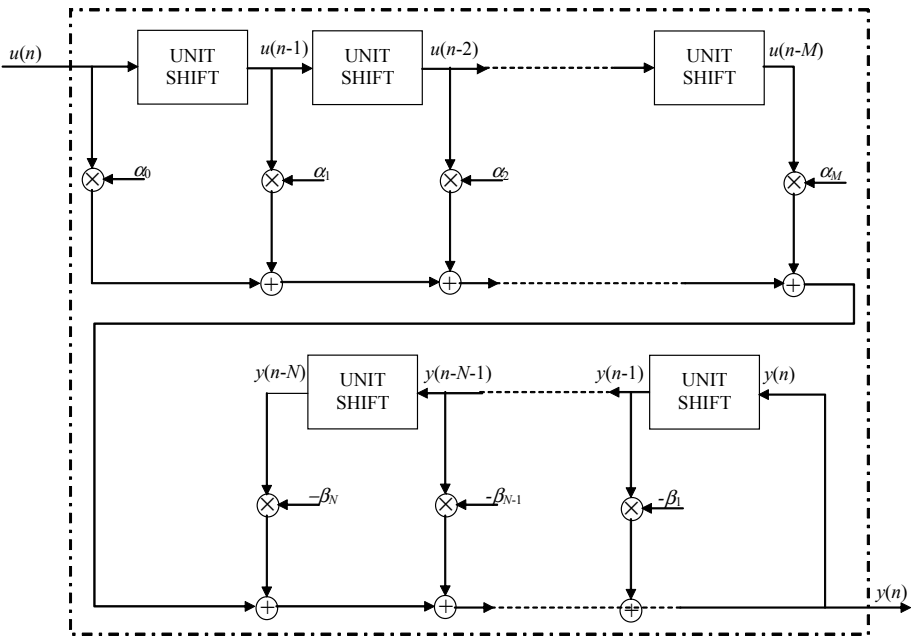


Figure 9.1. *n*-domain block diagram of an IIR filter

The state vector for this type of filter, according to (7.39), is given by the  $M$  values of the input sequence  $\{u\}$  and the  $N$  values of the output sequence  $\{y\}$  preceding the current step  $n$ :

$$\mathbf{x}(n) = [y(n-1) \ \dots \ y(n-N) \ u(n-1) \ \dots \ u(n-M)]^T \tag{9.8}$$

The block diagram of figure 9.1 shows that an IIR filter can be seen as the cascade of a FIR filter and a purely recursive IIR section with transfer function

$$H_R(z) = \frac{1}{1 + \sum_{k=1}^N \beta_k z^{-k}} \tag{9.9}$$

responsible of the infinite impulse response of the filter.

An important property of IIR filters is that the feedback loop due to the purely recursive stage allows the synthesis of more selective filtering transfer function synthesis if compared with the transfer function attainable with a FIR filter of the same order.

By factorization of the purely recursive stage transfer function  $H_R(z)$ :

$$H_R(z) = \frac{1}{\prod_{k=1}^N (1 - P_k z^{-1})} = \frac{z^N}{\prod_{k=1}^N (z - P_k)} \tag{9.10}$$

it results that this stage is responsible for the synthesis of the  $N$  poles of the whole IIR filter (9.5), while the FIR stage realizes the  $M$  zeros of  $H(z)$ . Since each stage realize also  $N$  zeros and  $M$  poles in the origin respectively this explains why in (9.5) are present also  $N-M$  zeros in the origin.

PURELY RECURSIVE IIR FILTERS

Considering figure 9.1, the simplest IIR filter is a filter with no FIR stage and with a recursive stage composed by one unit shift block only, as shown in figure 9.2.

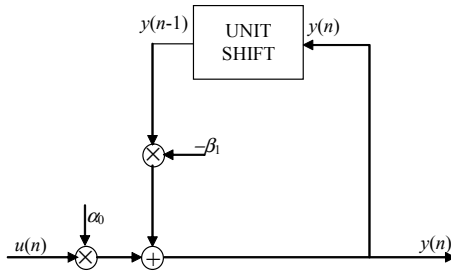


Figure 9.2. Block diagram of a first order purely recursive IIR filter

The consequential difference equation is given by:

$$y(n) = \alpha_0 u(n) - \beta_1 y(n-1) \tag{9.11}$$

Comparing (9.11) with the more general difference equation (9.7) it results  $M=0$  and  $N=1$ , giving a filter order  $q=M+N=1$ . This type of IIR filter is known as a *first order purely recursive IIR filter*.

The transfer function of the first order purely recursive filter is given by the  $z$ -transform of (9.11):

$$Y(z) = \alpha_0 U(z) - \beta_1 Y(z)z^{-1} = \frac{\alpha_0}{1 + \beta_1 z^{-1}} U(z) \tag{9.12}$$

$$\Downarrow$$

$$H(z) = \frac{Y(z)}{U(z)} = \frac{\alpha_0}{1 + \beta_1 z^{-1}} = \frac{\alpha_0 z}{z + \beta_1}$$

The first order purely recursive filter has 1 zero in the origin and one pole at  $z = -\beta_1$ .

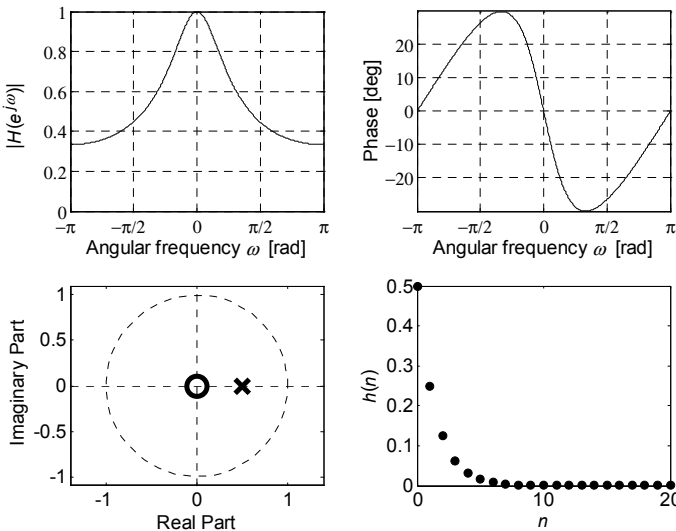


Figure 9.3. Frequency response, zero-pole diagram and impulse response for a first order purely recursive IIR filter with a pole in  $z=0.5$

As an example the frequency response, together with the zero-pole diagram and the impulse response, for this type of filter is shown in figure 9.3. The pole is placed at  $z=0.5$

by setting  $\beta_1 = -0.5$ . In order to maintain a unit DC gain the value of  $\alpha_0$  was forced to the following value:

$$|H(e^{j\omega})|_{\omega=0} = \frac{\alpha_0}{1 + \beta_1} = 1 \Rightarrow \alpha_0 = 1 + \beta_1 \tag{9.13}$$

From the frequency response diagram of figure 9.3 it appears that this simple IIR filter can behave as a low pass filter. A similar behavior can be obtained with a FIR filter of the same order and therefore of similar complexity. For example the response of a 1<sup>st</sup> order moving average FIR filter is shown in figure 9.4. It is evident how the feedback loop of the IIR filter results in a much higher selectivity in the frequency response, although it is noteworthy the highly non linearity of the IIR phase response.

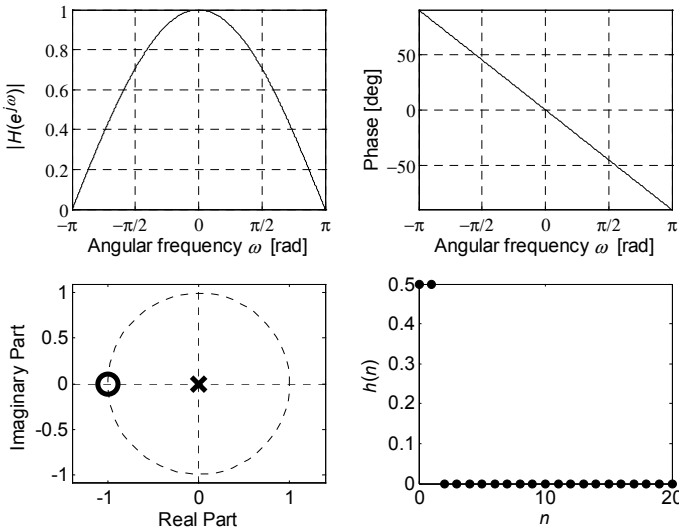


Figure 9.4. Frequency response, zero-pole diagram and impulse response for a first order moving average FIR filter

In the previous chapter it was shown that FIR filters have poles in the origin only; this property makes this type of filters always stable. On the contrary the poles of purely recursive IIR filters are not in the origin making the design of IIR filter somehow more cumbersome because it is necessary to guarantee the filter stability.

For example in figure 9.5 is shown the frequency response of a 1<sup>st</sup> order purely recursive filter with a pole in  $z=2$  and unit DC gain. By comparison of this picture with the frequency response shown in figure 9.3 for an IIR filter with a pole in  $z=0.5$  it can be seen that the amplitude response is the same, while the phase response is different. But

looking at the impulse response of the two filters another important difference emerge: the filter with a pole in  $z=0.5$  is stable, while the filter with a pole in  $z=2$  is unstable. This behavior, as explained in chapter 7, is due the filter pole location outside the unite circle.

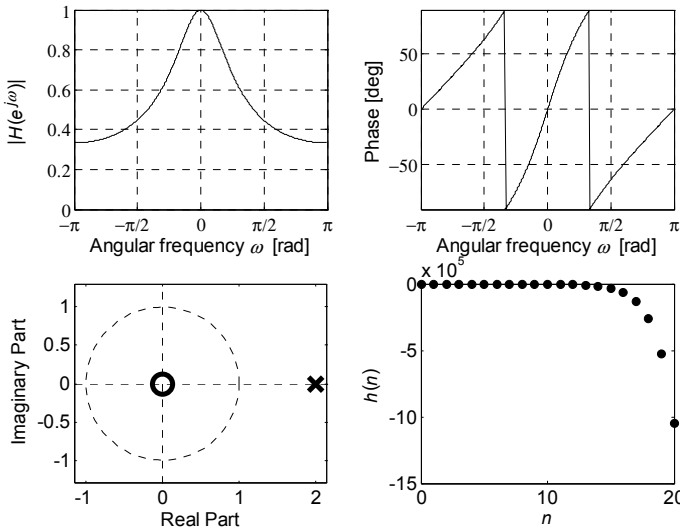


Figure 9.5. Frequency response, zero-pole diagram and impulse response for an unstable first order purely recursive IIR filter with a pole in  $z=2$

Another IIR filter noteworthy is the 2<sup>nd</sup> order purely recursive filter obtained adding one more state variable (i.e. a further unit shift block) to the first order purely recursive filter of figure 9.2, as shown in the block diagram of figure 9.6.

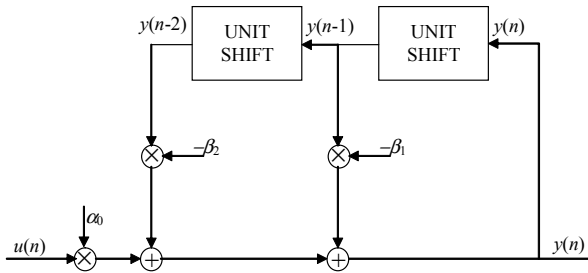


Figure 9.6. Block diagram of a second order purely recursive IIR filter

The difference equation implemented by the digital network of figure 9.10 is:

$$y(n) = \alpha_0 u(n) - \beta_1 y(n-1) - \beta_2 y(n-2) \tag{9.14}$$

The transfer function of the second order purely recursive filter is given by the z-transform of (9.14):

$$\begin{aligned}
 Y(z) &= \alpha_0 U(z) - \beta_1 Y(z)z^{-1} - \beta_2 Y(z)z^{-2} = \frac{\alpha_0}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}} U(z) \\
 &\quad \downarrow \\
 H(z) &= \frac{Y(z)}{U(z)} = \frac{\alpha_0}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}} = \frac{\alpha_0 z^2}{z^2 + \beta_1 z + \beta_2}
 \end{aligned} \tag{9.15}$$

giving for the transfer function of a second order purely recursive filter two zeros in the origin and two poles  $P_1$  and  $P_2$ , which are the roots of the denominator polynomial  $z^2 + \beta_1 z + \beta_2 = 0$ .

If the two coefficients  $\beta_1$  and  $\beta_2$  are real, the two poles  $P_1$  and  $P_2$  are a complex conjugate pair. Representing the poles in polar form as:

$$P_1 = r e^{j\theta}, \quad P_2 = r e^{-j\theta} \tag{9.16}$$

the coefficients  $\beta_1$  and  $\beta_2$  are given by:

$$z^2 + \beta_1 z + \beta_2 = (z - P_1)(z - P_2) \Rightarrow \begin{cases} \beta_1 = -P_1 - P_2 = -2r \cos \theta \\ \beta_2 = P_1 P_2 = r^2 \end{cases} \tag{9.17}$$

As an example the frequency response, together with the zero-pole diagram and the impulse response, for a second order purely recursive filter with real coefficients is shown in figure 9.7. The poles are placed at  $z = 0.5 e^{\pm j\pi/4}$  by setting  $\beta_1 = -\sqrt{2}/2$  and  $\beta_2 = 0.25$ . In order to maintain a unit DC gain the value of  $\alpha_0$  was forced to the following value:

$$\left| H(e^{j\omega}) \right|_{\omega=0} = \frac{\alpha_0}{1 + \beta_1 + \beta_2} = 1 \Rightarrow \alpha_0 = 1 + \beta_1 + \beta_2 \tag{9.18}$$

Usually the poles of a second order purely recursive filter are specified in terms of their radius  $r$ , called natural angular frequency and represented with the symbol  $\omega_n$ , and the damping ratio  $\xi$  defined as the ratio between their real part and the natural frequency:

$$\begin{cases} \omega_n = |P_1| = r \\ \xi = \frac{\text{Re}(P_1)}{\omega_n} = \frac{r \cos \theta}{r} = \cos \theta \end{cases} \quad (9.19)$$

Using this new parameters (9.19) for the poles classification with relationships (9.17) and (9.18), the transfer function  $H(z)$  in (9.15) can be rewritten as:

$$H(z) = \frac{(1 - 2\xi\omega_n + \omega_n^2)z^2}{z^2 - 2\xi\omega_n z + \omega_n^2} \quad (9.20)$$

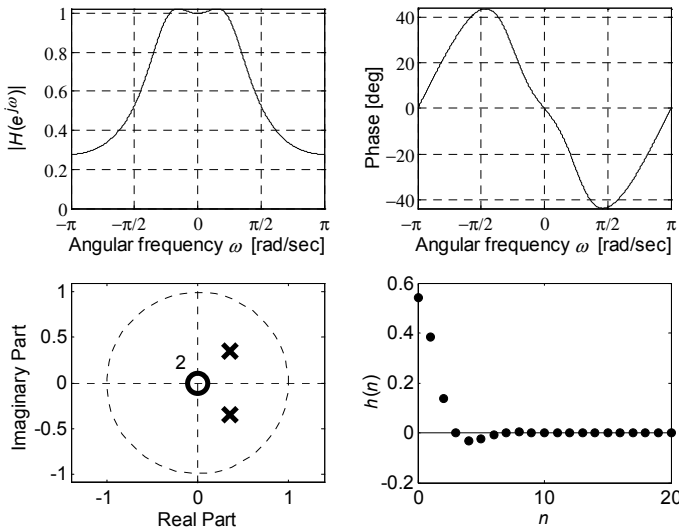


Figure 9.7. Frequency response, zero-pole diagram and impulse response for a second order purely recursive IIR filter with two poles in  $z=0.5e^{\pm j\pi/4}$

Figure 9.8 shows the complex conjugate pair location in the  $z$ -plane as a function of  $\omega_n$  and  $\xi$ . Since the angle  $\theta$  is in the range  $0 \div \pi$ , the damping ratio being the cosine of  $\theta$  must lay in the interval  $-1 \div 1$ .

Figure 9.9 shows the amplitude and impulse response for a 2<sup>nd</sup> order purely recursive filter with natural frequency  $\omega_n = 0.5$  and damping ratio  $\xi$  in the range  $0.1 \div 1$ . Looking at the impulse responses it is evident the significance of the damping ratio: the higher is its value the more damped is the response to an impulse input sequence. This behavior is also reflected on the shape of the magnitude frequency response.

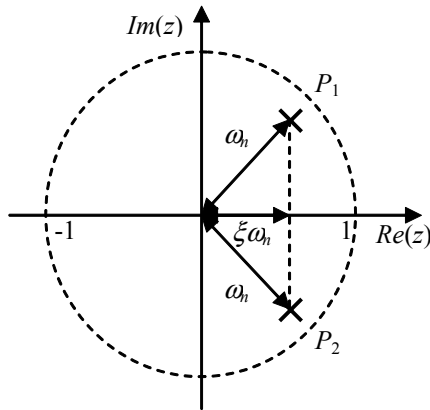


Figure 9.8. Complex conjugate poles location in the  $z$ -plane as a function of the natural angular frequency  $\omega_n$  and the damping ratio  $\xi$

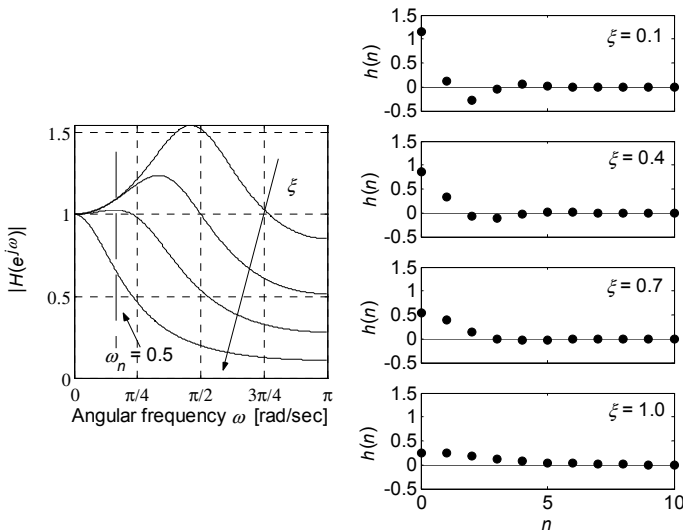


Figure 9.9. Amplitude and impulse responses for a second order purely recursive IIR filter with  $\omega_n=0.5$  and damping ratio  $\xi$  increasing from 0.1 to 1

IIR FILTERS DESIGN FROM ANALOG SPECIFICATIONS.

The simplest way for the design of IIR filters meeting some prescribed specifications is carrying out the following procedure:



- Convert the discrete domain specs in the continuous analog domain;
- Design the analog filter;
- Convert the analog filter from the continuous analog domain to the discrete domain.

Before stepping into the detail of the analog to discrete mapping, let us first introduce the characteristics of the three most popular analog filter prototypes.

### Butterworth filters

The square magnitude response of a  $N^{\text{th}}$  order Butterworth filter is

$$|G(\Omega)|^2 = \frac{1}{1 + \left(\frac{j\Omega}{j\Omega_c}\right)^{2N}} \tag{9.21}$$

where  $\Omega_c$  is the -3dB cutoff angular frequency. Figure 9.10 shows the Butterworth magnitude response for  $N=2, 5$  and  $8$ .

Butterworth filters have the following properties:

- maximally flat pass-band, i.e. the first  $2N-1$  derivatives of (9.21) are zero at DC;
- there are no ripples in both the pass-band and the stop-band, i.e. the magnitude response is monotonic decreasing with  $\Omega$ .

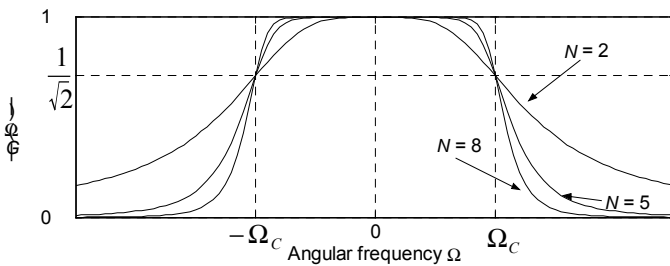


Figure 9.10. The monotonic magnitude frequency responses of the Butterworth low-pass filters of order  $N=2, 5$  and  $8$

The transfer function  $G(s)$  can be obtained replacing  $j\Omega$  with  $s$  in (9.21):

$$|G(s)|^2 = G(s)G(-s) = \frac{1}{1 + \left(\frac{s}{j\Omega_C}\right)^{2N}} \tag{9.22}$$

The  $2N$  poles of  $|G(s)|^2$  are:

$$\frac{s}{j\Omega_C} = \sqrt[2N]{-1} = e^{j\left(\frac{\pi}{2N} + k\frac{2\pi}{2N}\right)} \Rightarrow s = \Omega_C e^{j\left(\frac{\pi}{2} + \frac{\pi}{2N} + k\frac{2\pi}{2N}\right)}, \quad k = 0, 1, \dots, 2N - 1 \tag{9.23}$$

as shown in figure 9.11a. The poles (9.23), laying on the circle of radius  $\Omega_C$ , are the poles of both  $G(s)$  and  $G(-s)$ . If the filter is causal and stable the poles of  $G(s)$  are those with negative real part as shown in figure 9.11b.

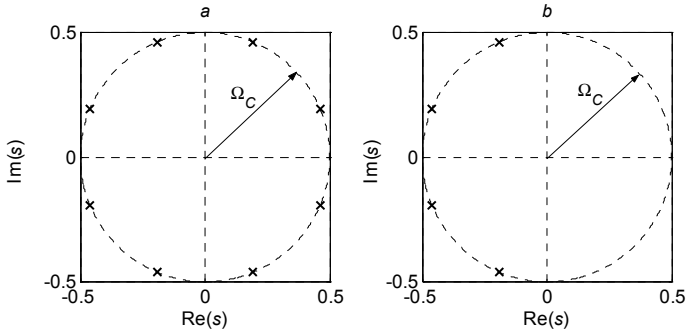


Figure 9.11. (a) – poles of  $|G(s)|^2 = G(s)G(-s)$  (b) – poles of  $G(s)$

### Chebyshev filters

Since filter specifications allow a certain ripple level in the pass and stop bands, a more efficient design can be attained with the Chebyshev filters, characterised by having a not-monotonic magnitude frequency response. There are two types of Chebyshev filters: type I filters are characterized by ripple in the pass-band only, while type II filters have ripple in the stop-band only. The squared magnitude frequency response is given by (9.24) and (9.25) for the type I and type II filters, respectively.

$$|G(\Omega)|^2 = \frac{1}{1 + \mathcal{E}[T_N(\Omega / \Omega_C)]^2} \tag{9.24}$$

$$\mathcal{E} = (1 - \delta)^{-2} - 1$$

$$|G(\Omega)|^2 = \frac{1}{1 + \varepsilon^{-1} [T_n(\Omega_c / \Omega)]^{-2}} \tag{9.25}$$

$$\varepsilon = (1 - \delta)^{-2} - 1$$

In (9.24) and (9.25)  $\delta$  is the ripple level either in the pass-band or in the stop-band. The function  $T_N(x)$  is the Chebyshev polynomial of order  $N$ :

$$T_N(x) = \begin{cases} \cos[N \arccos(x)] & |x| \leq 1 \\ \text{Ch}[N \text{arcCh}(x)] & |x| > 1 \end{cases} \tag{9.26}$$

Figure 9.12 shows the comparison between a Butterworth filter and Type I and II Chebyshev filters of the same order: the ripple in the Chebyshev filters allows a sharper transition between the pass-band and the stop-band.

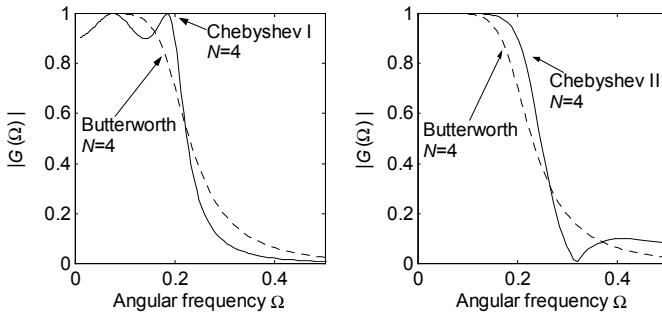


Figure 9.12. Comparison between the magnitude responses of Butterworth and Chebyshev I and II low pass filters of the same order

### Elliptic filters

Elliptic filters are even more efficient than Chebyshev since with this filters ripple is possible in both the pass-band and stop-band. The magnitude squared of the frequency response is given by:

$$|G(j\Omega)|^2 = \frac{1}{1 + \nu [U_N(\Omega / \Omega_p)]^2} \tag{9.27}$$

$$\nu = \frac{1}{(1 - \delta)^2} - 1$$

with  $\delta$  the ripple level and  $U_N(x)$  the Jacobian elliptic function of order  $N$ :

$$U_N(x) = \int_0^x \frac{dy}{\sqrt{(1-y^2)(1-N^2y^2)}} \tag{9.28}$$

Figure 9.13 shows a comparison between Butterworth, Chebyshev and elliptic filters with the same ripple level and transition bandwidth. The result is that the elliptic filter, allowing ripple in both the stop and pass bands, meet the specifications with the lowest order.

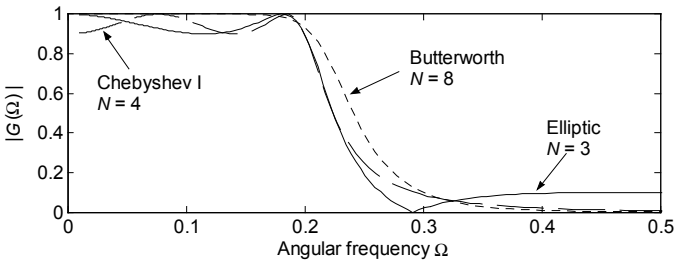


Figure 9.13. Comparison between the magnitude responses of Butterworth, Chebyshev I and elliptic filters with the same transition band and ripple level

### Analog to discrete domain mapping

The mapping of the continuous system to the discrete domain can be accomplished in two different ways. A first possibility is to define a discrete system such that its response to samples of a given class of input functions is equal to the samples of the response of the specified continuous system, as shown in the continuous frequency domain  $\Omega$  in the block diagram of figure 9.14.

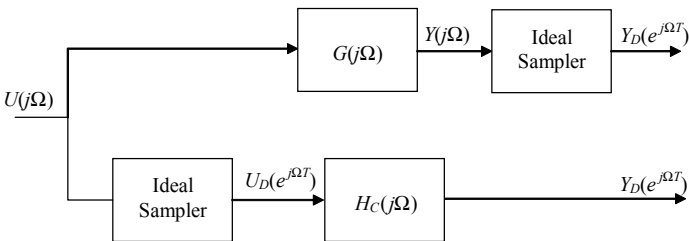


Figure 9.14. Impulse response invariance synthesis

Another possibility for the conversion from the continuous to the discrete domain consists in the definition of a mapping  $s=s(z)$  between the  $s$ -domain and the  $z$ -domain and, through this mapping, convert the continuous transfer function  $G(s)$  to the discrete system transfer function  $H_D(z)$  setting  $H_D(z) = G[s(z)]$ .

This two different approaches will be discussed in the following two paragraphs.

### Input response invariance

A common method for the synthesis of a discrete system transfer function  $H_C(j\Omega)$  is by conversion of a given transfer function  $G(j\Omega)$  of a continuous system in such a way that the responses of the two system to a special class of input signals are the same, as shown in the block diagram of figure 9.14. Typical input signal are the impulse, the unit step and the ramp.

If  $U(j\Omega)$  is the Fourier transform of the input signal to the continuous system with transfer function  $G(s)$ , the Fourier transform  $Y(j\Omega)$  of the output signal is:

$$Y(j\Omega) = G(j\Omega)U(j\Omega) \quad (9.29)$$

Once sampled, with sampling period  $T$ , the spectrum of the continuous system response is given by:

$$\begin{aligned} Y_D(e^{j\Omega T}) &= \frac{1}{T} \sum_{n=-\infty}^{+\infty} Y\left(j\Omega + jn \frac{2\pi}{T}\right) = \\ &= \frac{1}{T} \sum_{n=-\infty}^{+\infty} G\left(j\Omega + jn \frac{2\pi}{T}\right) U\left(j\Omega + jn \frac{2\pi}{T}\right) \end{aligned} \quad (9.30)$$

If  $H_C(j\Omega)$  is the frequency response of the discrete system in the continuous domain, the Fourier transform of the discrete system output must be equal to (9.30) and given by:

$$Y_D(e^{j\Omega T}) = H_C(j\Omega)U_D(e^{j\Omega T}) \quad (9.31)$$

From (9.31) and (9.30) it results the following expression for the discrete system frequency response satisfying the input response invariance:

$$H_C(j\Omega) = \frac{1}{TU_D(e^{j\Omega T})} \sum_{n=-\infty}^{+\infty} G\left(j\Omega + jn \frac{2\pi}{T}\right) U\left(j\Omega + jn \frac{2\pi}{T}\right) \quad (9.32a)$$

Considering as input signal an impulse (impulse invariance) expression (9.32a) simplifies to:

$$H_c(j\Omega) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} G\left(j\Omega + jn \frac{2\pi}{T}\right) \tag{9.32b}$$

If the condition:

$$G(j\Omega) = 0 \quad \text{for} \quad |\Omega| \geq \frac{\pi}{T} \tag{9.33}$$

is satisfied then there is no aliasing and the two transfer functions in (9.32) *a* and *b* are given by:

$$H_c(j\Omega) = \frac{U(j\Omega)}{TU_D(e^{j\Omega T})} G(j\Omega) \quad \text{for} \quad |\Omega| < \frac{\pi}{T} \tag{9.34a}$$

$$H_c(j\Omega) = \frac{1}{T} G(j\Omega) \quad \text{for} \quad |\Omega| < \frac{\pi}{T} \tag{9.34b}$$

The above conditions are satisfied when *T* is small enough to avoid aliasing in *G(jΩ)*; but according to (7.61) when *T*→0 it must be *U<sub>D</sub>(e<sup>jΩT</sup>)*≈*U(jΩ)/T* and (9.43a) simplifies to:

$$H_c(j\Omega) \approx G(j\Omega) \quad \text{for} \quad T \rightarrow 0 \tag{9.35}$$

Usually condition (9.33) is not met and therefore the invariance of the input response forces the transfer functions of the discrete and continuous systems to be different in the interval  $|\Omega| < \pi/T$ , although (9.35) shows that in the limit *T*→0 it must be *H<sub>C</sub>(jΩ)*→*G(jΩ)*.

The previous limit is not valid anymore if we consider the system response to an input impulse. In this case (9.34b) shows that when *T*→0 it must be *H<sub>C</sub>(jΩ)*→*G(jΩ)/T*. Thus the impulse invariance frequency response for short sampling periods *T* is characterized by a very large magnitude response; this is the reason why often it is necessary to implement a filter with frequency response *TH<sub>C</sub>(jΩ)* instead than *H<sub>C</sub>(jΩ)*.

In general to manage equation (9.32) *a* or *b* for the input response invariance design is quite cumbersome, but it can be avoided by considering the partial fraction expansion of the continuous system transfer function to elementary functions and using the table for the Laplace and *z* transforms for the inversion as shown in the following two examples.

**Example 1:** *Impulse invariance* - If for example *G(s)* has no repeated poles the partial fraction expansion has the form:

$$G(s) = \sum_{k=1}^N \frac{A_k}{s - s_k} \quad (9.36)$$

and the impulse response  $y(t)$  is the sum of  $N$  exponentials:

$$y(t) = \left( \sum_{k=1}^N A_k e^{s_k t} \right) I(t) \quad (9.37)$$

The samples of the impulse response (9.37) are:

$$y(nT) = \left( \sum_{k=1}^N A_k e^{s_k nT} \right) I(nT) \quad (9.38)$$

and the sequence associated to the samples of the impulse response of the continuous system is given by:

$$y_D(n) = \left( \sum_{k=1}^N A_k e^{s_k nT} \right) I(nT) \quad (9.39)$$

The  $z$ -transform of the impulse response (9.39), which by definition is the discrete system transfer function, can be derived by the  $z$ -transform table 7.1 and expressed as:

$$H_D(z) = \sum_{k=1}^N \frac{A_k}{1 - e^{s_k T} z^{-1}} \quad (9.40)$$

Equation (9.40) shows that the poles  $s_k$  of  $G(s)$  are mapped to  $e^{s_k T}$  in the  $z$ -domain; this mapping assures that a stable continuous system is mapped to a stable discrete system, since if  $|s_k| < 0$  it must be  $|e^{s_k T}| < 1$ .

The Fourier transform in the continuous domain  $\Omega$  of the discrete system is:

$$H_C(j\Omega) = H_D(e^{j\Omega T}) = \sum_{k=1}^N \frac{A_k}{1 - e^{s_k T} e^{-j\Omega T}} \quad (9.41)$$

Figure 9.15 compares the magnitude response of the continuous system with the magnitude response of the discrete system synthesized with the impulse invariance method when the sampling period is  $T = 0.2$  s and the continuous transfer function is:

$$G(s) = \frac{3}{3s + 10} - \frac{2}{s + 1} \tag{9.42}$$

It is noteworthy that the comparison, according to (9.34b), is between  $|G(j\Omega)|$  and  $|TH_C(j\Omega)|$ . The two frequency responses are different and in particular the discrete system transfer function is periodic with period twice the Nyquist angular frequency  $\Omega_N = \pi/T$ . In the time domain instead the two impulse responses shown in figure 9.15 are the same as imposed by the design method.

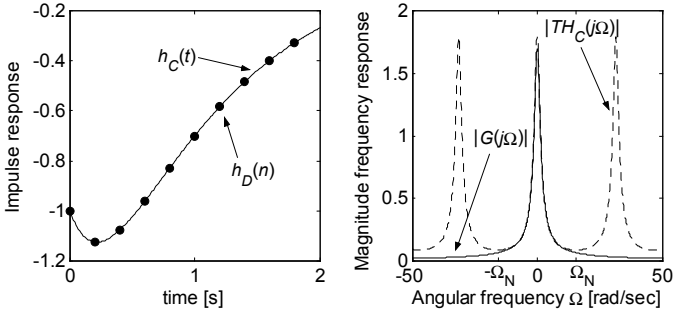


Figure 9.15. Comparison between the impulse and the magnitude responses of the continuous system (9.42) and the impulse invariance discrete system (9.41)

**Example 2: Step invariance** - If as in the previous example the transfer function  $G(s)$  has no repeated poles, the partial fraction expansion of the Laplace transform for the continuous system step response has the following form:

$$Y(s) = \frac{1}{s} G(s) = \frac{1}{s} \sum_{k=1}^N \frac{A_k}{s - s_k} \tag{9.43}$$

The step response in the time domain can be derived inverting the Laplace transform (9.43):

$$y(t) = \left( \sum_{k=1}^N \frac{A_k}{s_k} (e^{s_k t} - 1) \right) I(t) \tag{9.44}$$

The sequence composed by samples of the step response (9.44) is expressed by:



$$y_D(n) = y(nT) = \left( \sum_{k=1}^N \frac{A_k}{S_k} (e^{s_k nT} - 1) \right) I(t) \tag{9.45}$$

The step-invariance discrete system transfer function results from the ration between the  $z$ -transform of (9.45) and the  $z$ -transform of the step sequence  $I(n)$ :

$$H_D(z) = \frac{Y_D(z)}{\frac{z}{z-1}} = \frac{z-1}{z} \sum_{k=1}^N \frac{A_k}{S_k} \left[ \frac{z}{z-e^{s_k T}} - \frac{z}{z-1} \right] = \sum_{k=1}^N \frac{A_k}{S_k} \frac{(e^{s_k T} - 1)z^{-1}}{1 - e^{s_k T} z^{-1}} \tag{9.46}$$

Figure 9.16 compares the magnitude response of the continuous system with the magnitude response of the discrete system synthesized with the step invariance method, in case of a sampling period  $T = 0.2$  s and the continuous transfer function (9.42) of the previous example.

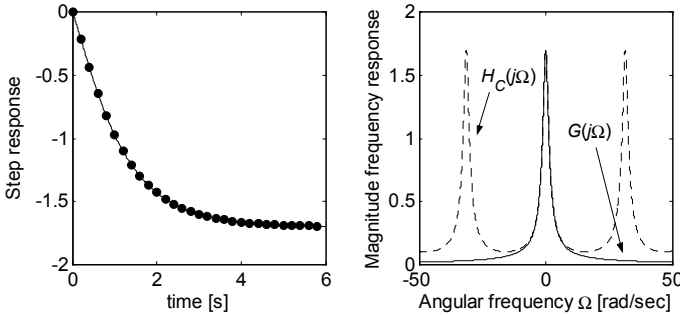


Figure 9.16. Comparison between the step and the magnitude responses of the continuous system (9.42) and the step invariance discrete system (9.46)

It is noteworthy that the comparison in this case, according to (9.35), is not between  $|G(j\Omega)|$  and  $|TH_C(j\Omega)|$  as for the impulse invariance synthesis but between  $|G(j\Omega)|$  and  $|H_C(j\Omega)|$ . The two frequency responses are different and in particular the discrete system transfer function is periodic with period twice the Nyquist angular frequency  $\Omega_N = \pi/T$ . In the time domain instead the two step responses shown in figure 9.16 are the same as imposed by the design method.

*Frequency domain design*

Another possibility for the conversion of a continuous system transfer function  $G(s)$  is to define a transformation

$$s' = s(z) \tag{9.47}$$

from the  $z$ -domain to the  $s$ -domain so that the discrete system transfer function is given by:

$$H_D(z) = G[s(z)] \tag{9.48}$$

Several ways for defining the mapping (9.47) are available. The methodologies described here are all based on numerical integration formulas approximating the continuous integration.

In the discrete domain any numerical integration algorithm can be expressed in terms of its  $z$ -transform  $H_D(z)$ . On the other side in the continuous domain context the integration is expressed, in terms of its Laplace transform, as  $1/s$ . Putting together the two expressions the following  $z$  to  $s$  mapping outcomes

$$\frac{1}{s'} = H_D(z) \Rightarrow s' = \frac{1}{H_D(z)} \tag{9.49}$$

FORWARD INTEGRATION

It is natural to obtain a first  $z$  to  $s$  transformation by considering the Euler's or forward rectangles integration formula reported below and motivated graphically in figure 9.17:

$$y(k) = y(k-1) + u(k-1) \cdot T \tag{9.50}$$

where  $T$  is the sampling period.

The  $z$ -transform of the digital integrator (9.50) is given by

$$H_D(z) = \frac{Tz^{-1}}{1-z^{-1}} \tag{9.51}$$

Applying (9.49) to the above expression of  $H_D(z)$  it follows the transformation:

$$\begin{aligned} s' &= \frac{z-1}{T} \\ z &= 1 + s'T \end{aligned} \tag{9.52}$$

The mapping of the stability region in the s-domain on the z-plane for the transformation (9.52) is shown in figure 9.18, where it can be seen that with the Euler's integration formula a stable continuous system can be transformed in an unstable discrete system.

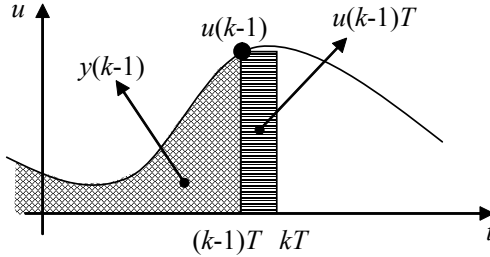


Figure 9.17. Euler's integration formula: the integral  $y(k)$  at step  $k$  is computed adding the rectangle area  $u(k-1)T$  to the integral  $y(k-1)$  at the previous step  $k-1$

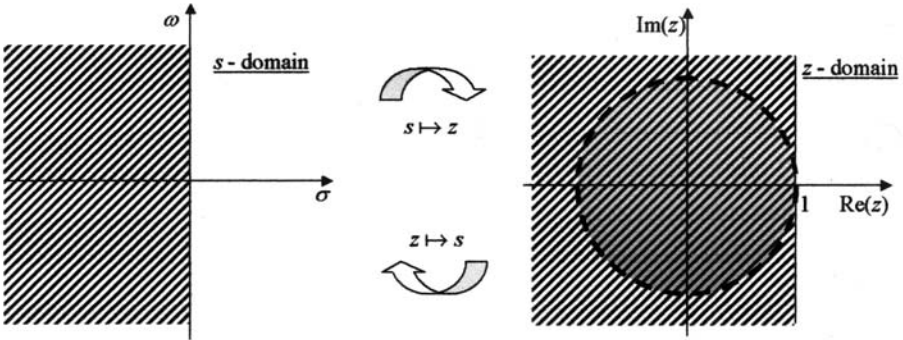


Figure 9.18. Mapping of the stability region  $|s| < 0$  in the s-domain on the z-plane for the forward integration formula (9.52)

The goodness of the approximation to a pure integrator obtained with Euler's approximation (9.52) can be analyzed in the frequency domain. The theoretical continuous time frequency response of a pure integrator is given by:

$$G(j\Omega) = \frac{1}{j\Omega} \tag{9.53}$$

while the frequency response  $H_C(j\Omega)$  can be derived taking the Fourier transform of the discrete integrator (9.51):

$$H_C(j\Omega) = H_D(e^{j\Omega T}) = \frac{T e^{-j\Omega T}}{1 - e^{-j\Omega T}} = -j e^{-j\frac{\Omega T}{2}} \frac{T/2}{\sin(\Omega T/2)} \quad (9.54)$$

The magnitude and phase of the two frequency responses  $H_C(j\Omega)$  and  $G(j\Omega)$  are shown in figure 9.19 (for  $T = 1$  s) and expressed by the following relationships:

$$\begin{cases} |G(j\Omega)| = \frac{1}{|\Omega|} \\ \angle G(j\Omega) = -\frac{\pi}{2} - \angle \Omega \end{cases} \quad \begin{cases} |H_C(j\Omega)| = \frac{T/2}{|\sin(\Omega T/2)|} \\ \angle H_C(j\Omega) = -\frac{\pi}{2} - \frac{\Omega T}{2} - \angle \sin(\Omega T/2) \end{cases} \quad (9.55)$$

From the responses reported in figure 9.19 it is evident the distortion in phase and magnitude due to the approximation of the Euler's integration formula.

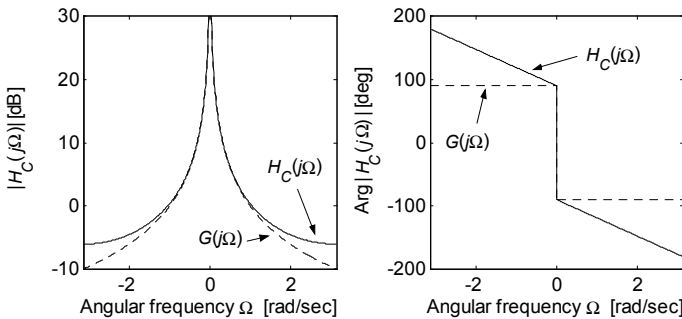


Figure 9.19. Magnitude and phase frequency responses for the forward integration formula with sampling period  $T = 1$  s

BACKWARD INTEGRATION

Another possibility for the numerical approximation of the pure integration is given by the backward rectangles integration formula (see figure 9.20):

$$y(k) = y(k-1) + u(k) \cdot T \quad (9.56)$$

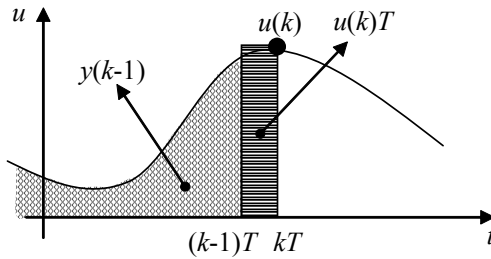


Figure 9.20. Backward rectangles integration formula: the integral  $y(k)$  at step  $k$  is computed adding the rectangle area  $u(k)T$  to the integral  $y(k-1)$  at the previous step  $k-1$

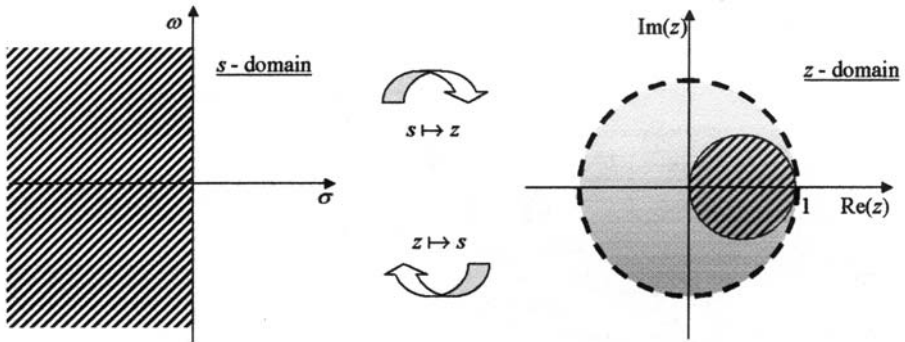


Figure 9.21. Mapping of the stability region  $|s| < 0$  in the  $s$ -domain on the  $z$ -plane for the backward rectangles integration formula (9.56)

In this case the transfer function of the discrete integrator is:

$$H_D(z) = \frac{T}{1 - z^{-1}} \tag{9.57}$$

which, according to (9.49), defines the  $z$ - $s$  mapping:

$$s' = \frac{z-1}{zT} \tag{9.58}$$

$$z = \frac{1}{1 - s'T}$$

The mapping of the stability region in the  $s$ -domain on the  $z$ -plane for the transformation (9.58) is shown in figure 9.21. This type of transformation is better than the forward integration formula because stable continuous systems are always transformed in a stable discrete systems. There are however also unstable continuous systems that are transformed into stable discrete systems.

The frequency response in the continuous domain  $\Omega$  of the discrete filter derives directly from (9.57)

$$H_C(j\Omega) = H_D(e^{j\Omega T}) = \frac{T}{1 - e^{-j\Omega T}} = -je^{j\frac{\Omega T}{2}} \frac{T/2}{\sin(\Omega T/2)} \quad (9.59)$$

and the comparison with the frequency response of a pure integrator is shown in figure 9.22 according to the following results:

$$\begin{cases} |G(j\Omega)| = \frac{1}{|\Omega|} \\ \angle G(j\Omega) = -\frac{\pi}{2} - \angle\Omega \end{cases} \quad \begin{cases} |H_C(j\Omega)| = \frac{T/2}{|\sin(\Omega T/2)|} \\ \angle H_C(j\Omega) = -\frac{\pi}{2} + \frac{\Omega T}{2} - \angle \sin(\Omega T/2) \end{cases} \quad (9.60)$$

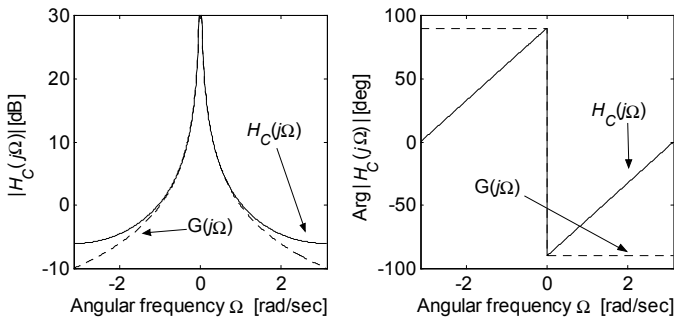


Figure 9.22. Magnitude and phase frequency responses for a backwards rectangles formula discrete integrator with sampling period  $T = 1$  s

As shown in figure 9.22 the backward rectangles integration formula is not a better approximation of a pure integrator than the forward transformation. Even at low frequency although the amplitude response of the discrete integrator approximate very well the response of the pure integrator, there is still a strong difference in the two phase responses.

CENTRAL INTEGRATION

A third possibility for the approximation of a pure integral is to use a further rectangles integration formula which is in between the two formulas (backward and forward) seen before as shown in graphically in figure 9.23. The finite difference equation resultant from the computation of figure 9.23 is:

$$y(k) = y(k-2) + u(k-1) \cdot 2T \tag{9.61}$$

The z-transform for this type of digital integrator is given by

$$H_D(z) = \frac{2Tz^{-1}}{1-z^{-2}} \tag{9.62}$$

which, according to (9.49), defines the z-s mapping:

$$s' = \frac{z^2 - 1}{2zT} \tag{9.63}$$

$$z = s'T \pm \sqrt{(s'T)^2 + 1}$$

The mapping of the stability region in the s-domain on the z-plane for the transformation (9.63) is shown in figure 9.24, where it can be seen that with the central rectangles integration system formula (9.61) a stable continuous system can be transformed in an unstable discrete system and an unstable continuous system can be transformed in a stable discrete system.

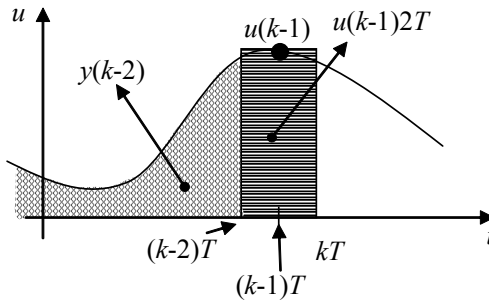


Figure 9.23. Central rectangles integration formula: the integral  $y(k)$  at step  $k$  is computed adding the rectangle area  $2u(k-1)T$  to the integral  $y(k-2)$  at step  $k-2$

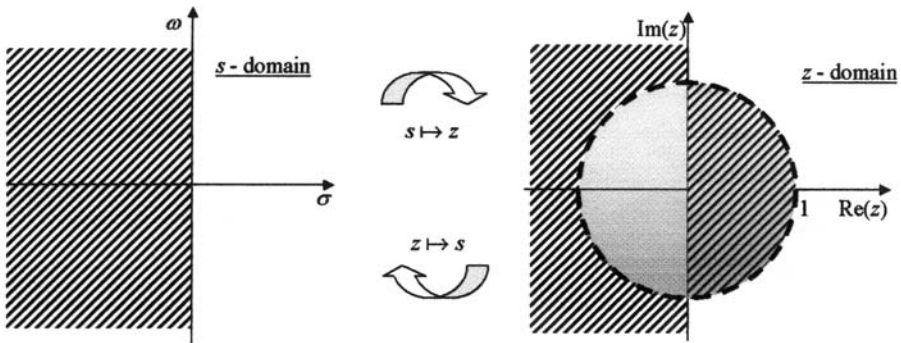


Figure 9.24. Mapping of the stability region  $|s| < 0$  in the  $s$ -domain on the  $z$ -plane for the central rectangles integration formula (9.61)

The comparison between the numerical and the continuous integration can be done comparing the pure integration Fourier transform (9.53) with the frequency response  $H_C(j\Omega)$  associated to the  $z$ -transform (9.62):

$$H_C(j\Omega) = H_D(e^{j\Omega T}) = \frac{2Te^{-j\Omega T}}{1 - e^{-j2\Omega T}} = -j \frac{T}{\sin(\Omega T)} \quad (9.64)$$

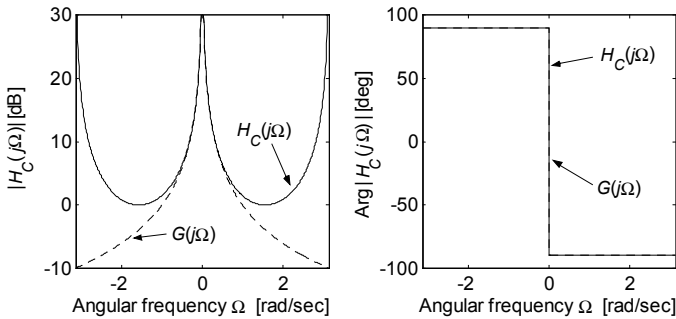


Figure 9.25. Magnitude and phase frequency responses for the central rectangle integration formula (9.61) with sampling period  $T = 1$  s

The magnitude and phase of the two frequency responses  $H_C(j\Omega)$  and  $G(j\Omega)$  are shown in figure 9.25 (for  $T = 1$  s) and expressed by the following relationships:



$$\begin{cases} |G(j\Omega)| = \frac{1}{|\Omega|} & |H_c(j\Omega)| = \frac{T}{|\sin(\Omega T)|} \\ \angle G(j\Omega) = -\frac{\pi}{2} - \angle \Omega & \angle H_c(j\Omega) = -\frac{\pi}{2} - \angle \sin(\Omega T) \end{cases} \quad (9.65)$$

In this case the transformation does not produce any phase distortion, but the magnitude response approximate the pure integrator response only at low frequency if compared with the two previous digital integrators.

SIMPSON INTEGRATION

A better approximation can be obtained considering a discrete filter which adopts the Simpson integration formula (see figure 9.26):

$$y(k) = y(k-1) + \frac{u(k) + u(k-1)}{2} \cdot T \quad (9.66)$$

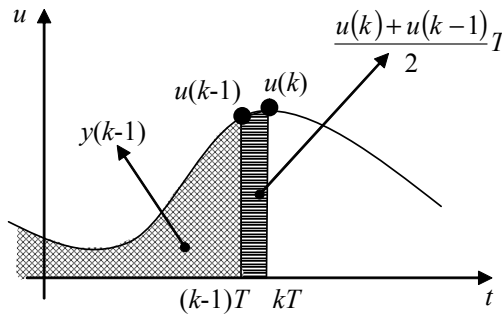


Figure 9.26. Simpson integration formula: the integral  $y(k)$  at step  $k$  is computed adding the trapezium area  $[u(k)+u(k-1)]T/2$  to the integral  $y(k-1)$  at the previous step  $k-1$

Applying the same analysis as before the frequency response in the discrete domain  $\omega$  is:

$$H_D(z) = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} \quad (9.67)$$

At (9.67) corresponds to the following frequency response in the continuous domain  $\Omega$ :

$$H_c(j\Omega) = \frac{T}{2} \frac{1 + e^{-j\Omega T}}{1 - e^{-j\Omega T}} = -j \frac{T}{2} \frac{1}{\tan\left(\frac{\Omega T}{2}\right)} \tag{9.68}$$

The magnitude and phase frequency responses of the Fourier transforms (9.68) are given by

$$\begin{cases} |G(j\Omega)| = \frac{1}{|\Omega|} \\ \angle G(j\Omega) = -\frac{\pi}{2} - \angle \Omega \end{cases} \quad \begin{cases} |H_c(j\Omega)| = \frac{T}{2 \left| \tan\left(\frac{\Omega T}{2}\right) \right|} \\ \angle H_c(j\Omega) = -\frac{\pi}{2} - \angle \tan\left(\frac{\Omega T}{2}\right) \end{cases} \tag{9.69}$$

Figure 9.27. shows graphically the frequency responses (9.69). It is clear from this figure that in this case the phase response of the numerical integrator is equal to the theoretical one and the magnitude response is in better agreement that with the central rectangles integration formula used before.

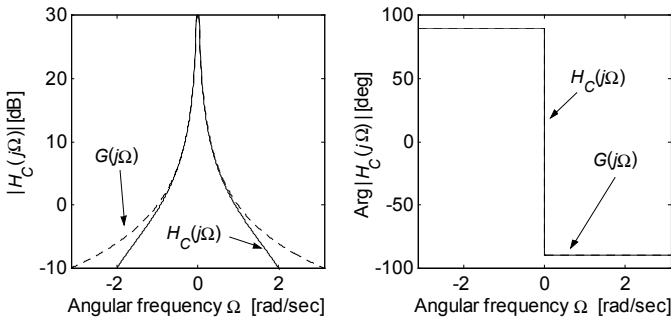


Figure 9.27. Magnitude and phase frequency responses for a discrete integrator based on the Simpson integration formula with sampling period  $T = 1$  s

The z-transform (9.67), according to (9.49), defines the z-s mapping:

$$s' = \frac{2}{T} \frac{z-1}{z+1} \tag{9.70}$$

$$z = \frac{2+s'T}{2-s'T}$$

The transformation (9.70) is known as the *bilinear* or *Tustin's* transformation. Figure 9.28 shows the mapping of the stability region in the *s*-domain on the *z*-plane for the bilinear transformation (9.70). With this formula the left half *s*-plane is mapped into the interior of the unit disc in the *z*-domain. Therefore stable continuous systems are transformed in a stable discrete system and unstable continuous systems are transformed in unstable discrete systems.

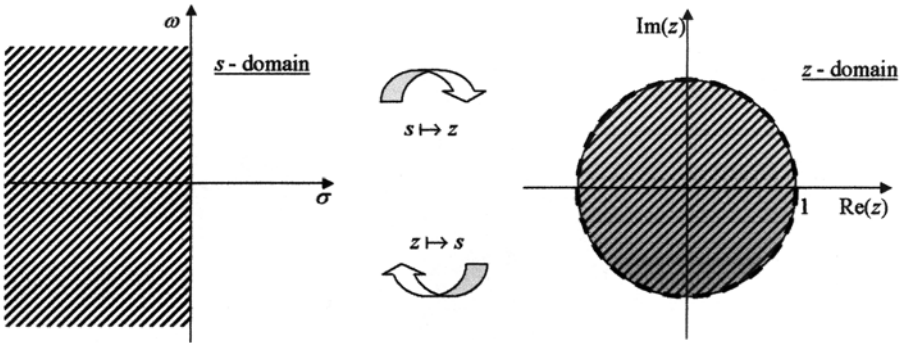


Figure 9.28. Mapping of the stability region  $|s| < 0$  in the *s*-domain on the *z*-plane for the bilinear transformation

A problem with the transformation just discussed is that the frequency axis is distorted. According to (9.53) and (9.68) the distortion is given by:

$$j\Omega' = \frac{1}{H_c(j\Omega)} \Rightarrow \Omega' = \frac{2}{T} \tan\left(\frac{\omega}{2}\right) \tag{9.71}$$

This effect, called frequency warping, is shown graphically in figure 9.29 where a continuous low-pass filter is approximated with a discrete low-pass filter. The approximation, due to the frequency warping, is good at low frequencies only and worsen increasing the frequency.

It is possible to modify the transformation (9.70) in order to eliminate the frequency distortion at a specific frequency  $\omega_1$  adopting the following *s*-*z* transformation:

$$s' = \frac{\omega_1}{\tan\left(\omega_1 \frac{T}{2}\right)} \frac{z-1}{z+1} \tag{9.72}$$

Using (9.72) the distortion at  $\omega_1$  will be eliminated, but there will be however a distortion at the other frequencies.

Figure 9.30 shows the magnitude frequency response of different digital filters obtained by impulse invariance and Tustin's methods approximating a 4<sup>th</sup> order low pass elliptic filter. The sampling period is 0.01 s in both cases, i.e. the Nyquist angular frequency is 314 rad/s. It is evident the frequency warping for the filter obtained by the Tustin's method.

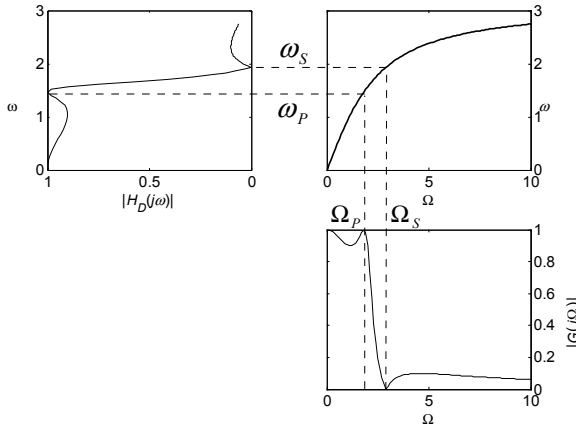


Figure 9.29. Frequency warping due to the bilinear transformation

### Frequency transformation from low-pass filters to band-pass and band-stop filters

The design of analog high-pass, band-pass and band-stop filters is typically done by firstly computing the parameters of a prototype low-pass filter and secondly by means of proper algebraic transformations to convert the analog low-pass filter to a high-pass, band-pass or band-stop one. A similar design method can be adopted in the digital domain.

This approach has the advantage to allow the design of high-pass or band-stop filters with the input invariance method. As matter of fact because of the aliasing it would be impossible to transform the transfer function of these types of filters; but it is possible to design with this method a low pass filter and subsequently to transform it to a frequency selective filter.

Since the IIR filter transfer functions are rational functions of  $z$ , the transformation in the digital domain can be formalized as a mapping between two complex planes,  $z$  and  $Z$ , defined by the function

$$Z=T(z) \tag{9.73}$$

If  $H_{LP}(z)$  is the transfer function of the prototype discrete low-pass filter, the transfer function of the frequency selective filter is given by

$$H(z) = H_{LP}[T(z)] \tag{9.74}$$

The transformation  $T(z)$  must guarantee that if the prototype transfer function  $H_{LP}(z)$  is a rational transfer function of causal and stable system than also  $H(z)$  must be a rational transfer function of a stable and causal system. This mapping must satisfy the following conditions:

- the unit circle in the  $z$ -plane is transformed in the unit circle in the  $Z$ -plane;
- the interior of the unit circle in the  $z$ -plane is mapped on the interior of the unit circle in the  $Z$ -plane;
- $T(z)$  must be a rational function.

The most general form for such a transformation is:

$$T(z) = \pm \prod_{k=1}^N \frac{1 - \alpha_k z^{-1}}{z^{-1} - \alpha_k} \tag{9.75}$$

In table 9.1 are reported all the transformations (9.75) from the low-pass prototype filter to any other frequency selective filter.

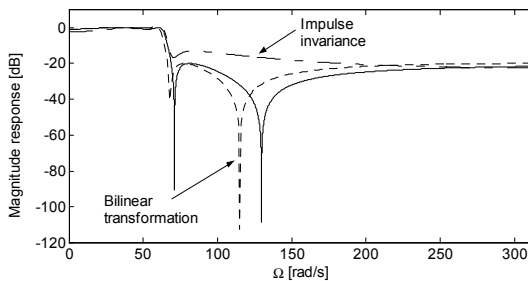


Figure 9.30. Magnitude response of a 4<sup>th</sup> order elliptic filter and discrete approximations obtained with the impulse invariance method and the bilinear transformation

Table 9.1. Transformation from a prototype low pass filter with cutoff frequency  $\theta_p$

Filter type	$T(z)$	Design parameters
Low-pass	$Z = \frac{1 - \alpha z^{-1}}{z^{-1} - \alpha}$	$\omega_p =$ cutoff frequency $\alpha = \frac{\sin\left(\frac{\theta_p - \omega_p}{2}\right)}{\sin\left(\frac{\theta_p + \omega_p}{2}\right)}$
High-pass	$Z = -\frac{1 + \alpha z^{-1}}{z^{-1} + \alpha}$	$\omega_p =$ cutoff frequency $\alpha = -\frac{\cos\left(\frac{\theta_p + \omega_p}{2}\right)}{\cos\left(\frac{\theta_p - \omega_p}{2}\right)}$
Band-pass	$Z = -\frac{(k-1)z^{-2} - 2\alpha kz^{-1} + k+1}{(k+1)z^{-2} - 2\alpha kz^{-1} + k-1}$	$\omega_{p1} =$ lower cutoff frequency $\omega_{p2} =$ upper cutoff frequency $\alpha = -\frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \cot\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$
Band-stop	$Z = -\frac{(1-k)z^{-2} - 2\alpha z^{-1} + 1+k}{(1+k)z^{-2} - 2\alpha z^{-1} + 1-k}$	$\omega_{p1} =$ lower cutoff frequency $\omega_{p2} =$ upper cutoff frequency $\alpha = -\frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \tan\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$

IIR DESIGN BY NUMERICAL OPTIMIZATION.

The IIR filter design methods shown in the previous paragraph were based on the transformation of a prototype analog transfer function. This approach to the synthesis of discrete filters transfer function is not always appropriate either because the specification are given directly in the frequency domain or because the prototype analog transfer functions do not satisfy specific constraints.

In this cases it is possible to proceed to the computation of the IIR filter coefficients using numerical optimization techniques. This approach is based on the definition of a loss function  $L$  which is a function of the unknown filter parameters. The optimization process consist in the minimization of  $L$  subject to the possible constraint imposed by the design method.

If  $H_{SP}(e^{j\omega})$  is the specified IIR filter frequency response and  $H(e^{j\omega}; \mathbf{m})$  is the actual approximation of  $H_{SP}(e^{j\omega})$  dependent on the unknown IIR design parameter vector  $\mathbf{m}$ , typical optimization loss functions have the form of a mean error of order  $p$ . For example if the design is in terms of the amplitude frequency response, the loss function is:

$$L'(\mathbf{m}) = \int_{-\pi}^{+\pi} W(\omega) \left[ |H_{SP}(e^{j\omega})| - |H(e^{j\omega}; \mathbf{m})| \right]^p d\omega \tag{9.76a}$$

with  $W(\omega)$  a proper weight function.

If the phase response is relevant, another type of loss function is given by:

$$L'' = \int_{-\pi}^{+\pi} W(\omega) \left[ \tau_{SP}(e^{j\omega}) - \tau(e^{j\omega}) \right]^p d\omega \tag{9.77a}$$

where  $\tau(e^{j\omega})$  is the filter group delay defined as:

$$\tau(e^{j\omega}) = -\frac{d}{d\omega} \{ \angle H(e^{j\omega}) \} \tag{9.78}$$

Usually the loss functions are not defined on the whole  $\omega$  axis, but only at certain frequencies  $\omega_k$ . In this case the previous expressions (9.76a) and (9.77a) are reformulated as:

$$L'(\mathbf{m}) = \sum_k W(\omega_k) \left[ |H_{SP}(e^{j\omega_k})| - |H(e^{j\omega_k}; \mathbf{m})| \right]^p \tag{9.76b}$$

$$L'' = \sum_k W(\omega_k) [\tau_{SP}(e^{j\omega_k}) - \tau(e^{j\omega_k})]^p \tag{9.77b}$$

If  $M$  is the number of the unknown filter parameters in the vector  $\mathbf{m}$ , and there are not any constrained on their values, the unconstrained optimization problem:

$$\tilde{\mathbf{m}} = \min_{\mathbf{m}} L(\mathbf{m}) \tag{9.79}$$

can be solved numerically finding the solution of the following set of  $M$  non-linear equations:

$$\frac{\partial L}{\partial m_i} = 0; \quad 1 \leq i \leq M \tag{9.80}$$

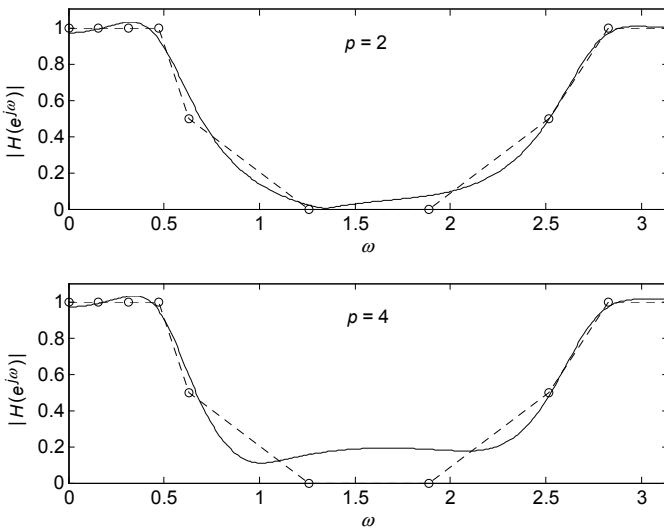


Figure 9.31. IIR filter designed by numerical optimization using as loss function the mean squared error and the mean error of order 4, respectively

If the transfer function  $H(z)$  is given in the factorized form:

$$H(z) = A \prod_{k=1}^N \frac{1 + a_k z^{-1} + b_k z^{-2}}{1 + c_k z^{-1} + d_k z^{-2}} \tag{9.81}$$



the  $M$  parameters  $m_i$  in (9.80) are the  $2N+1$  coefficients  $A, a_k, b_k, c_k$  and  $d_k$ .

Figure 9.31 shows the result obtained for  $N = 2$  with a loss function of type (9.76b) for the two cases with  $p$  equal to 2 and 4. The values of the weight function  $W(\omega)$  were 1 in the pass-band, 0.1 in the stop-band and 0.5 in the transition band.

Another important advantage of the IIR filter design by numerical optimization is that the specified frequency response does not require to be defined at uniformly spaced frequencies, as shown in the example of figure 9.31.

### EFFECTS OF FILTER COEFFICIENTS QUANTIZATION.

As for FIR filters, the limitation in the number of bits available for storing the IIR filters coefficients and the internal data results in a change in its frequency response. Expressing the IIR transfer function as the polynomials ratio (9.3), the quantization effect on the frequency response can be analyzed in the frequency domain in terms of a numerator and denominator error functions:

$$H(z) = \frac{N(z) + \Delta N(z)}{D(z) + \Delta D(z)} \tag{9.82}$$

Since, according to (9.3), the numerator and denominator of  $H(z)$  are given by

$$N(z) = \sum_{k=0}^M \alpha_k z^{-k}, \quad D(z) = \sum_{k=0}^N \beta_k z^{-k} \tag{9.83}$$

If  $\delta\alpha_k$  and  $\delta\beta_k$  are the rounding error in the filter coefficients, the frequency responses of the numerator and denominator error functions are:

$$\Delta N(e^{j\omega}) = \sum_{k=0}^M \delta\alpha_k e^{-jk\omega}, \quad \Delta D(e^{j\omega}) = \sum_{k=0}^N \delta\beta_k e^{-jk\omega} \tag{9.84}$$

If  $\Delta$  represents the quantization step the maximum values of  $\Delta N(e^{j\omega})$  and  $\Delta D(e^{j\omega})$  are given by:

$$\begin{aligned} \max |\Delta N(e^{j\omega})| &= \max \left| \sum_{k=0}^M \frac{\Delta}{2} e^{-jk\omega} \right| \leq (M+1) \frac{\Delta}{2} \\ \max |\Delta D(e^{j\omega})| &= \max \left| \sum_{k=0}^N \frac{\Delta}{2} e^{-jk\omega} \right| \leq (N+1) \frac{\Delta}{2} \end{aligned} \tag{9.85}$$

When  $N$  and  $M$  exceed few units the worst case estimation (9.85) of the coefficients quantization effect is too large and very unlikely. A better estimation can be obtained modeling the quantization error of each filter coefficient as a random noise uniformly distributed between  $-\Delta/2$  and  $\Delta/2$ . The filter coefficients variance is in this case:

$$\sigma^2(\alpha_k) = \sigma^2(\beta_k) = \frac{\Delta^2}{12} \quad (9.86)$$

Considering statistically independent the quantization noise on each filter coefficient, the standard deviation of the (9.84) is given by the uncertainty propagation law applied to (9.83):

$$\begin{aligned} \sigma^2[\Delta N(e^{j\omega})] &= \sum_{k=0}^M \left| \frac{\partial N(z)}{\partial \alpha_k} \right|^2 \sigma^2(\alpha_k) = \sum_{k=0}^M |e^{-jk\omega}|^2 \frac{\Delta^2}{12} \\ \sigma^2[\Delta D(e^{j\omega})] &= \sum_{k=0}^N \left| \frac{\partial D(z)}{\partial \beta_k} \right|^2 \sigma^2(\beta_k) = \sum_{k=0}^N |e^{-jk\omega}|^2 \frac{\Delta^2}{12} \end{aligned} \quad (9.87)$$

Since  $|e^{jk\omega}|=1$  expression (9.87) simplifies to:

$$\begin{aligned} \sigma(\Delta N) &= \sqrt{\frac{M+1}{3}} \frac{\Delta}{2} \\ \sigma(\Delta D) &= \sqrt{\frac{N+1}{3}} \frac{\Delta}{2} \end{aligned} \quad (9.88)$$

Since the filter behavior is determined by the zeros in the stop-band and by the poles in the pass-band, the consequences of rounding the filter coefficients will be considered separately in the two bands.

In the stop-band, neglecting the rounding effect on the poles, according to (9.82) the standard deviation of the frequency response approximation is given by:

$$H(z) = \frac{N(z) + \Delta N(z)}{D(z)} \Rightarrow \sigma[H(e^{j\omega})] = \frac{\sigma[\Delta N(e^{j\omega})]}{|D(e^{j\omega})|} \quad (9.89)$$

Similarly in the pass-band the standard deviation of the approximation obtained with the quantized coefficients is:

$$H(z) = \frac{N(z)}{D(z) + \Delta D(z)} \Rightarrow \sigma[H(e^{j\omega})] = \frac{|N(e^{j\omega})|}{|D(e^{j\omega})|^2} \sigma[\Delta D(e^{j\omega})] = \quad (9.90)$$

If  $\delta_s$  and  $\delta_p$  are the specified ripple in the stop-band and pass-band, and  $\delta_{s0}$  and  $\delta_{p0}$  are the ripple without filter coefficients quantization, the two standard deviations (9.89) and (9.90) are limited by:

$$\begin{aligned} \sigma[H(e^{j\omega})] &\leq \frac{N(z)}{D(z)} \frac{\delta_s - \delta_{s0}}{2} \Rightarrow \frac{\sigma[\Delta N(e^{j\omega})]}{|N(e^{j\omega})|} \leq \frac{\delta_s - \delta_{s0}}{2} \\ \sigma[H(e^{j\omega})] &\leq \frac{N(z)}{D(z)} \frac{\delta_p - \delta_{p0}}{2} \Rightarrow \frac{\sigma[\Delta D(e^{j\omega})]}{|D(e^{j\omega})|} \leq \frac{\delta_p - \delta_{p0}}{2} \end{aligned} \quad (9.91)$$

Generally the second condition in (9.91) is more restrictive since  $|D(e^{j\omega})|$  has very low value in the pass-band. This is especially true for very selective filters.

If  $N_C$  is the number of bit used for representing the filter coefficients (and theirs signs), the quantization step is expressed by:

$$\Delta = \frac{\max|\beta_k|}{2^{N_C-1}} \quad (9.92)$$

A useful formula for estimating the number of bits required for representing the filter coefficients can be obtained from (9.88) and (9.92) together with the inequality (9.91):

$$N_C = 1 + \log_2 \frac{\max|\beta_k|}{\Delta} \geq 1 + \log_2 \frac{\sqrt{N+1} \max|\beta_k|}{(\delta_{SP} - \delta_0) \sqrt{3} \min_{0 \leq \omega \leq \pi} |D|} \quad (9.93)$$

Filter coefficients quantization can be studied also in terms of its effects on the zeros and poles perturbation in the z-plane, as done in the previous chapter for the zeros of FIR filters. Starting from the expression of either the numerator or the denominator of the IIR filter transfer function  $H(z)$ :

$$P(z) = \sum_{k=0}^Q \gamma_k z^{-k} = \gamma_0 \frac{\prod_{k=1}^Q z - z_k}{z^Q} \quad (9.94)$$

with  $Q=M$  and  $\gamma_k = \alpha_k$  if  $P(z)$  represents the numerator  $N(z)$  or  $Q=N$  and  $\gamma_k = \beta_k$  if  $P(z)$  represents the denominator  $D(z)$ .

The  $Q$  roots  $z_i$  of the polynomial (9.94) represent either the  $M$  zeros or the  $N$  poles of the IIR filter. The perturbation  $\Delta z_i$  of their values due to the quantization  $\Delta \gamma_k$  of the coefficients  $\gamma_k$  is given by:

$$\Delta z_i = \sum_{k=0}^Q \frac{\partial z_i}{\partial \gamma_k} \Delta \gamma_k \tag{9.95}$$

The partial derivative  $\partial z_i / \partial \gamma_k$  in (9.95) is the  $i$ -th pole/zero sensitivity to the quantization of the  $k$ -th coefficient. As done for the analysis of FIR filters, each sensitivity can be determined using the derivative chain rule applied to the  $P(z)$  derivative with respect to  $\gamma_k$ :

$$\left[ \frac{\partial P(z)}{\partial \gamma_k} \right]_{z=z_i} = \left[ \frac{\partial P(z)}{\partial z_i} \right]_{z=z_i} \frac{\partial z_i}{\partial \gamma_k} \tag{9.96}$$

The derivatives  $\partial P / \partial \gamma_k$  and  $\partial P / \partial z_i$  in (9.96) derive directly from (9.94):

$$\left[ \frac{\partial P(z)}{\partial \gamma_k} \right]_{z=z_i} = z_i^{-k} \tag{9.97}$$

$$\left[ \frac{\partial P(z)}{\partial z_i} \right]_{z=z_i} = -\gamma_0 \frac{\prod_{\substack{j=1 \\ j \neq i}}^Q (z_i - z_j)}{z_i^Q} \tag{9.98}$$

Replacing the expressions (9.97) and (9.98) in (9.96) and solving for  $\partial z_i / \partial \gamma_k$  the  $i^{\text{th}}$  zero sensitivity to the  $k^{\text{th}}$  filter coefficient quantization is given by the following formula:

$$\frac{\partial z_i}{\partial \gamma_k} = - \frac{z_i^{Q-k}}{\gamma_0 \prod_{\substack{j=1 \\ j \neq i}}^Q (z_i - z_j)} \tag{9.99}$$

Equation (9.99) is equivalent to the formula (8.67) obtained for the sensitivity to coefficients quantization in FIR filters. It shows how the sensitivity of zeros/poles location in the  $z$ -plane to the coefficients quantization could be very high if the filter has zeros or poles  $z_i$  and  $z_j$  very close to each other.

IIR FILTER ARCHITECTURES.

Several architectures are possible for implementing IIR filters. A first solution is given by the direct formulation of the IIR transfer function (9.3) reformulated as the cascade of a FIR filter, with transfer function  $H_1(z)$ , synthesizing the numerator and a purely recursive IIR filter, with transfer function  $H_2(z)$ , synthesizing the denominator:

$$H(z) = H_1(z) \cdot H_2(z) = \sum_{k=0}^M \alpha_k z^{-k} \cdot \frac{1}{1 + \sum_{k=1}^N \beta_k z^{-k}} \tag{9.100}$$

Figure 9.32 shows the resultant filter structures, called *direct I* form.

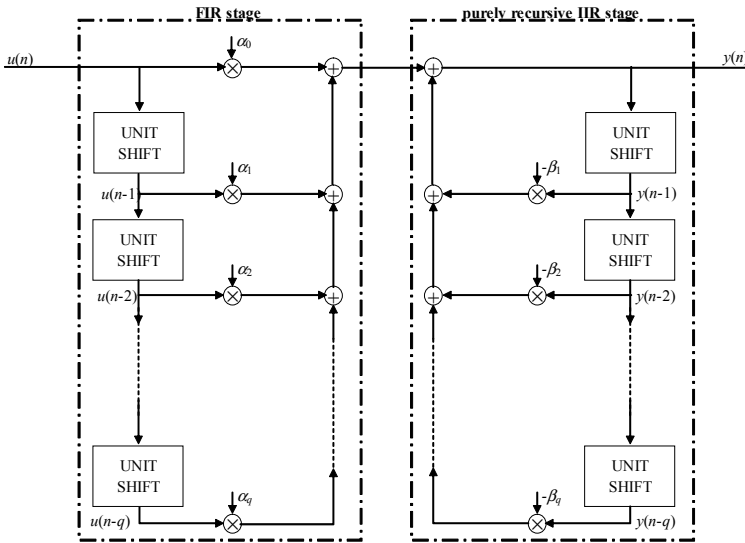


Figure 9.32. Direct I structure of a  $q^{th}$  order IIR filter, with  $q = \max(N, M)$

Another possibility is to invert the position of the IIR and FIR section in the cascade, as shown in figure 9.33. This new architecture is called *direct II* form. It is evident from figure 9.33 that the state variables are stored twice in this numerical network. Figure 9.34 shows the result obtained using the same state variables for both the IIR and FIR sections of the filter. The direct II form has the advantage of requiring the minimum number of memory registers for the state variables.

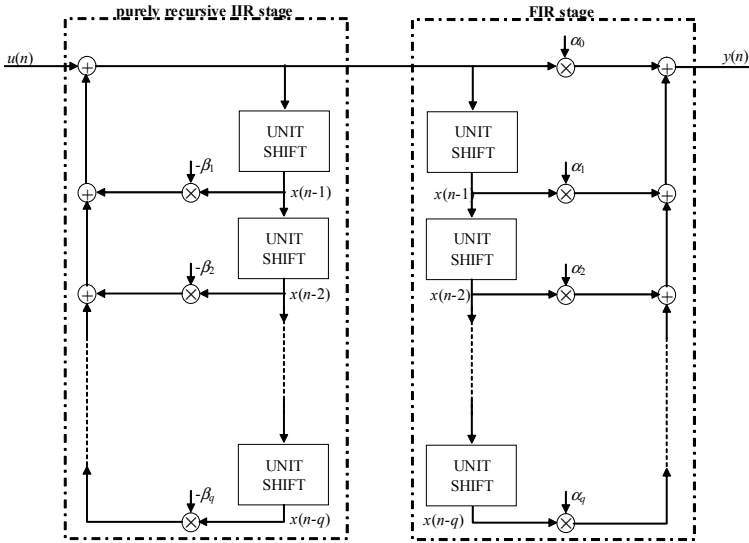


Figure 9.33. Direct II structure

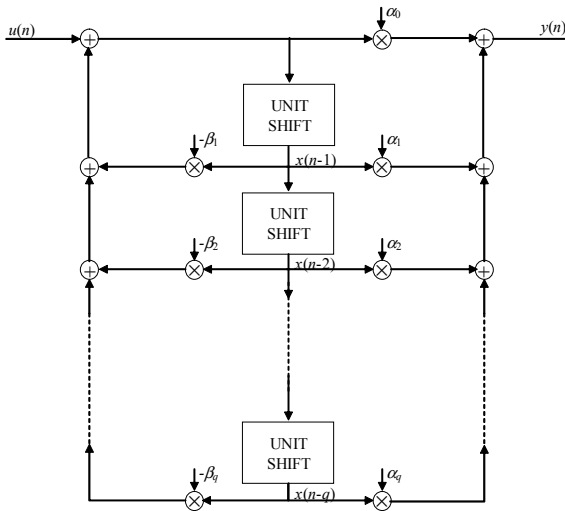


Figure 9.34. Direct II structure using the minimum number of unit shift

For limiting the effects of coefficients quantization, it is better to realize IIR filters as a cascade of 2<sup>nd</sup> order direct II form sections, called biquad. The block diagram of a biquad is shown in figure 9.35.

Other architectures derived from the previous direct I and II forms are the transposed version of these structures, known as the transposed direct I and transposed direct II forms and shown in figures 9.36 and 9.37.

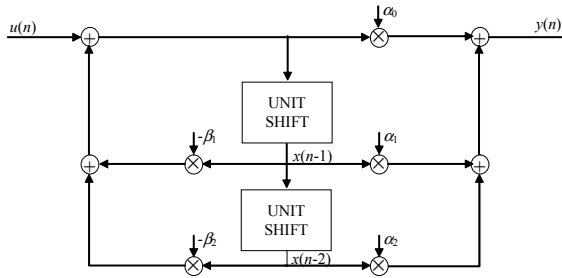


Figure 9.35. 2<sup>nd</sup> order direct II structure (biquad)

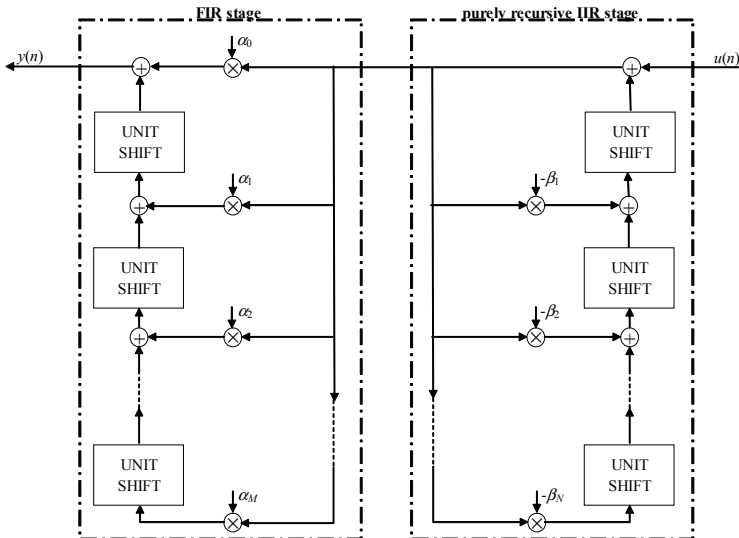


Figure 9.36. Transposed direct I structure

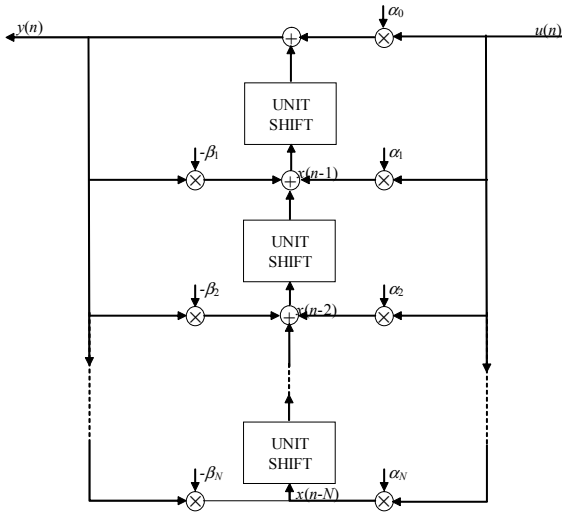


Figure 9.37. Transposed direct II structure



## REFERENCES

Apostol, T.M., (1969). *Calculus*. John Wiley&Sons.

Åström, K.J. and Wittenmarl, B., (1990). *Computer-controlled systems: theory and design*. Prentice-Hall International Ltd.

Bellanger, M., (2000). *Digital processing of signals: theory and practice*. John Wiley&Sons.

Cook, P.A., (1986). *Nonlinear dynamical systems*. Prentice-Hall International Ltd.

Couch, L.II W., (1990). *Digital and analog communication systems*. Macmillan Publishing Company.

Marlow, W.H., (1993). *Mathematics for operations research*. Dover Publications Inc.

Oppenheim, A.V., Shafer, R.W., Buck, J.R., (1999). *Discrete-time signal processing*. Prentice-Hall Ltd.

Ott, E., (1993). *Chaos in Dynamical systems*. Cambridge University Press.

## INDEX

- aliasing 39–40, 42, 48, 81, 110, 115–119, 124, 170, 176, 236, 250
- Analog-to-Digital Converter 83–84
- angular frequency 37, 40, 47–48, 51–54, 76, 148, 153–154, 168, 171–172, 174, 176, 184, 187–188, 228, 230–231, 238–239, 250
- anti-aliasing filter 115–117, 119, 124
- anti-aliasing filter - maximum phase delay
- anti-aliasing filter - minimum gain
- ASIC 108
- backward integration formula
- bandwidth 2, 42, 76, 83, 100, 106, 110–113, 115, 117, 119–120, 124, 234
- bandwidth estimation 110
- Bartlett (triangular) window 196
- BIBO stability 138, 140–141, 164
- bilinear transformation 249–251
- biquad 261
- Blackman-Harris window 69–70
- Butterworth filter 231, 233
- cascaded systems
- causality condition
- central integration formula
- Chebyshev filter 232–233
- circular buffer memory 109
- comb filter 182–183
- conjugate asymmetric sequence
- conjugate symmetric sequence 31–32
- continuous-time domain 3, 7
- continuous-time signal 5, 7, 9–10, 17, 30, 33, 36–43, 45, 47, 51, 53, 58, 124
- conversion time 84–85, 96–98, 100, 104–105, 110, 125
- convolution 14–17, 19–23, 31, 47–49, 51–53, 55, 77–79, 142, 149, 151–152, 163–165, 179, 192–194, 212
- cosine filter 183–185
- cosine windows 60, 63–65, 67, 70, 74, 80
- difference equation 127, 129–130, 166, 179, 183, 185, 213–214, 216, 222, 224–225, 227, 245
- differential non-linearity error 94–96
- direct I structure 259, 261
- direct II structure 260–262
- direct structure 210, 212
- discrete time systems 12, 22, 24–25
- discrete-time domain 3, 7, 58–59
- discrete-time signal 4–7, 9, 24, 37–39, 45, 51, 53, 125, 263
- dither 88–89, 91
- dithering 88, 91
- DPLL structure 81–82
- $\Delta\Sigma$  converters 105
- DSP 1, 3, 33–34, 76, 83, 96–97, 107–108
- DSP-based instrument 3, 83–84, 104, 106–108, 110, 115, 119
- dynamical discrete system 127, 129–130, 134, 163, 165
- effective number of bits 102
- elliptic filter 233–234, 250–251
- equivalent-time sampling 122–124
- Eulero relationship 45
- Euler's integration formula 241–242
- exponential sequence 8–9, 11, 25–26, 150, 165
- FIR filter 105, 179–183, 185–186, 188–192, 194–196, 198–200, 202–216, 218–220, 222, 224, 226, 255, 257–259
- flash converters 105
- flat-top window 70
- forward integration formula 241–242, 244
- Fourier transform 27–32, 34–42, 47–50, 52, 57–60, 62, 64–73, 147–148, 152–154, 172–177, 181, 184–187, 192, 201–202, 235, 237, 242, 246, 248
- FPGA 108
- frequency response 24–29, 31, 116–118, 167–168, 180–185, 187–192, 198, 200–207, 222, 225–229, 231–233, 235–236, 238–239, 241–242, 244, 246–248, 250, 253, 255–256
- frequency-domain interpolation method 76, 80
- full-scale voltage 84–86, 102
- gain error 92–93, 96, 104
- Gauss relationship 45
- Gibb's phenomenon 194–195, 203
- Hamming window 68–70, 196, 198–200
- Hanning window 60, 64, 66, 69, 72–73, 77, 196
- hardware synchronization 80
- \*\*\*\*\*heterodyne concersion 119–122, 124
- ideal characteristic 91–94
- \*\*\*\*\*ideal transition 86–88
- IIR filter 221–227, 229–230, 251, 253–255, 257–259, 261
- impulse invariance 235–239, 250–251
- impulse response 24, 135, 139–141, 163–165, 179–181, 184–187, 189, 192, 195, 198, 201–203, 212–214, 217, 219, 221, 224–229, 237
- integrating converters 104
- interpolation formula for periodic signals 52

- integral non-linearity error 95–96
- inverse z-transform 155–162, 212
- Kaiser window 196
- leakage errors 52, 55, 57, 60, 67, 70, 74, 76, 82
- linear phase FIR filter 191, 209, 211, 216–219
- linear systems 12–14, 16–17
- long range leakage errors 60, 67, 70
- LSB 85–97, 101, 104–105, 118
- main lobe 50–55, 59–60, 62, 64, 66–70, 72, 76–80, 193–198
- maximum asymptotic decay 63–64, 67, 69–71, 76, 80
- memory 2, 83, 97–98, 106, 108–113, 115, 125, 210–214, 216–218, 259
- Minimax 205
- moving average filter 179–180, 182
- MSB 84
- MSE-Minimum Square Error 205
- multi converter structure 124–125
- multi-channel structure 84, 110–111
- multiplexed structure 112–113
- multiplier 81, 120–122, 213
- non-linearity error 94–96, 105, 124–125
- non-multiplexed structure 111, 112
- normalization condition 63, 70
- number of bits 84, 86, 97, 102–103, 105–106, 118, 206, 208, 210, 255, 257
- observation interval 42, 46–51, 57–58, 60, 72, 74, 76, 78, 106, 124
- odd sequence 32
- offset error 91–93
- output transformation 129–132, 162, 166, 210
- parallel systems
- pass band 82, 116, 118–119, 186–187, 194, 231–233, 250, 255–257
- periodic sequence 9–10
- phase space 130
- pole 146, 149–151, 156–157, 159–161, 163–164, 166–169, 180, 182, 184–185, 209, 222, 224–230, 232, 236–238, 256–258
- pole-zero diagram 146, 152–154, 184–186
- product operation between sequences 11, 142
- purely recursive IIR filter 224–227, 229, 259
- quantization 3, 84, 86, 101–102, 105, 206–210, 212, 214–215, 217–218, 255–258, 261
- quantization band 86, 88–90, 94–97, 101
- quantization error 86–89, 91, 101, 207, 256
- raised cosine filter 185
- real sequence 32
- rectangle's integration formula 240, 242–246, 248
- rectangular window 59–60, 62–70, 72, 76, 192, 195–196, 198–200
- resolution 42, 45, 57, 85–87, 96–98, 100–106, 117, 119, 122, 218
- ROC-Region Of Convergence 143
- sample 2-3, 6–8, 11–16, 19–26, 29–31, 33–36, 42–43, 45, 48, 51, 53, 57–60, 72, 74–76, 78, 83–84, 91, 106–107, 109–113, 115, 117, 122–124, 150, 170–171, 174, 176–177, 179, 200–201, 203–205, 234, 237–238
- Sample & Hold 97–100
- sampling distribution 47–49, 52, 57
- sampling frequency 35, 39–40, 42, 53, 76, 79, 81, 115, 124, 174, 203, 205, 220
- sampling theorem 33, 35, 40–43, 46, 48, 51–52, 57, 72, 74, 76, 78, 83, 111, 115, 122
- sequence 2–3, 5–17, 19–20, 22–42, 45, 48, 51, 53, 58, 60, 75, 127–129, 131–132, 134–136, 138–139, 142–143, 145–150, 152–157, 163–165, 167, 170–171, 174–177, 179, 181, 186, 189, 192, 206, 221–223, 229, 237–239
- Shannon interpolation formula 41–42, 52, 201
- shift invariant linear systems 13–14, 16–17
- shift operation 11
- short range leakage errors 55, 70
- side lobes 55, 59–60, 63–64, 66, 68–70, 193–198
- Simpson's integration formula 247–248
- $\sin^6$  window 67
- single-channel structure 110
- sinusoidal sequence 10, 24, 96
- spectral replica
- stability condition
- state 83, 96, 107, 127–141, 162–164, 166–167, 179, 209–211, 213–214, 216, 218, 221, 223, 227, 259
- state equations 129–132, 162, 209
- state motion 130, 134–139
- state space 130–131
- step invariance 238–239
- stop band 116–117, 186, 195, 231–233, 255–257
- successive approximation converters 104–105
- sum operation between sequences
- synchronous sampling condition 45–46, 50, 52–55, 63, 72, 76, 78–80
- time-domain interpolation method 74, 76
- trajectory 130–132, 137–141
- transfer function 162–166, 168, 180–183, 185–186, 193, 204–209, 212–213, 215–216, 218, 221–222, 224–225, 228–229, 231, 235–240, 243, 250–251, 253–255, 257, 259

- transition 88, 116–117, 186, 194–195, 198, 203–205, 233–234, 255
- transition band 116–117, 186, 195, 198, 203–204, 234, 255
- transposed direct I structure 261
- transposed direct II structure 262
- transposed structure 210
- Tustin's transformation 249
- unit sample response
- unit sample sequence 7–8, 11–16, 22–26, 29–31, 58
- unit step sequence 7–8, 24
- zero 8, 12, 19, 23–24, 44–45, 50–51, 60, 63, 72, 77–78, 84, 87–88, 91, 95–96, 100–101, 105, 116, 119, 134–137, 140–141, 146, 149–151, 154, 157, 162–164, 166–169, 180–85, 89, 208–209, 212–219, 222, 224–225, 228, 231, 256–258
- ZIM-Zero Input Motion 134
- ZIR Zero Input Response 135
- ZSM-Zero State Motion 134
- ZSR-Zero State Response 135
- z-transform 127, 141–153, 155–164, 169–171, 174, 180, 192, 205–206, 212, 237, 239–240, 245–246, 248