Multimedia

Gregor Rozinaj, Renata Rybárová, Ivan Minárik, Juraj Kačur

Authors: Gregor Rozinaj, Renata Rybárová, Ivan Minárik, Juraj Kačur

Title: Multimedia

Published by: Czech Technical University in Prague

Compiled by: Faculty of Electrical Engineering

Contact address: Technicka 2, Prague 6, Czech Republic

Phone Number: +420 2 2435 2084

Print: (only electronic form)

Number of pages: 131

Edition: 1st Edition

ISBN 978-80-01-05311-9

Reviewed by: Jozef Čepko

Innovative Methodology for Promising VET Areas
http://improvet.cvut.cz

Lifelong
Learning
Programme

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Explanatory notes

📚	Definition
💡	Interesting
✎	Note
⚙	Example
🧩	Summary
+	Advantage
–	Disadvantage

ANNOTATION

The module contains information about multimedia. At first basic knowledge concepts are introduced: analog and digital signals, digitization process and time and frequency representation. The other part describes signal processing, analog and digital technologies, filters and structure of a communication channel. Further there are described compression techniques of audio and video signals and multimedia technologies as speech synthesis, speech recognition and image recognition.

OBJECTIVES

The main goal of the module is to introduce a student with the fundamental of signal processing, specifically multimedia processing. The student is clearly acquainted with the base principles of Fourier transformation, digital filters, linear systems, compression techniques and practical use of all knowledge in modern science.

LITERATURE

[1] DÚHA, J., GALAJDA, P., KOTULIAK, I., LEVICKÝ, D., MARCHEVSKÝ, S., MIKÓCZY, E.,PODHRADSKÝ, P. a kol. *Multimedia ICT technologies, network platforms and multimedia services*, Vydavateľstvo STU Bratislava, 2005, ISBN 80-227-2310-X.

[2] GAMEC, J. *Spracovanie multimdiálnych signálov*, publikované v rámci projektu ESF NGN, kód projektu v ITMS: 13120110126, Bratislava, 2007.

[3] TALAFOVÁ, R. – ROZINAJ, G.– CEPKO, J.– VRABEC, J.Multimedia *SMS Reading in Mobile Phone* In: INTERNATIONAL JOURNAL of MATHEMATICS AND COMPUTERS IN SIMULATION, Issue 1, Volume 1, ISSN 1998-0159, 2007.

[4] KOTULIAKOVÁ,J. – ROZINAJ, G. *Číslicové spracovanie signálov I*, FABER Bratislava, 1996.

[5] KOTULIAKOVÁ,J. – ROZINAJ, G. – POLEC, J. – PODHRADSKÝ, P. a kolektív. *Číslicové spracovanie signálov II*, FABER Bratislava, 1997

[6] HARDESTY, L. *Explained: The Discrete Fourier Transform,* 2009, http://web.mit.edu/newsoffice/2009/explained-fourier.html [online]

[7] MINÁRIK, I. *Coding of audio signals at low speed*, Diploma Thesis, 2011

[8] MARMOL, F.G., et al. *ANALYSIS: State of The Art on Identity, Security and Trust*, Deliverable D3.1, HBB-Next FP7-ICT-2011-7, 2012

[9] DEVENTER, O. et al. *ANALYSIS: Multi-User, Multimodal & Context Aware Value Added Services*, Deliverable D5.1, HBB-Next FP7-ICT-2011-7,2012

[10] LEVICKÝ, D., RIDZOŇ, R. *Multimédiá a multimediálne technológie*, publikované v rámci projektu ESF NGN, kód projektu v ITMS: 13120110126, Vydalo Vydavateľstvo STU v Bratislave v spolupráci s AGROGENOFOND Nitra, 2007, ISBN 978-80-227-2604-7

[11] PODHRADSKÝ, P. *Fourierov rad a Fourierova transformácia*, publikované ako študijný materiál pre predmet inžinierskeho štúdia "Analógové a digitálne signály a sústavy I", študijný program Telekomunikácie, FEI STU Bratislava, 2003

Index

In the context of this book the term multimedia means an integration of text, picture and sound with purpose of conveying information. Each of the mentioned parts of information, or modalities, can be represented by several ways. Standard text, hypertext, tables or web pages can all be considered as text. To the pictures group belong static picture, dynamic picture, graphics, animation, video, etc. Specific parts of sound are music, speech, tones, sound signals, etc.

Multimedia is perceived by our senses. In addition to vision and hearing, which help us to “see and hear” the mentioned modalities, we sense the world also by other three senses: smell, taste and touch. Therefore we generally include signals that affect all five senses into the concept of multimedia. Multimedia communication often surrounds communication, which needs at least two senses; even though there are multimedia applications using just one sense.

By definition, multimedia application usually has a possibility of human interaction; it means it can be controlled by man. From this point of view classic TV does not belong to multimedia because it is not manageable.

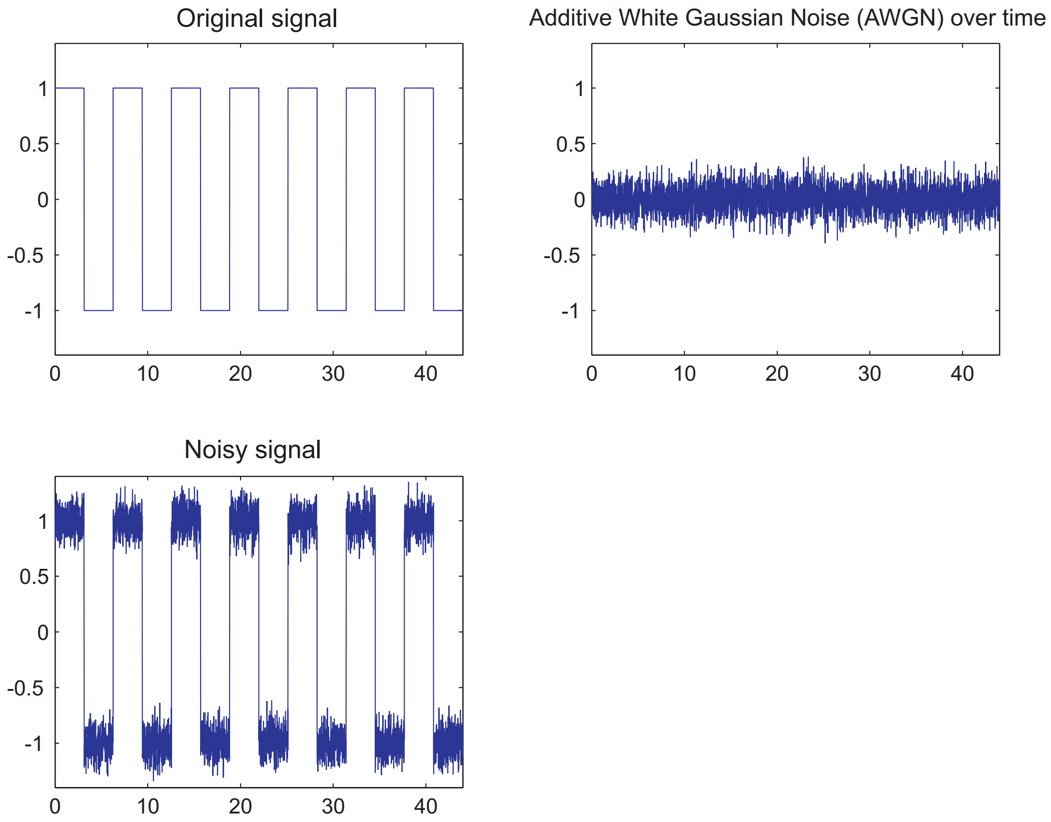
A medium is defined as a way or a resource, which is used to present, sense, save or transfer information. Multimedium is a medium based on several modalities. Carrier of the media information is the signal.

1 Signals

1.1 What is a signal?

Definition of a signal, as used in this material, is a function that comprehensively describes information about the behavior of a phenomenon.

In the physical world, any quantity exhibiting variation in time (such as voice) or variation in space (such as an image) is potentially a signal that might provide information on the status of a physical system, or convey a message between observers, among other possibilities. Real signal is always interferenced with noise.



Real signal – signal+noise

In the area of signal processing and electrical engineering, two types of signal are distinguished – an analog and a digital signal.

Analog Signal

An analog signal is any continuous signal and as such can have any value.

Analog signal has infinitely many values in time and in amplitude. It represents how characteristic feature or phenomenon is varying in time.

Typical analog signal is electrical signal or day temperature varying in time. For some experiments or analysis processes deterministic (can be described by mathematical formulas) or stochastic (behavior is sporadic, random and cannot be predicted) signals are used.

Digital Signal

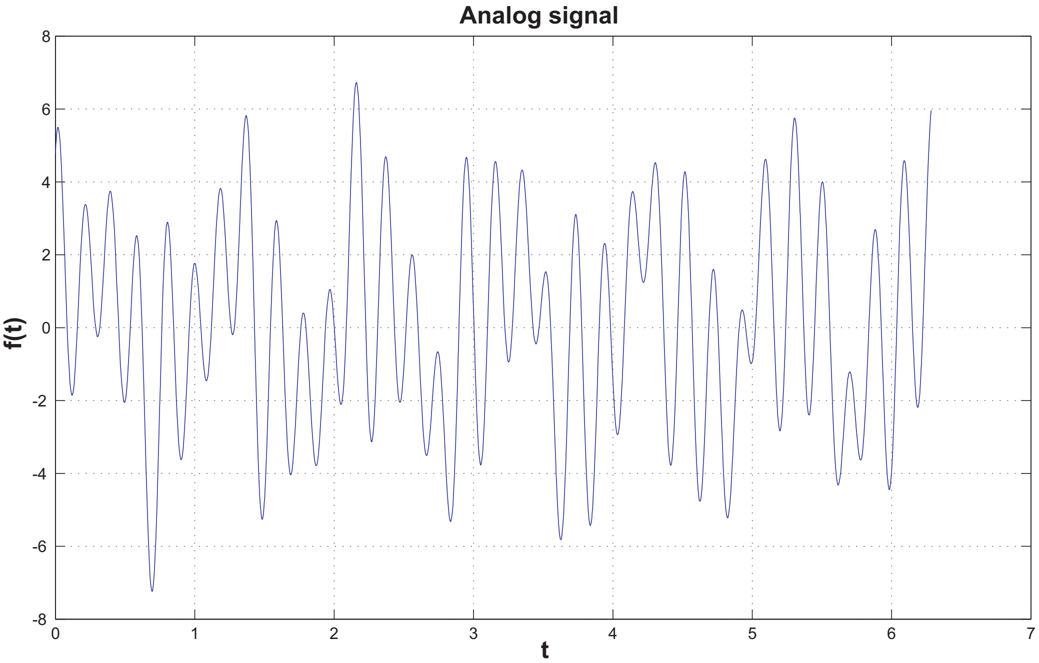
A digital signal is represented by a sequence of discrete (usually predefined) values.

Digital signal has finite number of samples in a particular time points. Simple example how to get discrete signal is sampling of continuous (or analog) signal. An example of digital signal can be air temperature measured only each five minutes, or ones and zeroes used in computing.

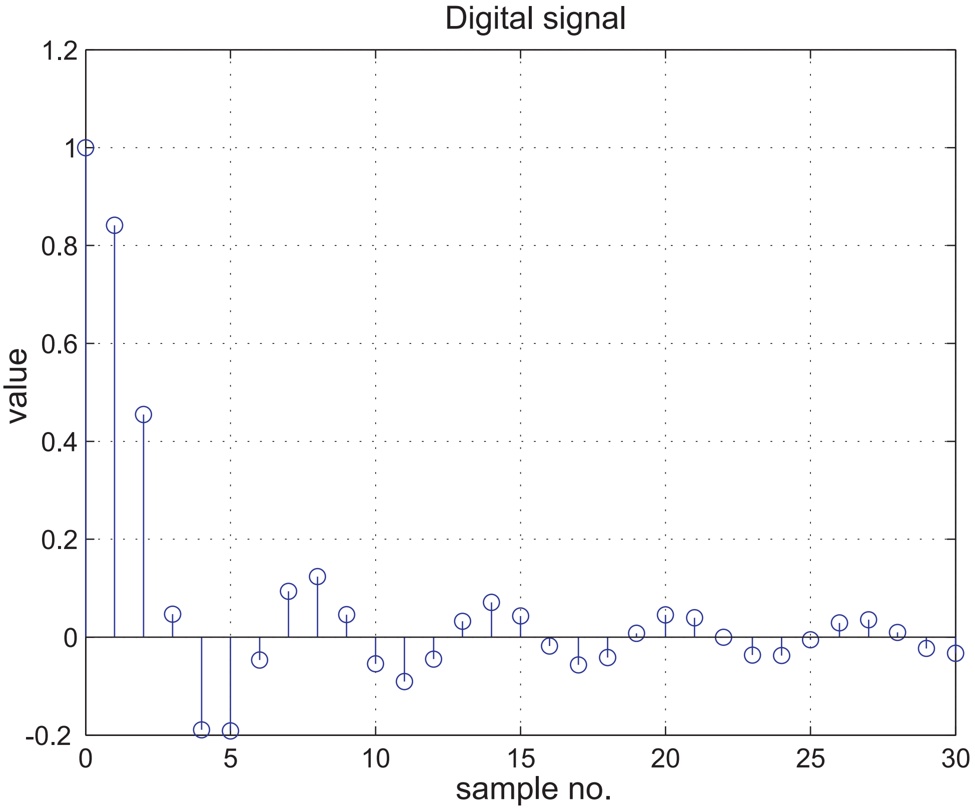
All processes in the nature are analog by design (think of a chart showing temperature over time, or speed of a car over time).

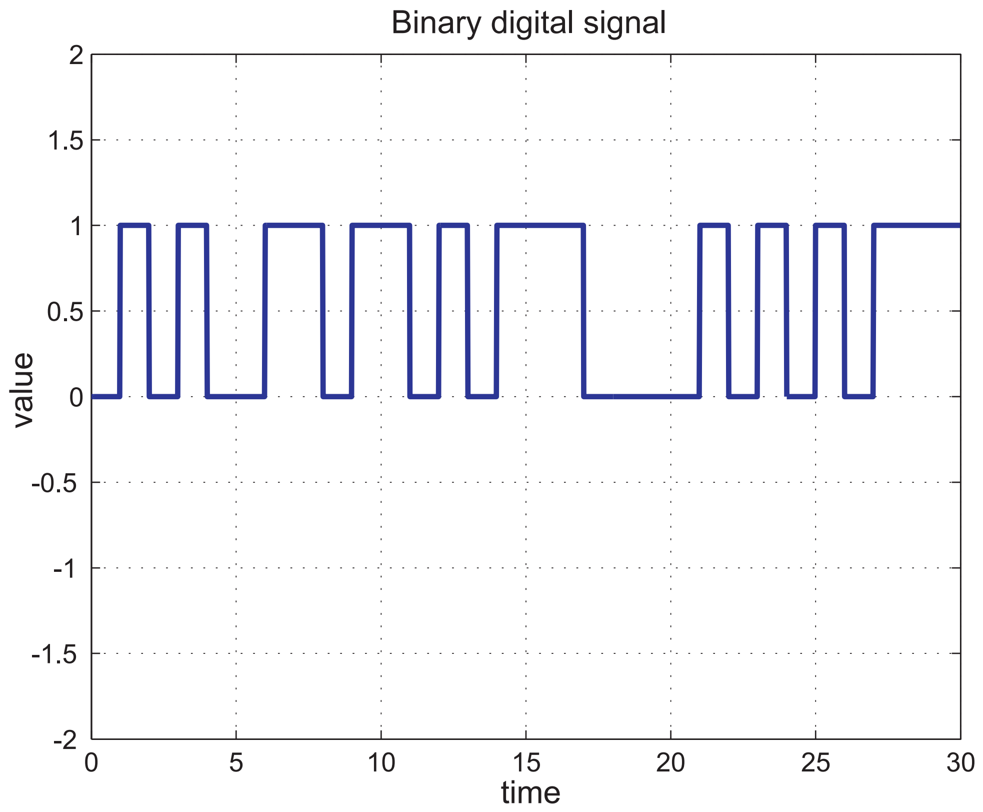
+ So the main advantage of processing of an analog signal is that we do not lose any information. However digital values are much easier to process and work with (think of a music CD – how easily it can be converted to a MP3, digital signal is less vulnerable to noise).

− Analog signal is more difficult to process and work with (think of a vinyl gramophone record – good quality, but not as convenient to work with as a CD). On the other hand with digital signal we lose certain amount of information via process called sampling and quantization (think of a table showing temperature in each hour of the day – 24 values).



Example of analog and digital signal




Binary digital signal
value
time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
20
25
30

1.2 Important signals

In previous chapter was explained the meaning of the word signal. It was not mentioned if the signal is one or more dimensional. In this chapter the most important signal in digital signal processing and multimedia will be introduced.

One-dimensional signals

A signal which is a function of single independent variable is called one-dimensional signal. Usually, the only independent variable is time *t* (for example *f(t)*=5*t*), in case of the discrete signal the independent variable is number *n* (for example *f(n)* = *n*+1).

In all following definitions and formulas *x* refers to a set of real numbers $\{R\}$ and *n* to a set of natural numbers $\{N\}$.

Dirac delta function or δ function is generalized function on the real number line that is zero everywhere except at zero. The delta function is sometimes thought of as an infinitely high, infinitely thin spike at the origin, with total area one under the spike. In the area of signal processing it is often referred to as the unit impulse symbol.

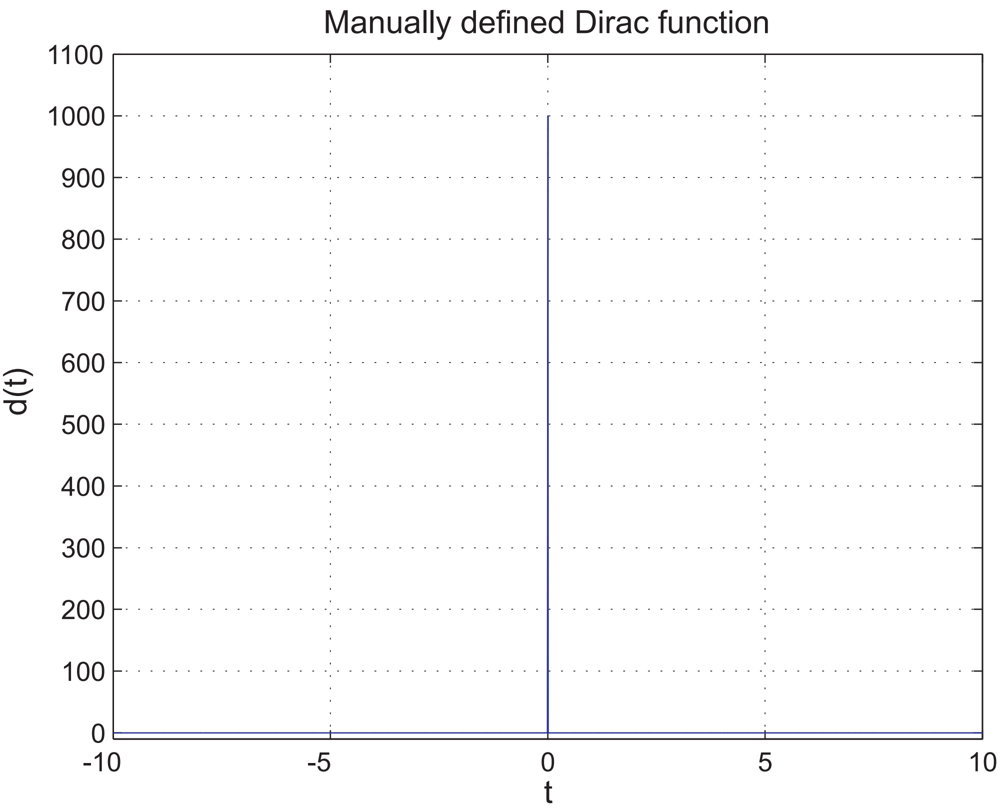
Mathematical definition:

$$\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases}$$

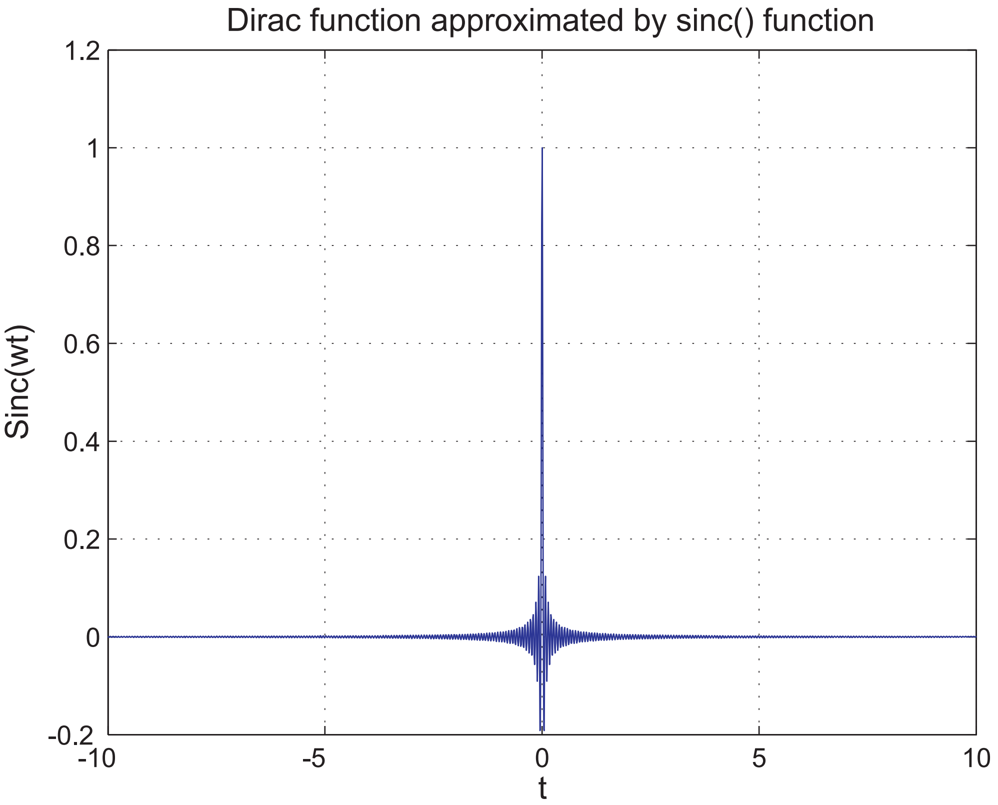
and which is also constrained to satisfy the identity

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

On the picture below is displayd ideal and approximated Dirac delta function. The appoximated Dirac is just for better explanation, how we can get in real word Dirac delta function.



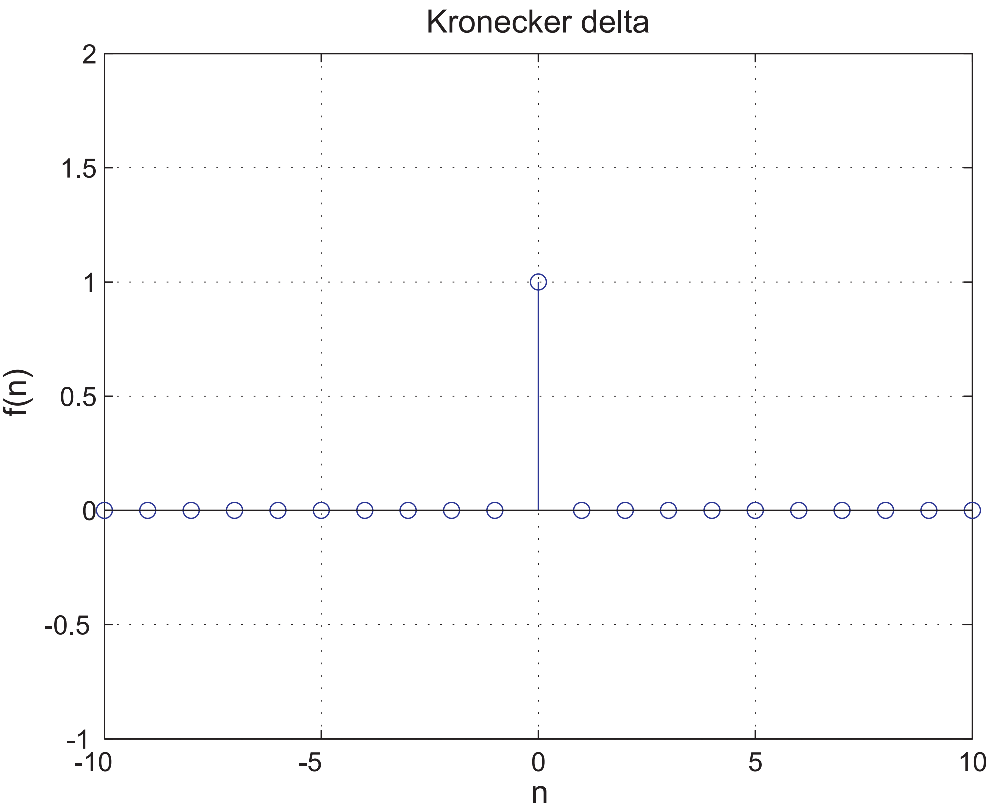
Dirac delta function – ideal and approximated by *sinc()* function



In the discrete domain the equivalent of Dirac delta function is Kronecker delta function. In the area of digital signal processing, the function is referred to as an impulse, or unit impulse. And when it stimulates a signal processing element, the output is called the impulse response of the element.

Mathematical definition:

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



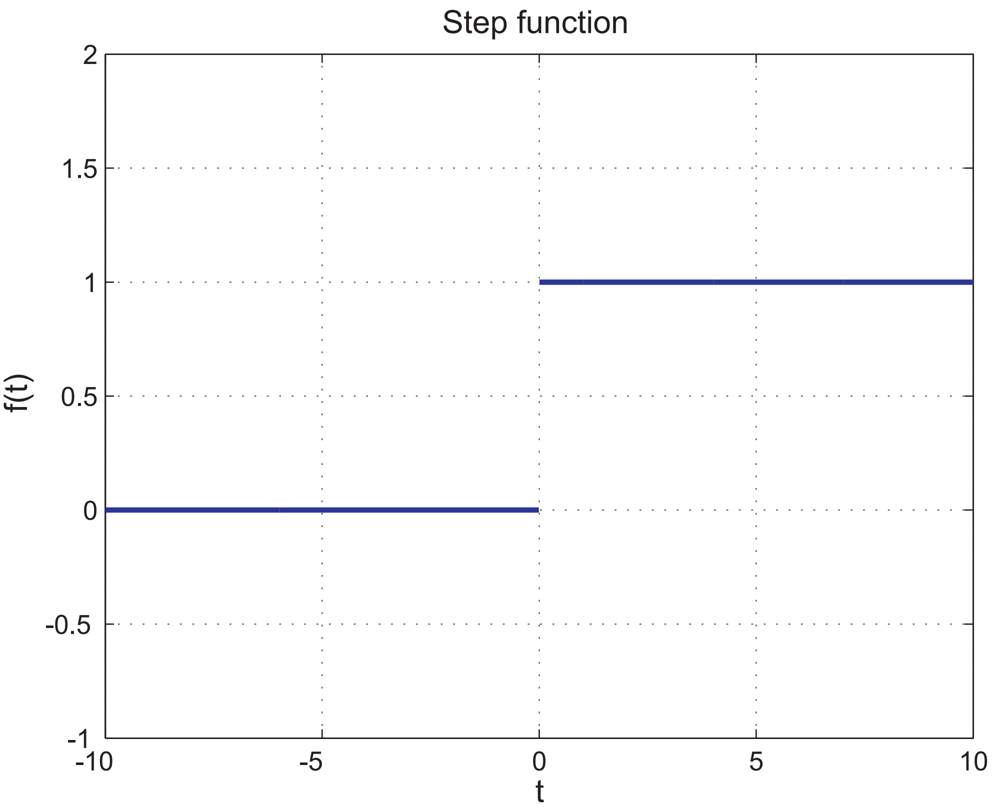
Kronecker delta function

Unit step function, usually denoted as *u*, is a discontinuous function whose values is zero for negative argument (negative number) and one for positive argument. The function is used in signal processing to represent a signal that switches on at a specified time and stays switched on indefinitely. The unit step function is the integral of the Dirac delta function.

$$u(x)=\int_{-\infty}^{x}\delta(s)\,ds$$

Mathematical definition:

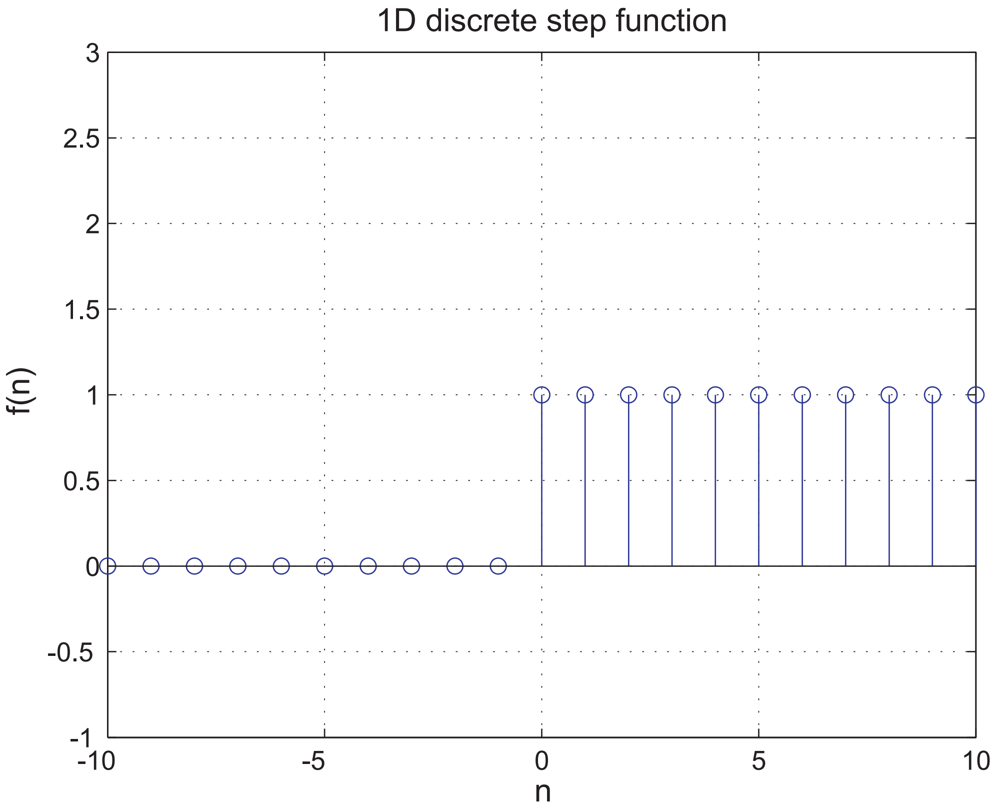
$$u(x)=\begin{cases}0 & x<0\\ 1 & x\geq 0\end{cases}$$



Unit step function (continuous)

Discrete form of unit step function:

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



Unit step function (discrete)

Other special group of signals is periodic signals. Periodic function (which describe periodic signal) is a function that repeats its values in regular intervals or periods. Here belong for example all trigonometric functions (sine, cosine, tangent, cotangent with period 2π). If the period is *P*, then mathematical definition of a periodic function is:

$$f(x) = f(x+P)$$

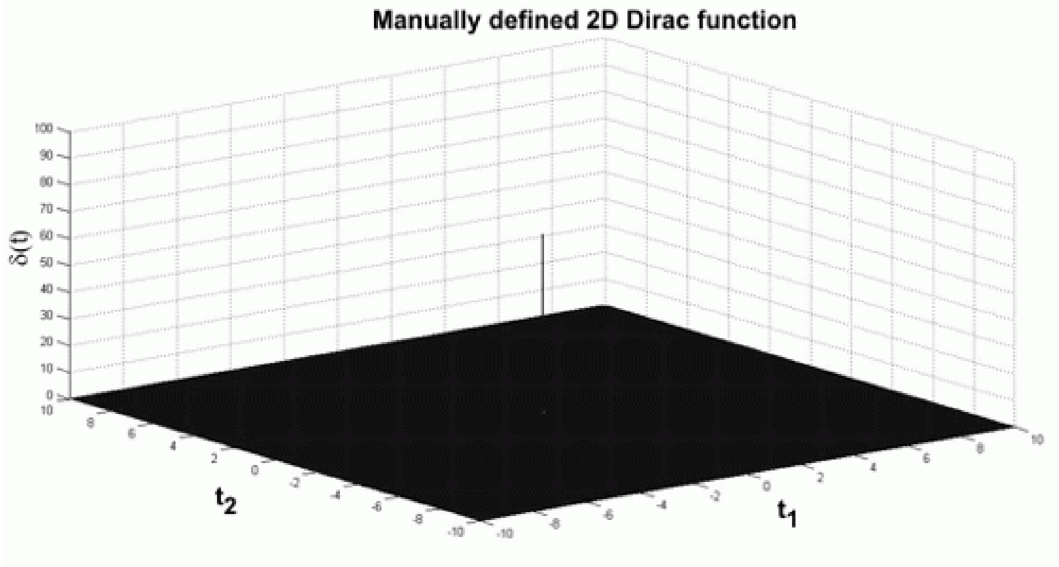
Two-dimensional signals

A signal which is a function of two independent variables is called two-dimensional signal. A typical example of two – dimensional signal is a picture. Picture consists of a brightness and luminescence signal. A 2D image can have a continuous spatial domain, as in a traditional photograph or painting; or the image can be discretized in space, as in a raster scanned digital image.

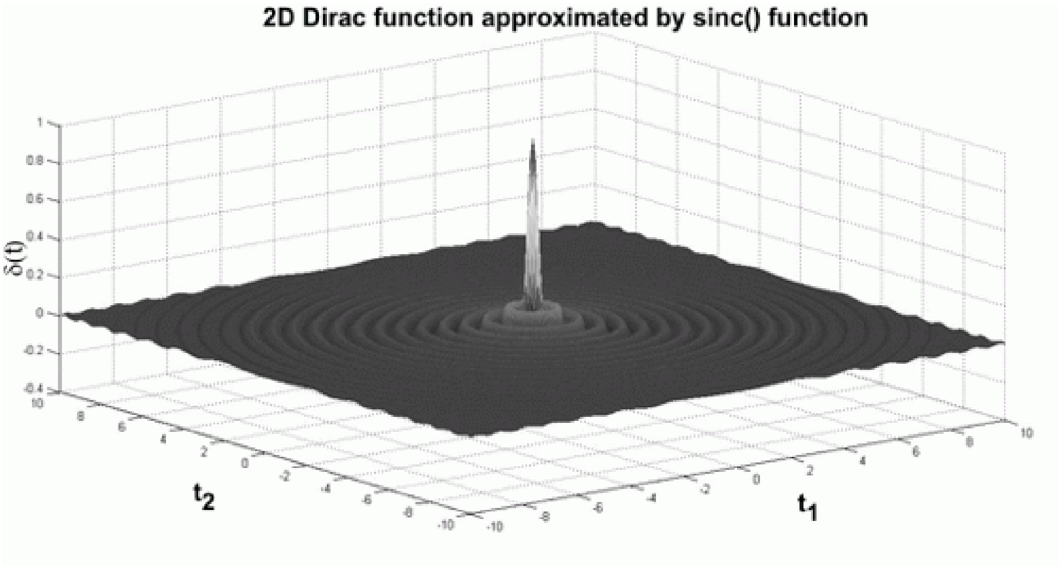
All important signals listed for one-dimensional domain are defined also in two-dimensional domain. Only mathematical definitions will be listed.

Dirac delta function

$$\delta\left(x_{1,}x_2\right)=\begin{cases}+\infty & x_{1,}x_2=0\\ 0 & x_{1,}x_2\neq 0\end{cases}$$

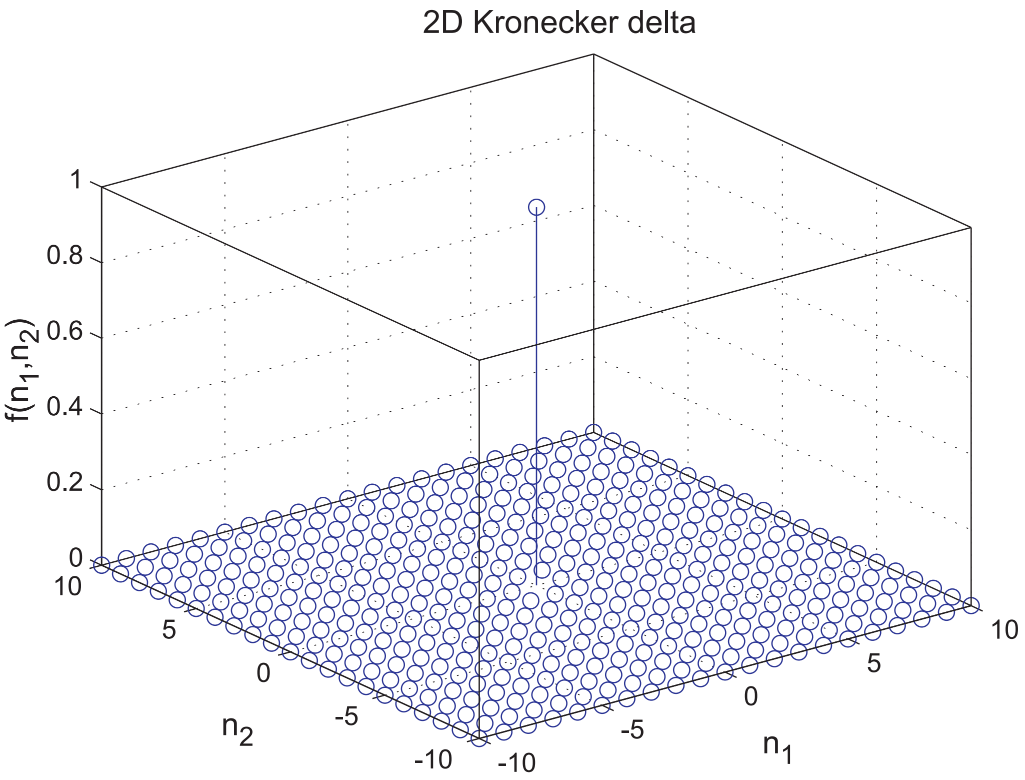


2D Dirac delta function – ideal and approximated by *sinc()* function



Kronecker delta

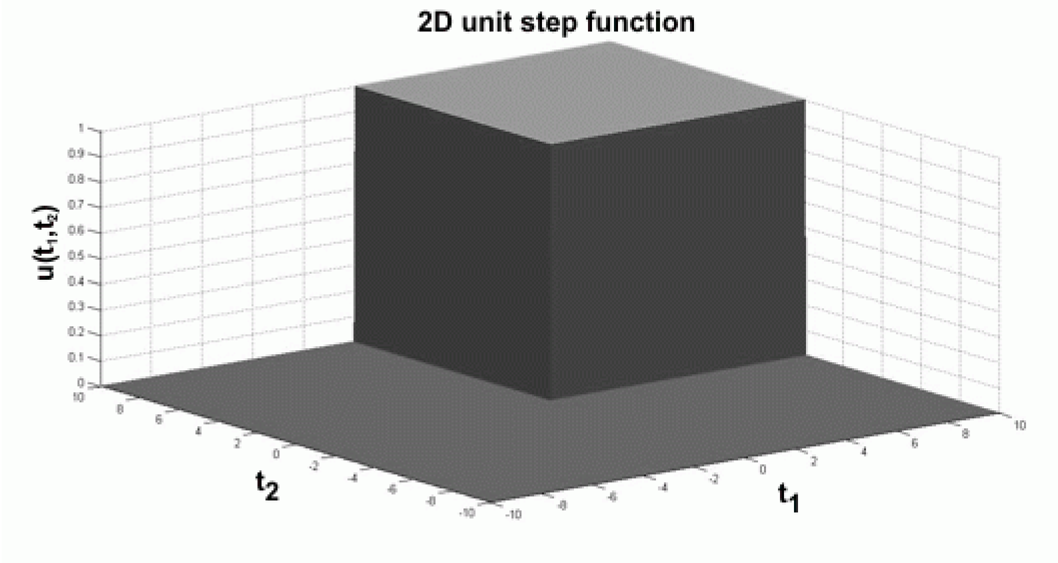
$$\delta\left(n_{1,}n_{2,}\right)=\begin{cases}1 & n_{1,}n_{2,}=0\\ 0 & n_{1,}n_{2,}\neq 0\end{cases}$$



2D Kronecker delta

2D Unit step function (continuous)

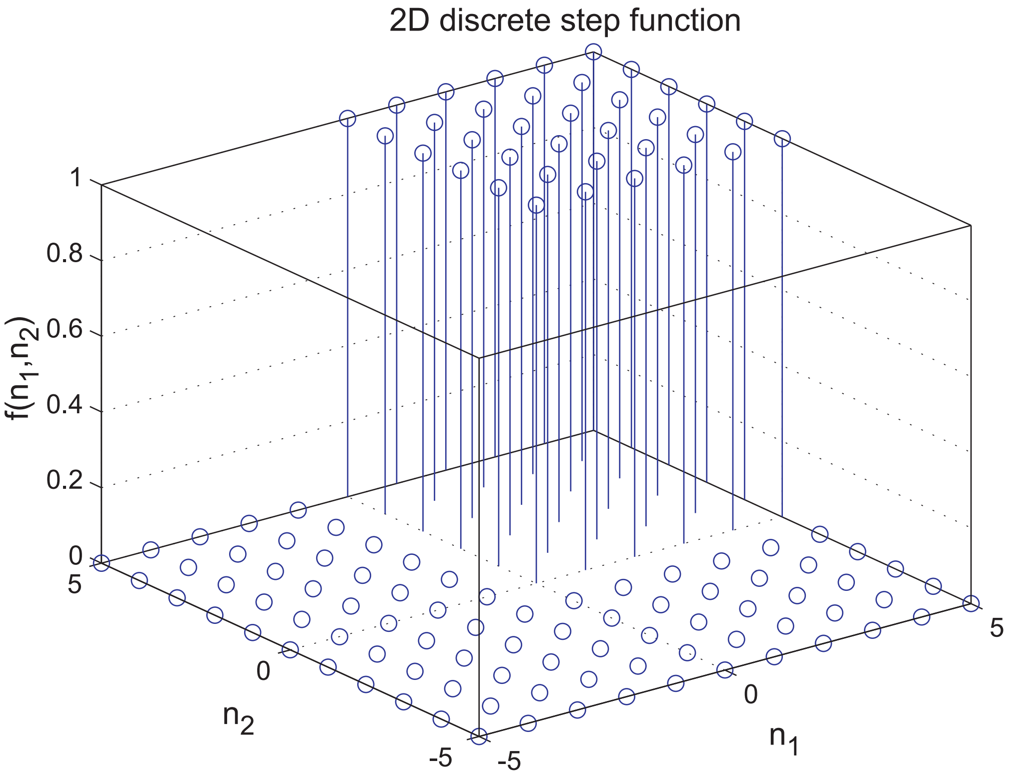
$$u\left(x_{1,}x_{2,}\right)=\begin{cases}0 & x_{1,}x_2<0\\ 1 & x_{1,}x_2\geq 0\end{cases}$$



2D Unit step function (continuous)

Unit step function (discrete)

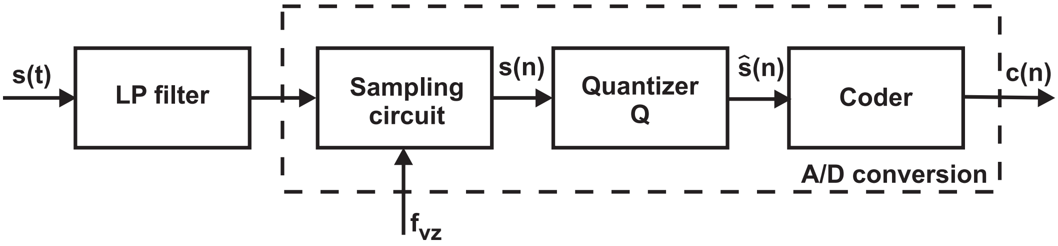
$$u\left(n_{1,}n_{2,}\right)=\begin{cases}0 & n_{1,}n_2<0\\ 1 & n_{1,}n_{2,}\geq 0\end{cases}$$



2D Unit step function (discrete)

1.3 Analog signal digitalization

Analog signal digitalisation is a procedure in which the analog signal (usually representing the specific media) is transformed into digital form. The signal is sampled, quantized and coded. The result is then a sequence of binary digits which is further processed.



Analog signal conversion to digital form (PCM method)

The main methods of multimedia signal coding in the time domain used in multimedia telecommunications are as follows:

- Pulse Code Modulation (PCM),
- Differential PCM (DPCM),
- Adaptive DPCM (ADPCM).

The signal is after low pass filtering (antialiasing filter) sampled in a sampling circuit and a sequence of samples is obtained.

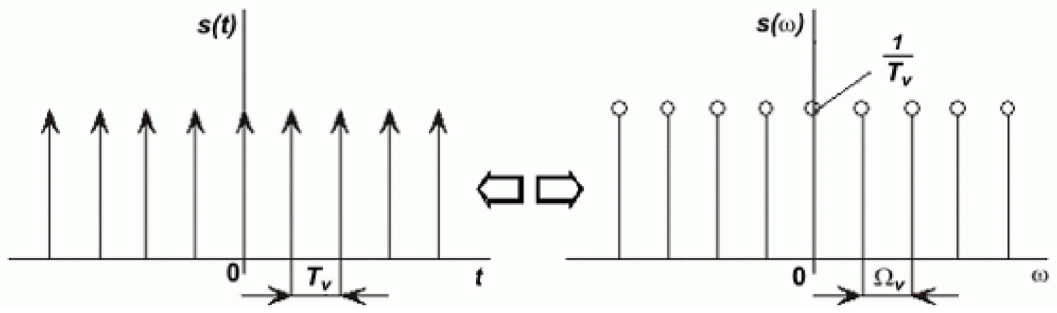
Sampling is the reduction of a continuous signal to a discrete signal. Samples are taken in defined time periods; a sample refers to a value or set of values at a point in time and/or space. The value of sampling rate is given by the sampling theorem (known as Shannon-Kotelnik theorem), e.g. the sampling rate must be at least two times of the highest frequency in the sampled multimedia signal.

The value of sampling rate F_s is given by the bandwidth of the sampled signal. For example a telephone speech has a frequency bandwidth 300 – 3400 Hz (4000 Hz), therefore the sampling rate is F_s = 8 kHz.

Sampling theorem can be mathematically defined as $T_s \leq \dfrac{1}{2F_m}$.

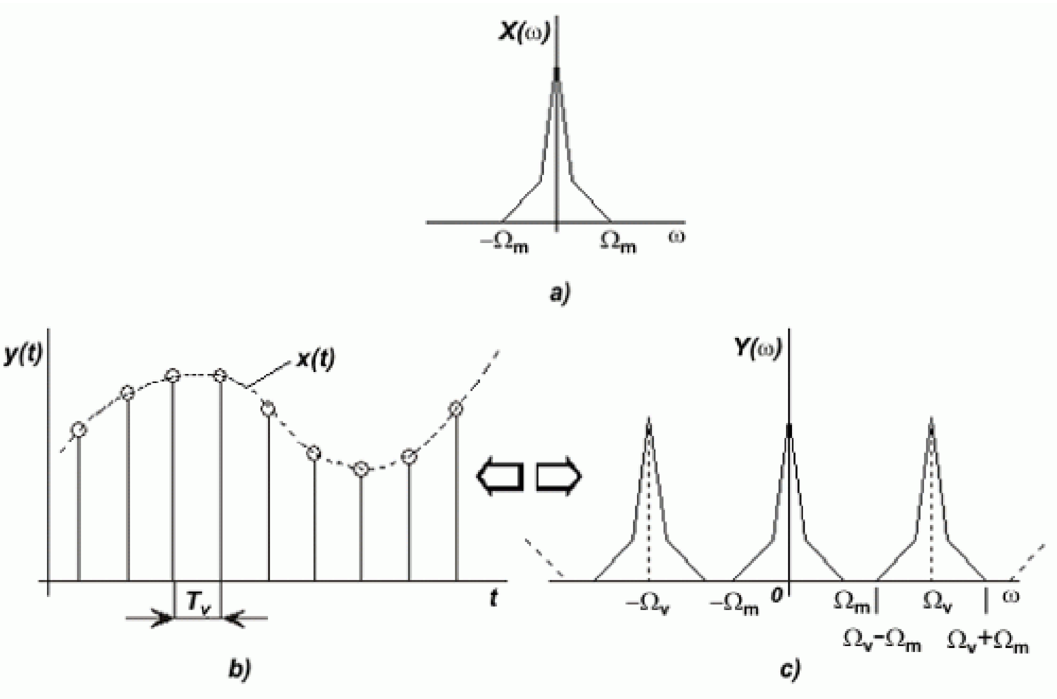
The minimum sampling frequency $F_{s\min} = 2F_m$ is called the Nyquist frequency.

Sampled signal, let's mark it as *y(t)*, can be defined as a product of the original signal *x(t)* and sampling function *s(t)* represented as an infinite series of the Dirac impulses. Distance of impulses in the time domain is $t = T_s$. In the frequency domain spectrum of signal (which is periodic) is discrete with distance of frequency components Ω_s .



Sampling function in time and frequency

Spectrum *Y*(ω) of a sampled signal *y(t)* is a result of the convolution of the spectrum *X*(ω) and *S*(ω).



a)spectrum of original signal *X*(ω), b)sampled signal *y(t)*in time domain , c) spectrum of sampled signal *Y(ω)*

Sampling signal with smaller frequency than is signal's maximum frequency ($\Omega_s < 2\Omega_m$) causes overlapping of spectral components. This is called aliasing. Aliasing can happen also in case when signal has unlimited spectrum. Result is that the signal reconstructed from samples is different from the original continuous signal.

Each sample of the signal is substituted by corresponding quantization level (fixed set of numbers such as integers, or natural numbers), which results in a sequence of the quantized samples, a process known as quantization . Quantization levels are obtained by dividing of the amplitude into small intervals. Interval length is called quantization step. In case the steps are equal the quantization process is linear, in other case the process is non linear.

The main disadvantage of this process is quantization error or noise. It is the difference between the analog input to the **ADC** **(***analog-digital convertor***)** and the output digitized value. The noise is non-linear and signal-dependent. This error causes problems during conversion of digital signal back to analog. The signal is never converted back to the original form; it can be only approximated from the quantization values.

The next step in digitalization process is coding.

The coding of quantized samples is performed by assigning a binary code word to each quantized sample. In this way a sequence of the code words is obtained.

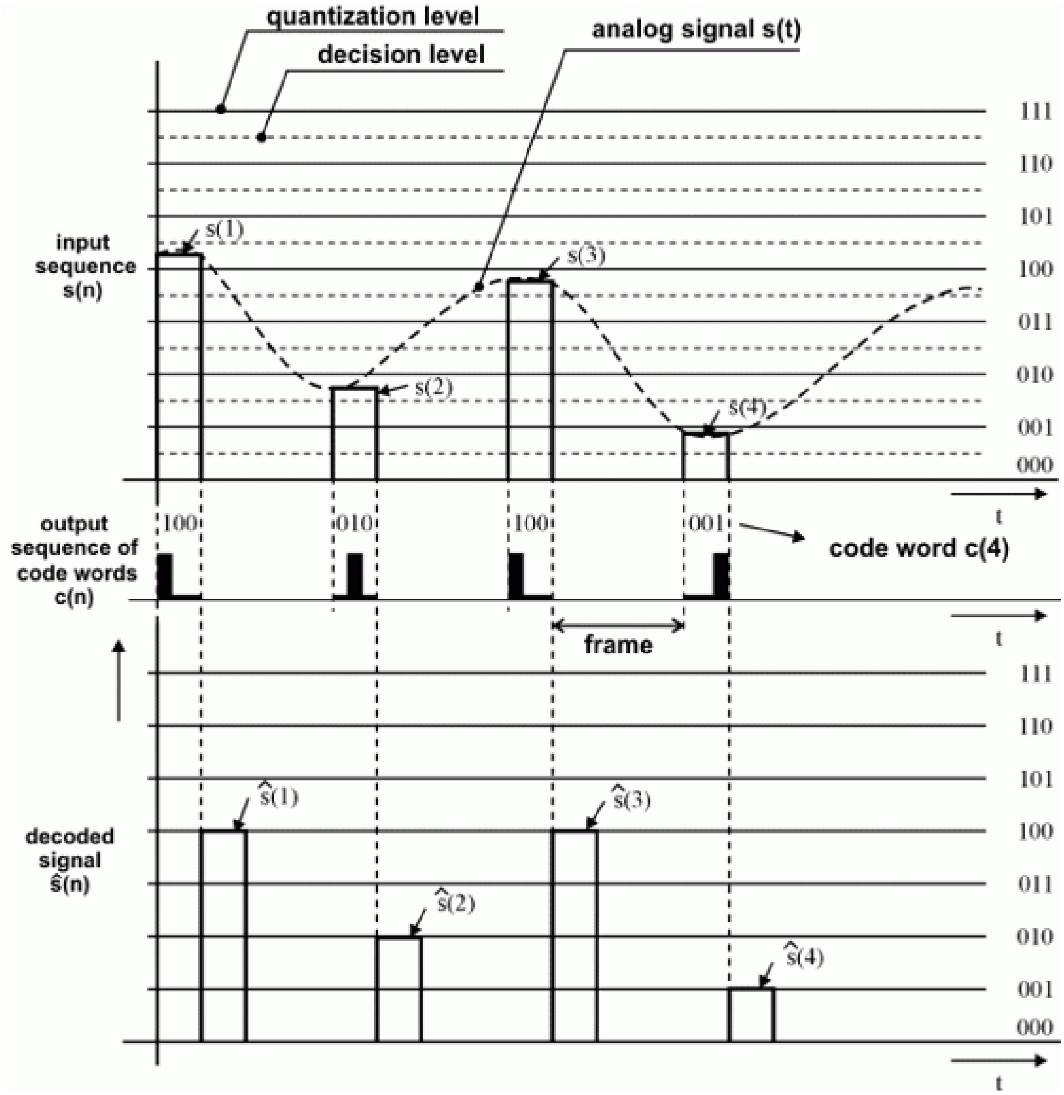
PCM method is an international standard for multimedia signals coding and transmission. The principle of this method is depicted on picture below.

The first systems based on PCM have used 7 bits code words N, e.g. the number of quantization levels has been 128. If we consider a sampling rate F_s = 8 kHz and N = 8, then the required bit rate for speech transmission in telephone bandwidth is $8\cdot10^3\cdot8 = 64$ kb/s.

The advantage of PCM coding in comparison to analog methods of transmission is the resistance of transmitted signal against distortion.

On the other side the disadvantage of this method is the broader frequency bandwidth that is required for signal transmission.

The waveforms of signals coded by using PCM are shown in picture below.



Waveforms in PCM coding of multimedia signals

Linear PCM uses the same constant quantization step in the whole range of the quantization. Therefore the range (dynamic) of quantized signal depends upon number and size of quantization steps. The number of quantization levels for a particular signal determines the size of quantization error. Higher number of quantization levels provides smaller quantization error, but the requirements on the transfer rate are higher. These disadvantages of the method can be solved by non-linear arrangement of quantization levels, which is the idea of non-linear PCM.

Non-linear PCM uses non-linear arrangement of quantization levels. The size of quantization steps for higher signal amplitudes gets larger. A modification of this method uses compression of dynamics of input signal at the transmitter side and expansion of dynamics at the receiver side. In this way small samples are amplified and large one are reduced by a compressor. At the receiver side the expander returns the samples in their original range.

Decoding process is the same process as coding but in reverse order. The output of the decoder is the sequence of quantized samples.

1.4 Modulation

An analog signal, as was defined in the first chapter, is a variable signal continuous in both time and amplitude. If we were to graphically represent alternating current, it would appear as a wave, with voltage bouncing above and below the zero level. There are three factors to consider: frequency, amplitude and phase.

Frequency is the rate at which the current alternates above and below the zero current level.

When the current rises above zero, dips below zero and then returns to zero, we say the current has completed one "cycle". The name applied to the number of cycles per second is Hertz (Hz). Therefore, if there are 500 cycles per second for an analog signal, we say the frequency is 500 Hertz (500 Hz).

Amplitude would be viewed as the height (peak) and the depth (trough) of the graphic wave.

As analog data travels over distance, the amplitude of the wave decreases. This characteristic is called "attenuation". Analog waves are less susceptible to attenuation problems, but occasionally they have to be amplified. The amplitude of analog waves is measured in watts, amps or volts. The measurement decibel is often used to describe the power of a signal. A decibel (dB) allows us to understand the ratio of two different power levels of a signal. Decibel is logarithmic and dimensionless unit.

Finally, phase describes the difference in the start of the cycle of one signal to the start of the cycle of another. One signal acts as a reference signal; the other signal is the phased signal.

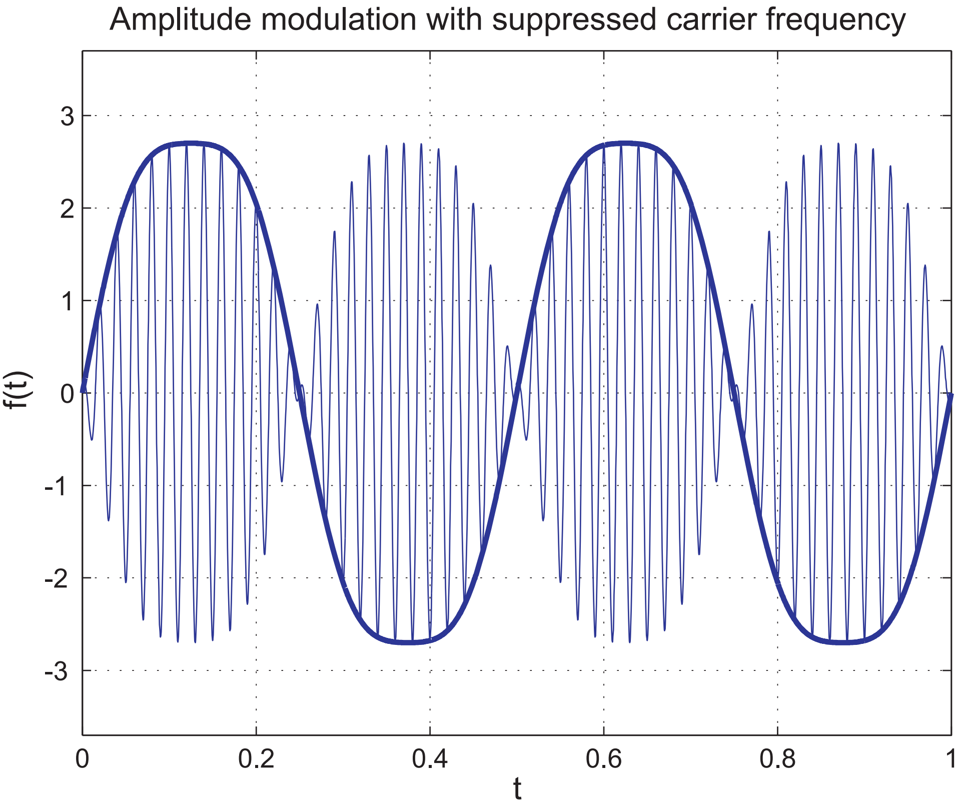
A phased signal is created by slightly delaying it in order to cause its peaks and troughs to be out of sync with the reference signal. The level of non-synchronization is measured in degrees. If a signal is 180º out of phase, it means that as the reference signal reaches zero voltage following a peak, the phased signal begins.

The importance in looking at frequency, amplitude and phase, lies in the fact that it is these components that can be varied in order to allow an analog signal to carry data.

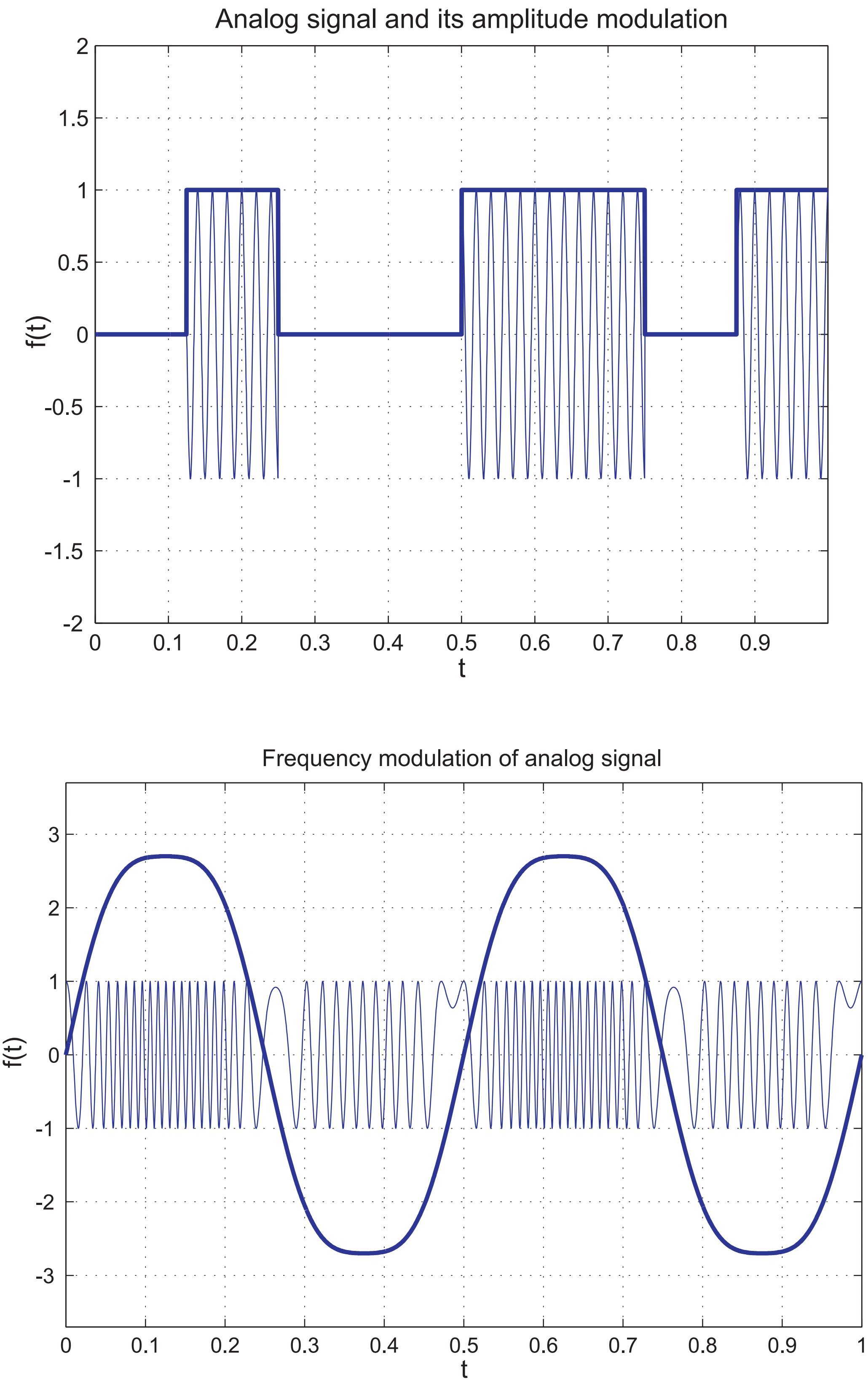
Modulation is the process of conveying a message signal, for example a digital bit stream or an analog audio signal, inside another signal that can be physically transmitted.

The main purpose of modulation is to enable transmission of many signals in a channel with limited bandwidth. Signals are modulated based on requirement for certain sub-bandwidth availability. The main advantage is that one transmission medium (for example one optical cable) is shared by many signals.

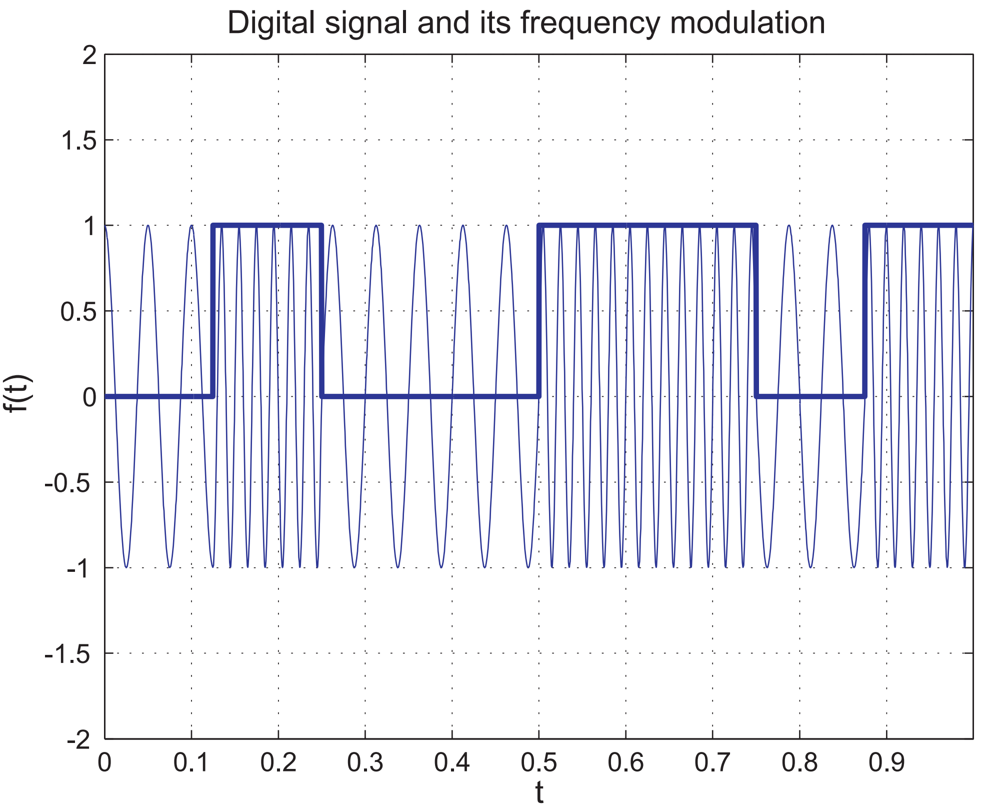
In analog modulation, the modulation is applied continuously in response to the analog information signal. There are many types of analog modulations that can be used, one of the simplest ones are *amplitude modulation* (**AM**), *phase modulation* (**PM**) and *frequency modulation* (**FM**).

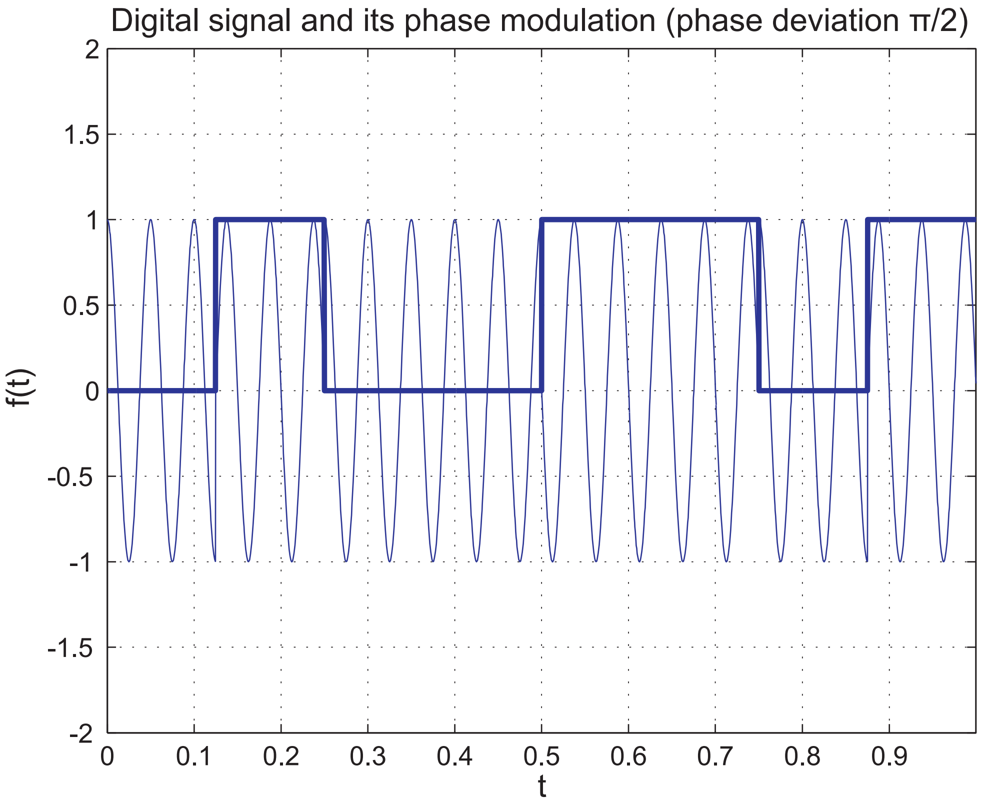


Amplitude modulation of analog and digital signal



Frequency modulation of analog and digital signal





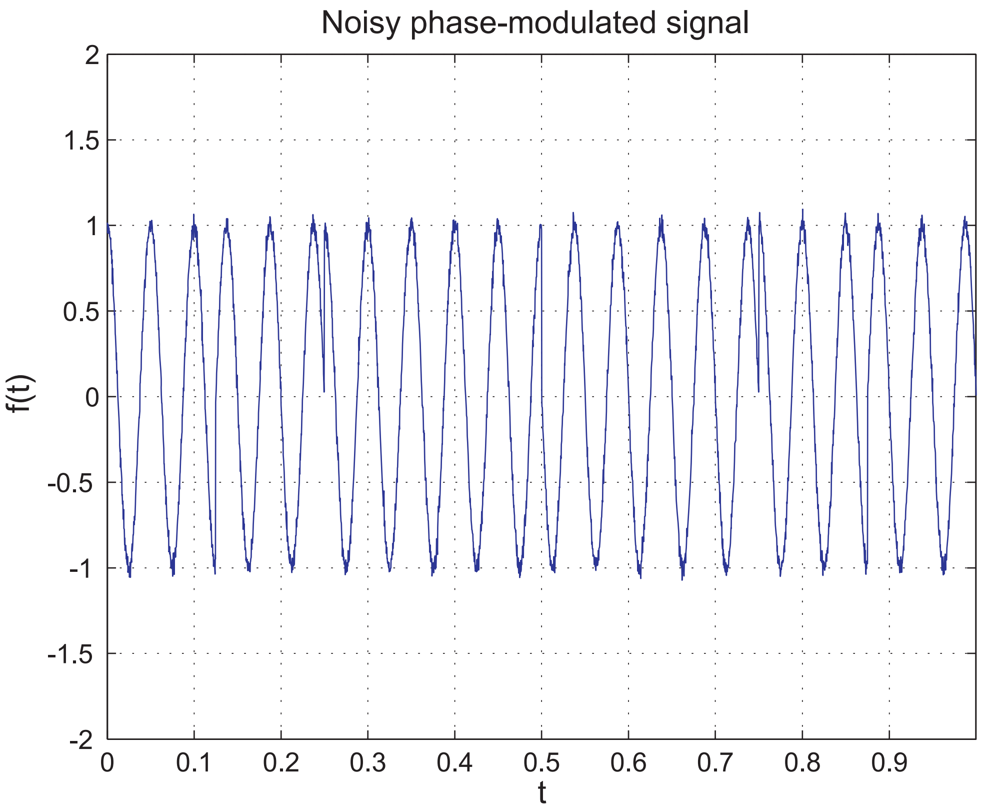
Phase modulation of digital signal

AM works by varying the strength of the transmitted signal in relation to the information being sent. The carrier wave or carrier is a waveform (usually sinusoidal) that is modulated (modified) with an input signal for the purpose of conveying information. This carrier wave is usually a much higher frequency than the input signal. Carrier wave has its amplitude modulated by an input signal (information which needs to be transmitted) before transmission. The input waveform modifies the amplitude of the carrier wave and determines the envelope of the waveform.

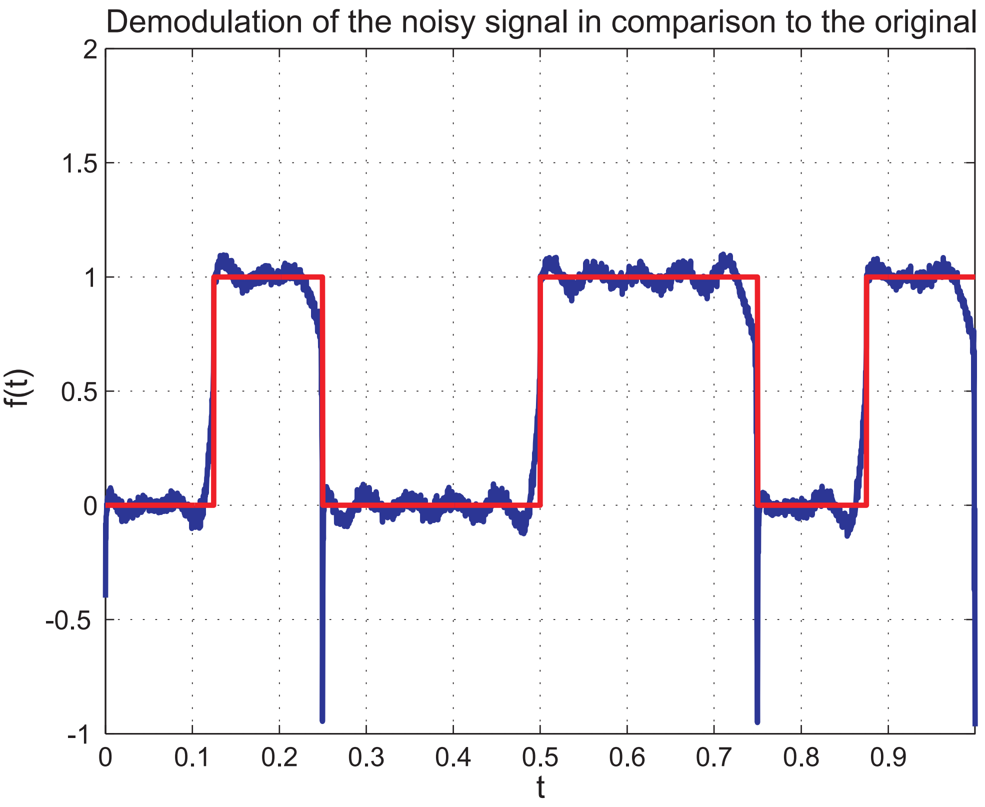
FM conveys information over a carrier wave by varying its momentary frequency. In analog applications, the difference between the momentary and the base frequency of the carrier is directly proportional to the instantaneous value of the input signal amplitude.

PM is a form of modulation that represents information as variations in the instantaneous phase of a carrier wave. We can say that modification in phase according to low frequency will give phase modulation. PM is not very widely used for radio transmissions. This is because it tends to require more complex receiving hardware and there can be ambiguity problems in determining whether, for example, the signal has changed phase by +180° or –180°. PM is used, however, in digital music synthesizers.

Demodulation is extracting the information signal from a modulated carrier wave. There are several ways of demodulation depending on how parameters of the signal (amplitude, frequency or phase) are transmitted in the carrier signal. For example, for a signal modulated with a linear modulation, like AM, we can use a synchronous detector. On the other hand, for a signal modulated with an angular modulation, we must use an FM demodulator or a PM demodulator.



Received modulated signal



Demodulated signal compared to original signal

2 Time and frequency representation

2.1 Fourier Transformation

The Fourier transform, named by Joseph Fourier, is important in mathematics, engineering, and the physical sciences.

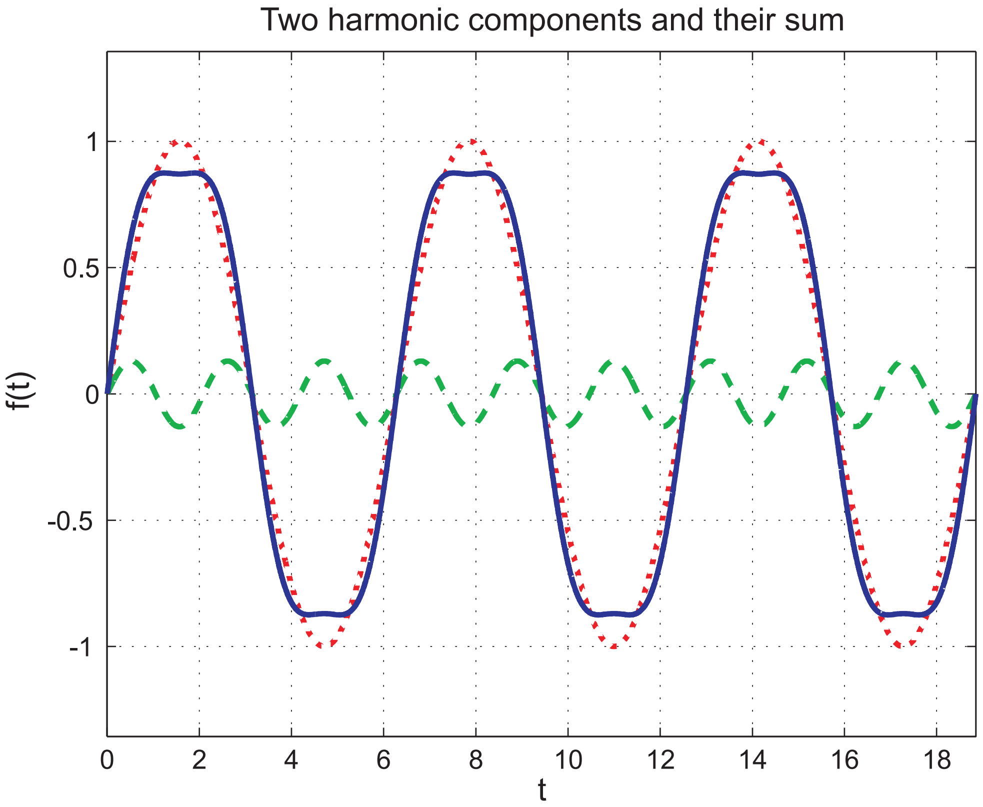
In simple way we can say that Fourier transformation represents mathematical function of time as a function of frequency. This function is known as frequency spectrum.

Very important note, Fourier Transformation is used only for non-periodic analog signals. In case of periodic analog signal Fourier series are used.

Let's say you have a function *f(t)* that maps some time value *t* to some value *f(t).*

Now, we try to approximate *f* as the sum of simple harmonic oscillations, i.e. sine waves of certain frequencies *ω*. Of course, there are some frequencies that fit well to *f* and some that approximate it less well. Thus we need some value *f(ω)* that tells us how much of a given oscillation with frequency ω is present in the approximation of *f*.

Take for example the function (black line) from here:



Two harmonic frequencies forming signal

which is defined as *f(t)=sin(t)+0.13sin(3t)*. The oscillation (dotted line) with *ω*=1 has the biggest impact on the result, so let's say *F(1)*=1. The other wave (*ω* = 3, dashed line) has at least some impact, but its amplitude is much smaller. Thus we say *F(3)*=0.13. Other frequencies may not be present in the approximation at all, thus we would write *F(ω)*=0 for these.

Now if we knew *F(ω)* not only for some but all possible frequencies *ω*, we could perfectly approximate our function *f(t)*. And that's what the continuous Fourier transform does.

It takes some function *f(t)* of time and returns some other function *F(ω)*=**FT**(*f*), it's *Fourier transformation*, that describes how much of any given frequency is present in *f*. It's just another representation of *f(t)*, of equal information but with a completely different domain. Often though, problems can be solved much easier in this other representation (which is like finding the appropriate coordinate system).

But given a Fourier transform, we can integrate over all frequencies, put together the weighted sine waves and get our *f* again, which we call *inverse Fourier transform* **IFT**.

Most importantly, the Fourier transform has many nice mathematical properties (i.e. convolution is just multiplication). It's often much easier to work with the

Fourier transforms than with the function itself. So we transform, have an easy job with filtering, transforming and manipulating sine waves and transform back after all.

Let's say we want to do some noise reduction on a digital image. Rather than manipulating a function image:Pixel→Brightness, we transform the whole thing and work with F(image):Frequency→Amplitude. Those party of high frequency that cause the noise can simply be cut off – *F*(image)(ω)=0,ω>...Hz.

The Fourier transform (usually known as forward transform) is defined as:

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

and inverse Fourier transformation is defined:

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

Fourier function *F(ω)* is frequency representation of signal *f(t)*, also called spectral function. Spectral function depends on real variable ω so it can be defined as:

$$F(\omega)=A(\omega)+jB(\omega)=Re\{F(\omega)\}+jIm\{F(\omega)\}=|F(\omega)|e^{j\varphi(\omega)}$$

where $|F(\omega)|$ is absolute value and $\varphi(\omega$ is phase spectrum.

In every formula *j* is defined as $j=\sqrt{-1}$. The complex exponential is the heart of the transform. A complex exponential is simply a complex number where both the real and imaginary parts are sinusoids. The exact relation is called Euler's formula $e^{j\varphi} = cos\varphi + jsin\varphi$, which leads to the famous (and beautiful) identity $e^{j\pi}+1=0$. Complex exponentials are much easier to manipulate than trigonometric functions, and they provide a compact notation for dealing with sinusoids of arbitrary phase, which form the basis of the Fourier transform.

Complex exponentials (or sinus and cosines) are periodic functions, and the set of complex exponentials is complete and orthogonal. Thus the Fourier transform can represent any piecewise continuous function and minimizes the least-square error between the function and its representation.

There exist other complete and orthogonal sets of periodic functions; for example, Walsh functions (square waves) are useful for digital electronics (more about it in chapter 2.4).

Why do we always encounter complex exponentials when solving physical problems? Why are monochromatic waves sinusoidal, and not periodic trains of square waves or triangular waves? The reason is that the derivatives of complex exponentials are just rescaled complex exponentials. In other words, the complex exponentials are the functions of the differential operator. Most physical systems obey linear differential equations. Thus an analog electronic filter will convert a sine wave into another sine wave having the same frequency (but not necessarily the same amplitude and phase), while a filtered square wave will not be a square wave. This property of complex exponentials makes the Fourier transform uniquely useful in fields ranging from radio propagation to quantum mechanics.

Fourier transformation is defined also for two-dimensional signals and mathematically is defined as:

$$F\left(\omega_{1,}\omega_2\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_{1,}x_2)e^{-j\omega_1}e^{-j\omega_2}\,dx_1dx_2$$

and inverse transformation is defined:

$$f\left(x_{1,}x_2\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F\left(\omega_{1,}\omega_2\right)e^{j\omega_1}e^{j\omega_2}\,d\omega_1 d\omega_2$$

2.2 Discrete Fourier Transformation

Discrete Fourier transformation **DFT** is used to get spectrum of discrete signals. Only a finite number of sinusoids are needed to describe signal in frequency domain.

To illustrate what DFT does we use following example. MP3 player sends the speaker audio information as fluctuations in the voltage of an electrical signal. The result are moving air particles and producing sound. An audio signal's fluctuations over time can be depicted as a graph: the x-axis is time, and the y-axis is the voltage of the electrical signal. This look like erratic wavelike squiggle which in real is sum a number of more regular squiggles, which represent different frequencies of sound. Frequency just means the rate at which air molecules go back and forth, or a voltage fluctuates

"The DFT does mathematically what the human ear does physically: decompose a signal into its component frequencies. Unlike the analog signal from, say, a record player, the digital signal from an MP3 player is just a series of numbers, representing very short samples of a real-world sound: CD-quality digital audio recording, for instance, collects 44,100 samples a second. If you extract some number of consecutive values from a digital signal –8, or 128, or 1,000 – the DFT represents them as the weighted sum of an equivalent number of frequencies. ("Weighted" just means that some of the frequencies count more than others toward the total.) "

Discrete Fourier Transformation (DFT) is mathematically defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-n2\pi jk/N}$$

and inverse discrete Fourier transformation (**IDFT**) is defined as:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{n2\pi jk/N}$$

where N is number of samples in discrete signal and n = 0, 1, 2, ..., N–1, $e^{2\pi jk/N}$ is very often substituted as Ω (Ω = $e^{2\pi jk/N}$).

The result of the DFT of an N-point input time series is an N-point frequency spectrum, with Fourier frequencies *k* ranging from – (*N*/2 – 1), through the 0-frequency or so-called direct component, and up to the highest Fourier frequency *N*/2. Each bin number represents the integer number of sinusoidal periods present in the time series. The amplitudes and phases represent the amplitudes A_k and phases $\varnothing_k$ of those sinusoids. In summary, each bin can be described by $X(k) = A_k e^{j\varnothing_k}$.

Discrete Fourier transformation is defined also for two-dimensional signals and it can be represented as the series expansion of an image function (over the 2D space domain).

Definition of forward and inverse **2D FT** is following:

$$X\left(\Omega_1,\Omega_2\right)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}x(n_1,n_2)e^{-j\Omega_1 n_1}e^{-j\Omega_2 n_2}$$

$$x\left(n_1,n_2\right)=1/(2\pi)^2\int_{\Omega_1=-\pi}^{\pi}\int_{\Omega_1=-\pi}^{\pi}X\left(\Omega_1,\Omega_2\right)e^{j\Omega_1 n_1}e^{j\Omega_2 n_2}d\Omega_1 d\Omega_2$$

Spectrum $X\left(\Omega_1,\Omega_2\right)$ is complex even if the sequence $x\left(n_1,n_2\right)$ is real. Spectrum can be defined also as sum of real and imaginary part or as product of magnitude and phase.

$$X\left(\Omega_1,\Omega_2\right)=\left|X\left(\Omega_1,\Omega_2\right)\right|e^{j\theta_x\left(\Omega_1,\Omega_2\right)}=X_R\left(\Omega_1,\Omega_2\right)+jX_I\left(\Omega_1,\Omega_2\right)$$

2.3 Spectrum

The frequency spectrum of a time-domain signal is a representation of that signal in the frequency domain. The frequency spectrum can be generated via Fourier transform (or discrete Fourier transform) of the signal, and the resulting values are usually presented as amplitude and phase, both plotted versus frequency.

Any signal that can be represented as amplitude that varies with time has a corresponding frequency spectrum. This includes familiar concepts such as visible light (color), musical notes, radio/TV channels, and even the regular rotation of the earth. When these physical phenomena are represented in the form of a frequency spectrum, certain physical descriptions of their internal processes become much simpler. Often, the frequency spectrum clearly shows harmonics, visible as distinct spikes or lines that provide insight into the mechanisms that generate the entire signal.

Spectrum analysis is the technical process of decomposing a whole signal into simpler parts. As described above, many physical processes are best described as a sum of many individual frequency components. Any process that quantifies the various amounts (e.g. amplitudes, powers, intensities, or phases), versus frequency can be called spectrum analysis.

Spectrum analysis can be performed on the entire signal (usually periodic signal). Alternatively, a signal (mainly non periodic or quasi-periodic) can be broken into short segments, called frames, and spectrum analysis may be applied to these individual segments.

The Fourier transform of a function produces a frequency spectrum which contains all of the information about the original signal, but in a different form. This means that the original function can be completely reconstructed (synthesized) by an inverse Fourier transform ation.

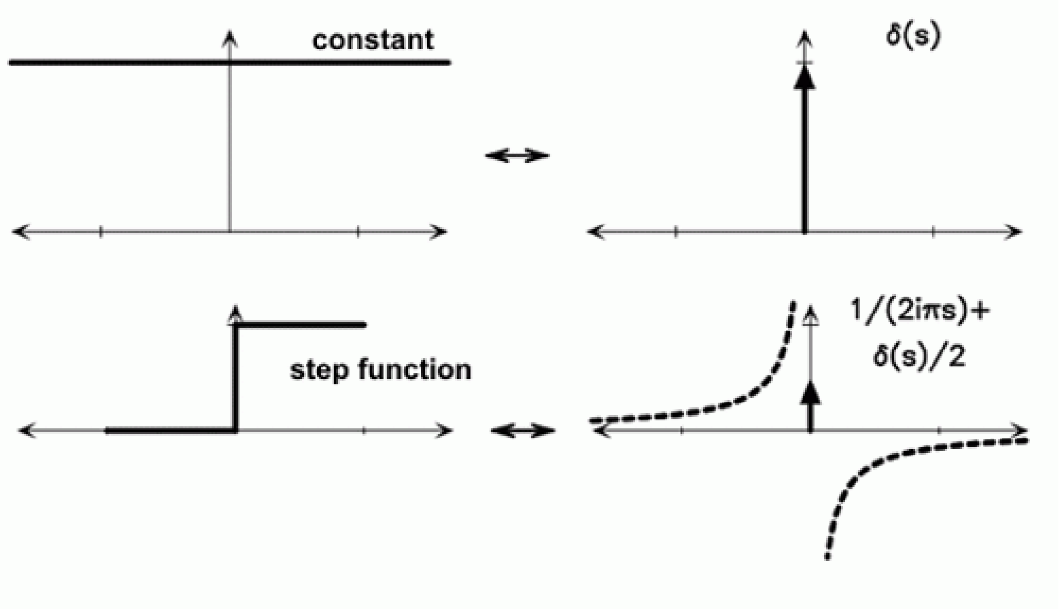
For perfect reconstruction, the spectrum analyzer must preserve both the amplitude and phase of each frequency component. These two pieces of information can be represented as a 2-dimensional vector, as a complex number, or as magnitude (amplitude) and phase in polar coordinates. A common technique in signal processing is to consider the squared amplitude, or power; in this case the resulting plot is referred to as a power spectrum.

The following table summarizes types of signals and their spectrum.

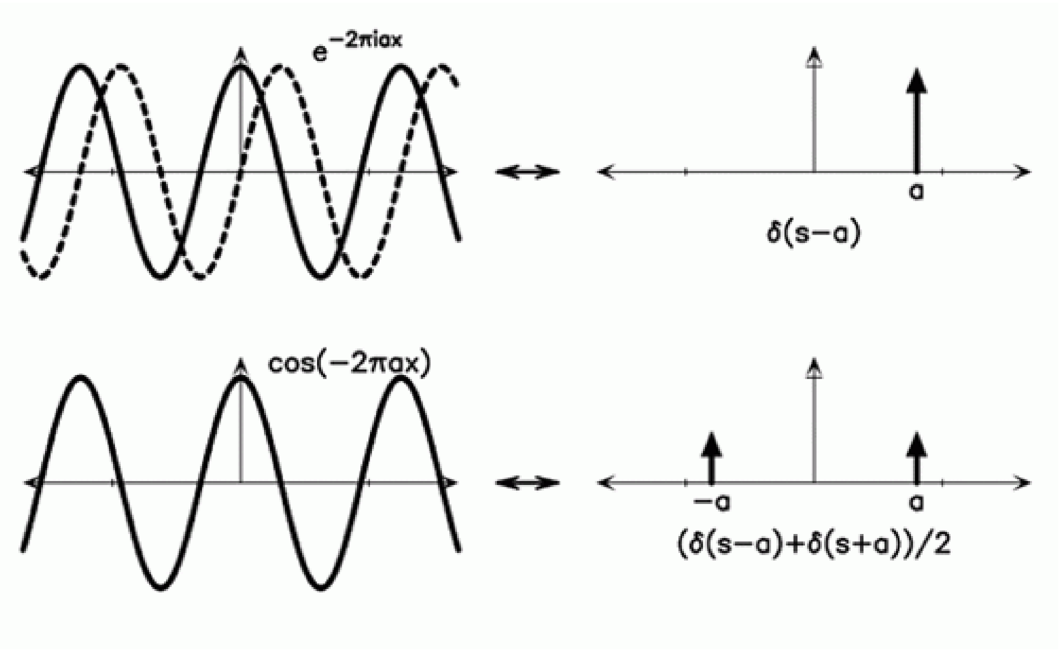
Signals and their spectrum

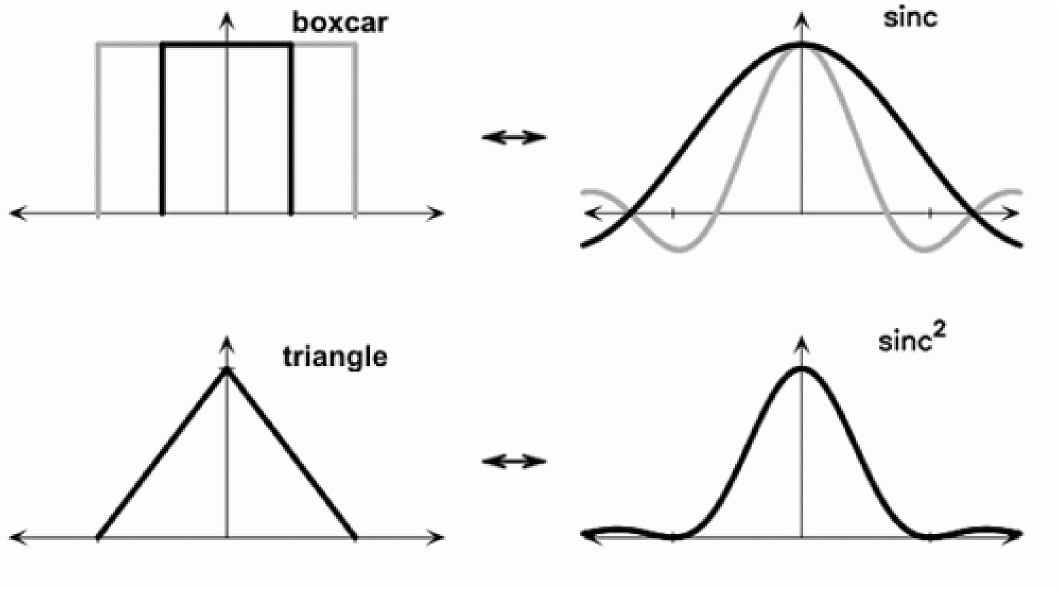
Signal	Spectrum
continuous periodic	discrete non-periodic
continuous non-periodic	continuous non-periodic
discrete periodic	discrete periodic
discrete non-periodic	continuous periodic

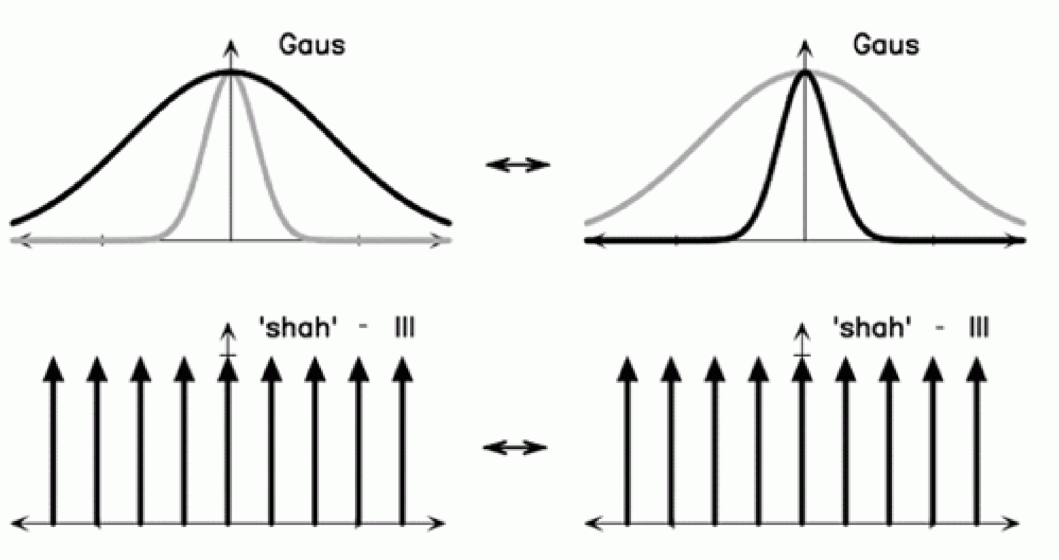
The following images show basic Fourier transform pairs (only amplitudes are present in the pictures). These can be combined using the Fourier transform theorems below to generate the Fourier transforms of many different functions.



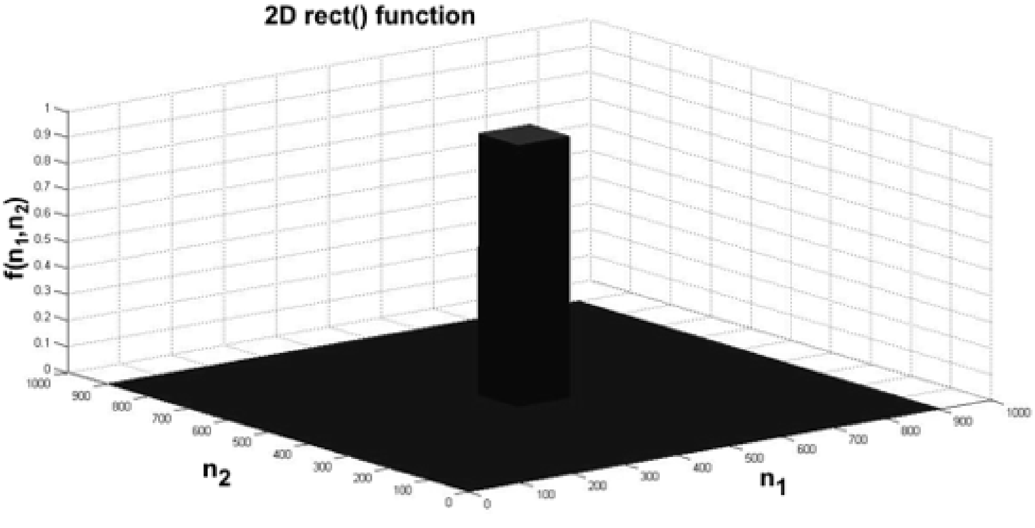
Basic Fourier transforms pairs


$e^{-2\pi iax}$
$\delta(s-a)$
a
$\cos(-2\pi ax)$
−a
a
$(\delta(s-a)+\delta(s+a))/2$

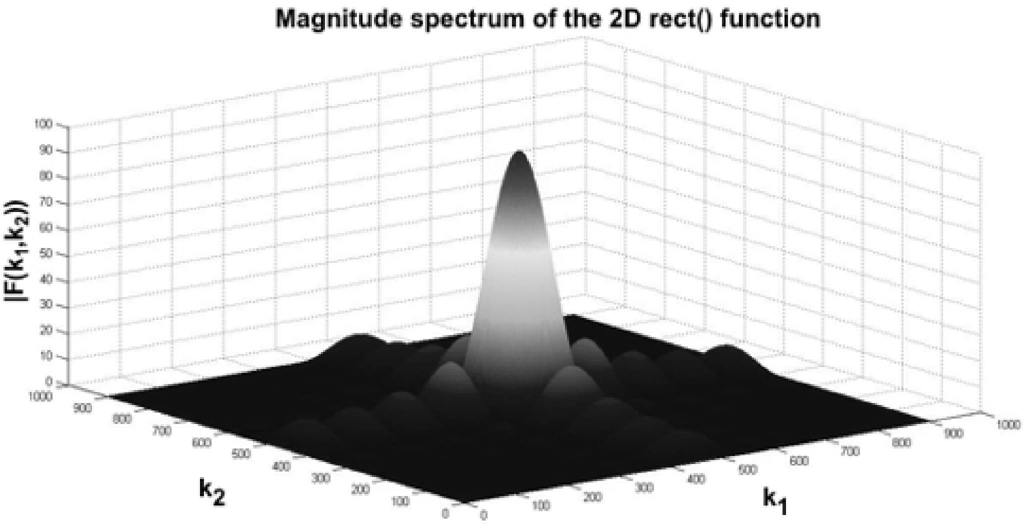

boxcar
sinc
triangle
sinc^2



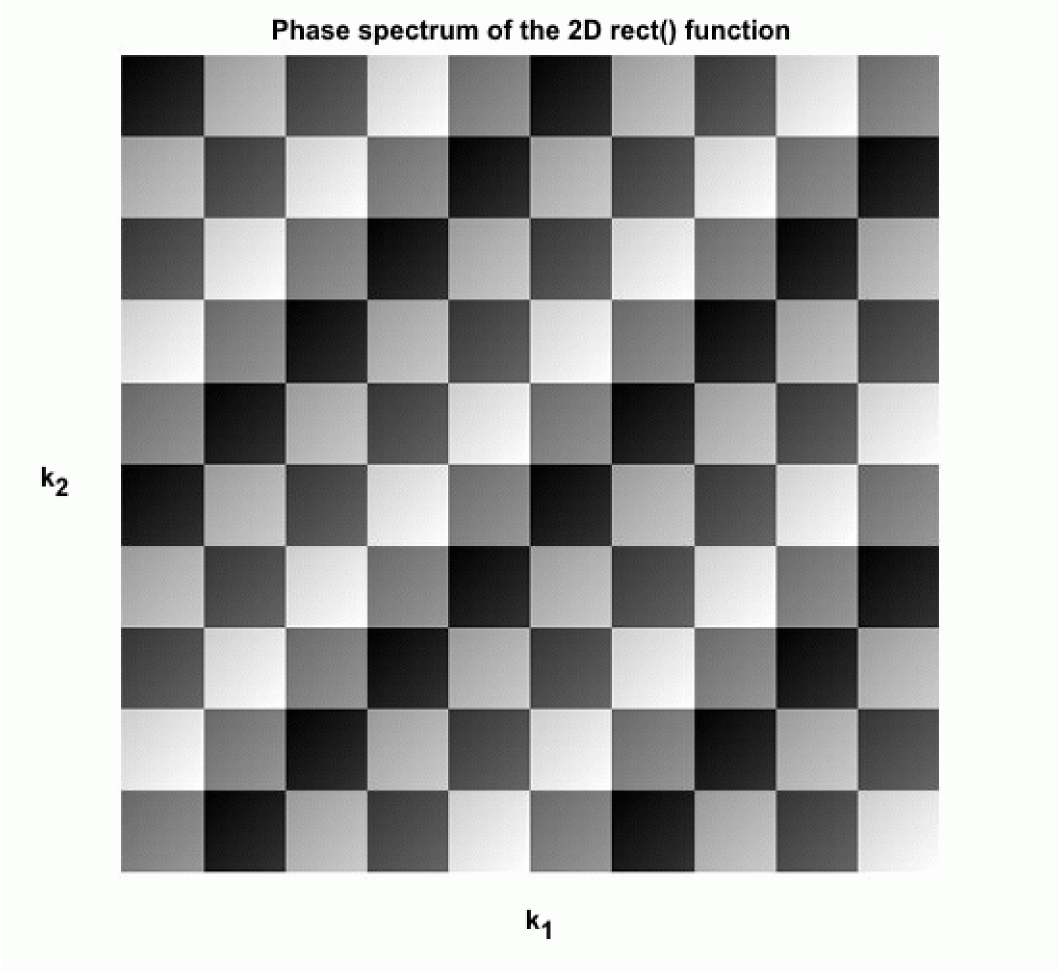
The following pictures display two-dimensional signals and their spectrum (magnitude and phase characteristic).



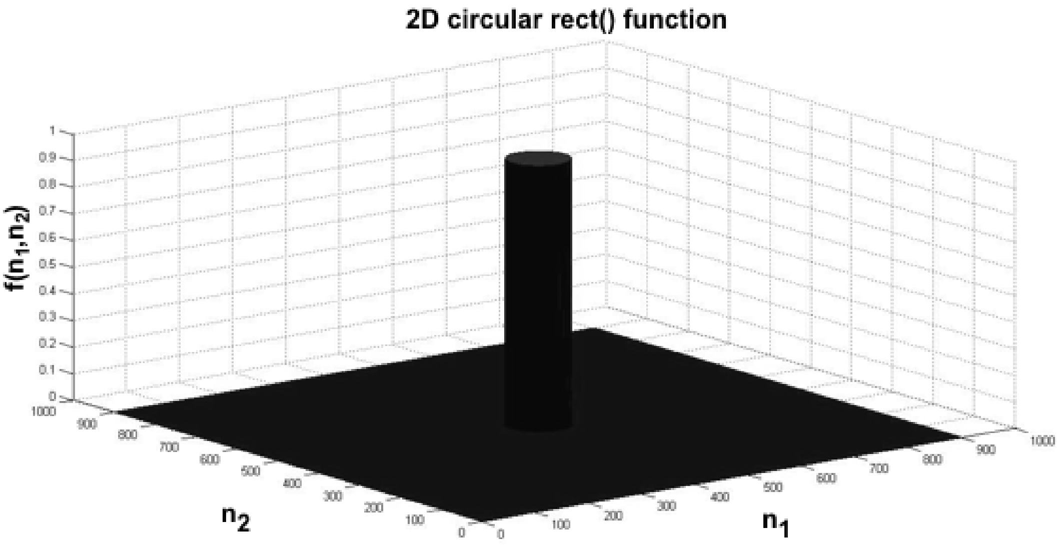
2D rectangular function, magnitude and phase frequency characteristic



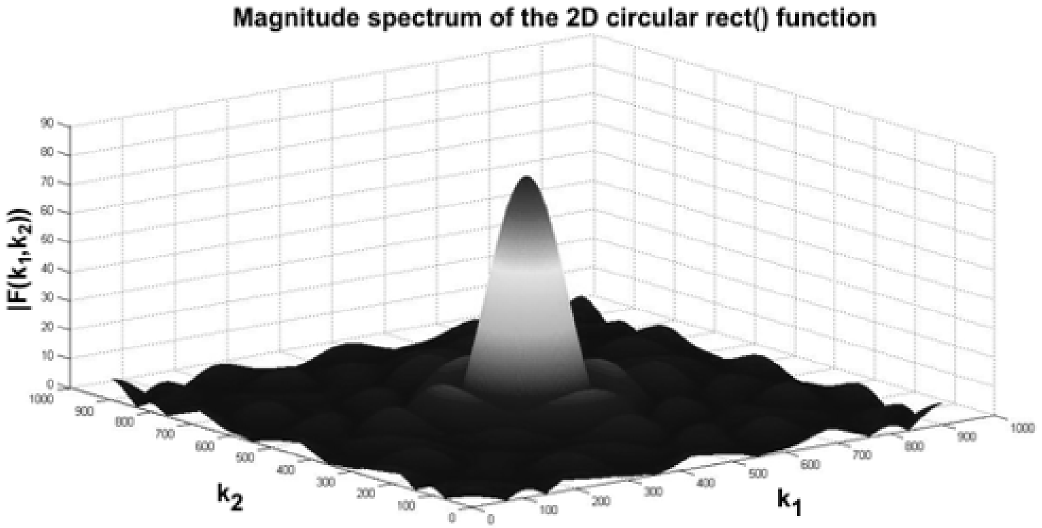
Magnitude spectrum of the 2D rect() function

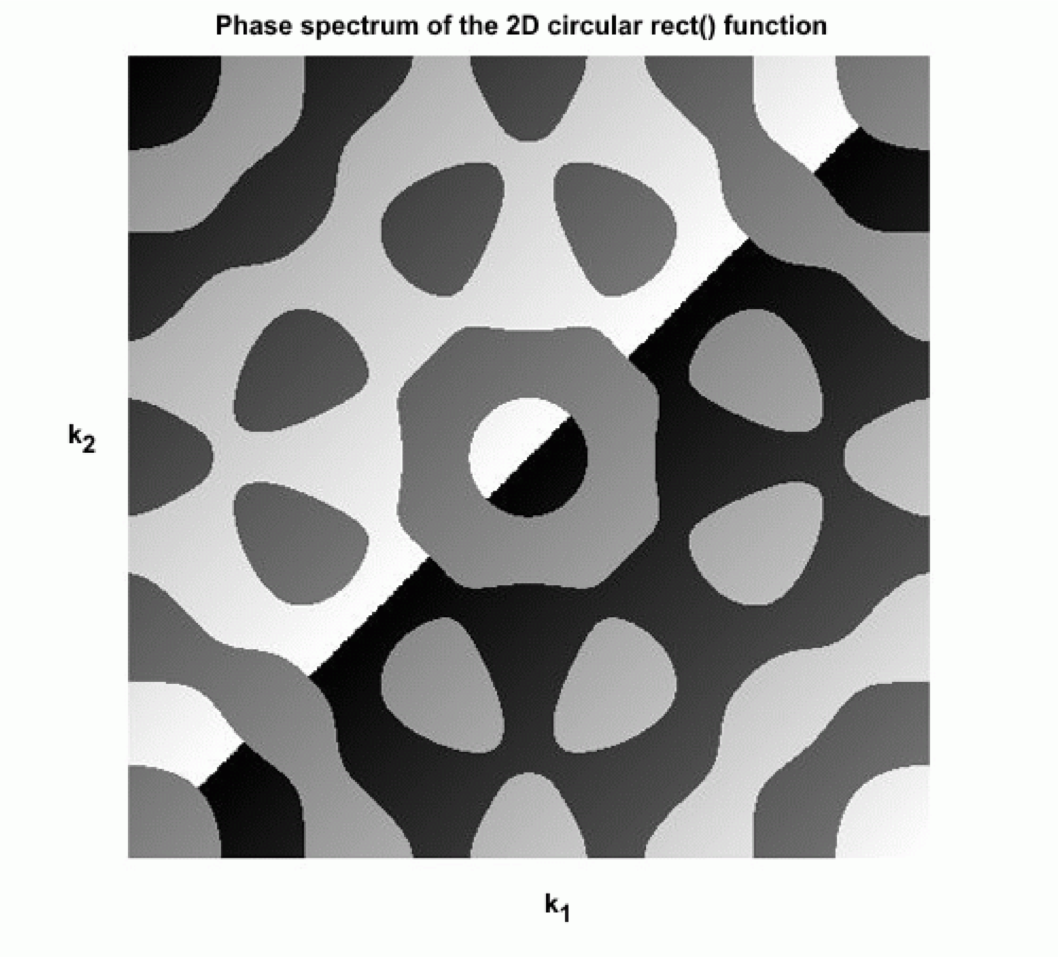


Phase spectrum of the 2D rect() function



2D circular function, magnitude and phase frequency characteristic




Phase spectrum of the 2D circular rect() function
k_2
k_1

2.4 Other Transformations

Orthogonal transformations, in general, allow representing any function of time in spectral domain where it's easier to work with signals and make some mathematical operations (as convolution, reducing redundancy, etc.).

Discrete orthogonal transformations are mainly used in area of data compression, picture recognition, speech synthesis, etc. Our focus is only on one-dimensional orthogonal functions and transformations.

The simplest way how to express one-dimensional signal is as combination of bases functions. The main advantage for linear systems is the superposition principle (or superposition property). It's desirable for the basis functions *u(k, t)* to be easily calculated and in simple form. Orthogonal functions fulfill all these requirements.

Mathematically, the orthogonal functions *u(*0*,t), u(*1*,t),…, u(N –* 1*,t)* on time interval <t1, t2> are defined as:

$$\int_{t_1}^{t_2} u(k,t)\,u(m,t)\,dt = 0, \qquad k \neq m$$

$$\int_{t_1}^{t_2} u^2(k,t)\,dt = U_k, \qquad k \neq m$$

In case *Uk* = 1 the functions are orthonormal.

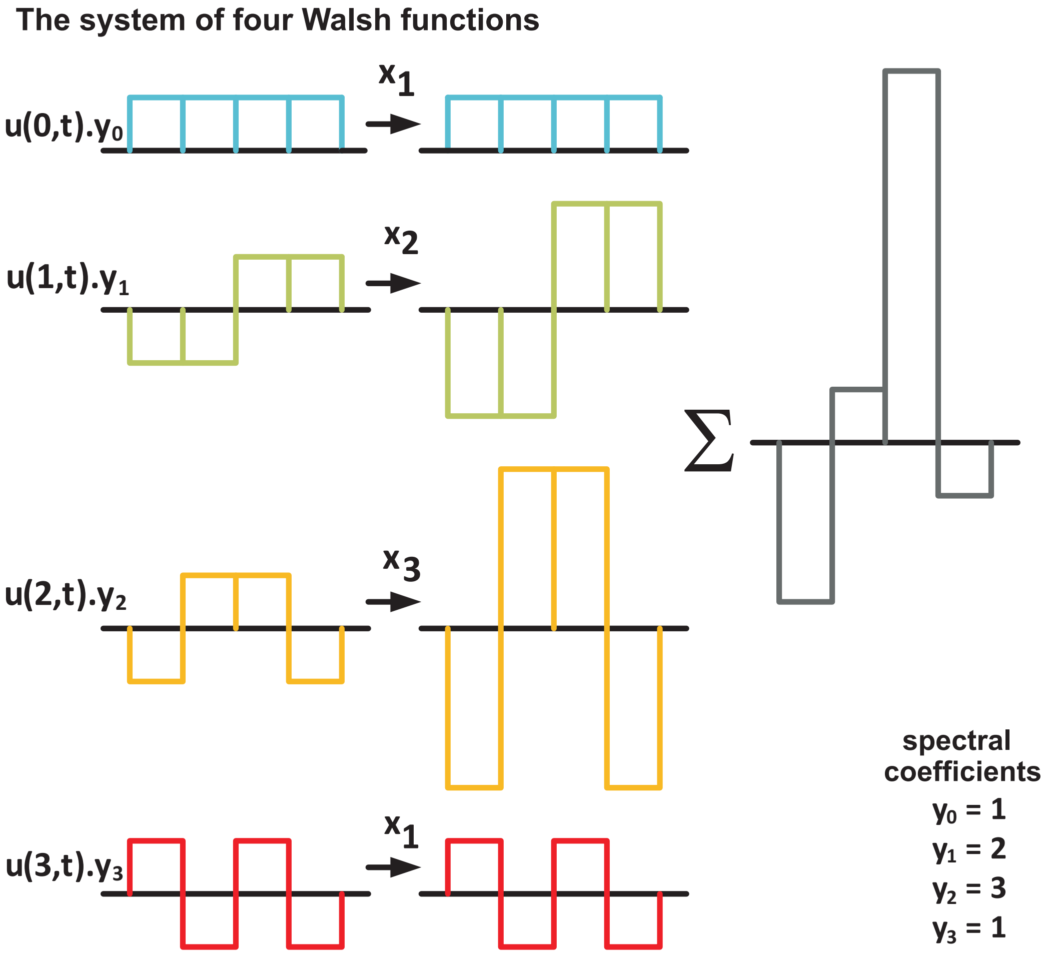
Signal *x(t)* approximated with basis orthogonal functions using superposition principle in mathematical language is as follows:

$$x(t) \cong \sum_{k=0}^{N-1} y_k u(k,t)$$

Where y_k are spectral coefficients defined as:

$$y_k = \frac{1}{U_k} \int_{t_1}^{t_2} x(t)\,u(k,t)\,dt$$

Please read following example for better understanding.



Example of signal x approximated with basis orthogonal functions using superposition principle

The most common orthogonal functions used in signal processing are Walsh, Haar and Rademacher.

In case of discrete orthogonal functions approximation of signal *x(nT)*, where *T* is period with *M* samples, is given as

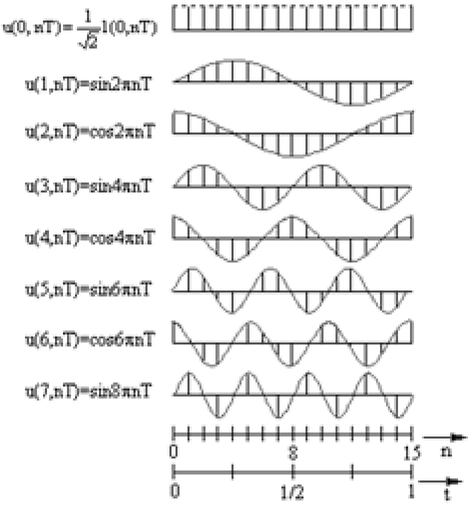
$$x(nT) = \sum_{k=0}^{N-1} y_k u(k,nT) \qquad n = 0,1,\ldots,M-1$$

Optimal coefficients are defined as:

$$y_k = \sum_{n=0}^{M-1} x(nT)u(k,nT)$$

In case of harmonic functions the main parameter of function is frequency. In case of non-harmonic functions the main parameter is sequence. Sequence is given as number of intersections with zero level per second. In case of discrete signals sequence is given as number of changes from negative to positive and vice versa, also per second.

Sampled orthogonal basis functions create system of discrete orthogonal basis functions.

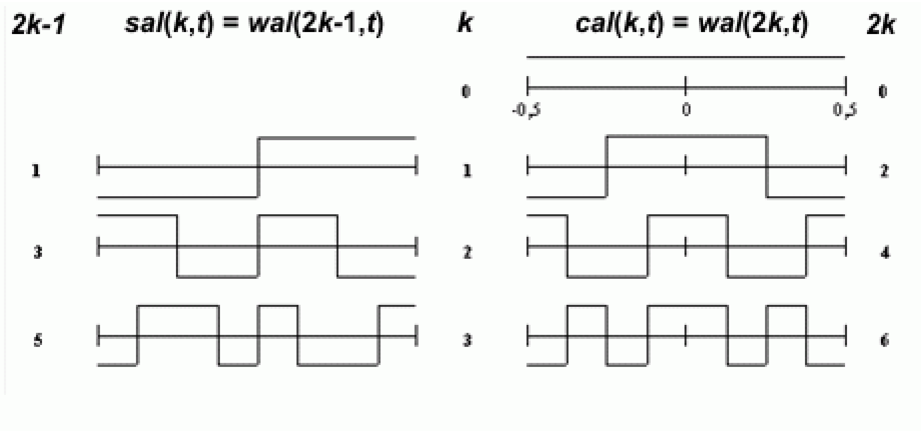


Walsh basis functions

Walsh functions are set of the square integrable functions on the unit interval. The functions take the values +1 and –1 only. Functions are time (*t*) and numerical (*k*) dependent. Generally are marked as *wal(k,t)* and can be grouped as even (cosines-Walsh) *cal(k,t)* and odd (sinus-Walsh) *sal(k,t)*. Mathematical definition is also:

$$cal(k,t)=wal(2k,t), \qquad k=0,1,2,\ldots$$

$$sal(k,t)=wal(2k-1,t), \qquad k=0,1,2,\ldots$$



Even and odd Walsh functions

Based on ordering of Walsh functions three groups can be distinguished:

1. Walsh (sequence) ordering *walw(k,t)*
2. Dyadic (Paley) ordering *walp(k,t)*
3. Hadamard (natural) ordering *walh(k,t)*

All three groups contain the same functions but in different order. As example we will introduce you natural ordered Hadamard functions.

The main advantage of this ordering is very simple way how to create basis functions of higher dimensions.

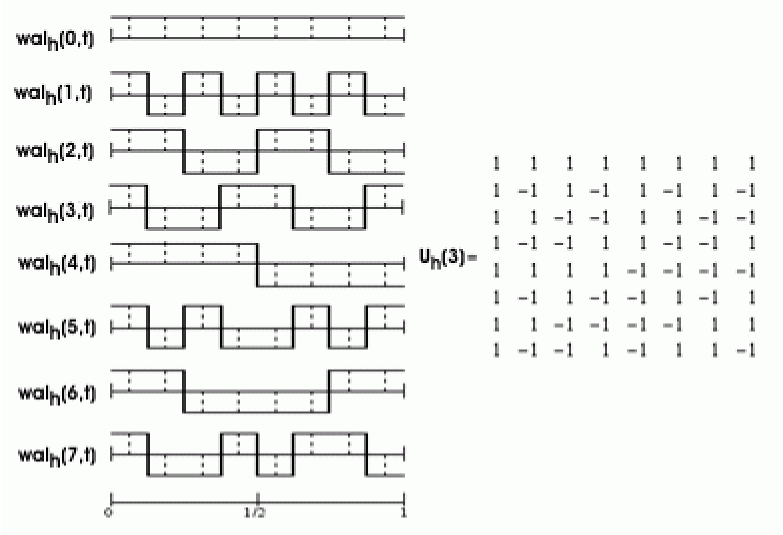
In the picture below, there are first eight continuous and discrete Walsh functions, naturally ordered. If we write all values of discrete functions, the values create Hadamard matrix $U_h(3)$ with dimensions 8x8.

In general Hadamard matrix $U_h(r)$ with dimensions MxM, where $M=2^r$, is calculated as Kronecker product of matrixes from $U_h(r-1)$.

$$\mathrm{Uh(r)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \mathrm{Uh(r\text{-}1)} = \begin{bmatrix} \mathrm{Uh(r\text{-}1)} & \mathrm{Uh(r\text{-}1)} \\ \mathrm{Uh(r\text{-}1)} & \mathrm{-Uh(r\text{-}1)} \end{bmatrix}$$

where $U_h(0)=1$

Kronecker product, denoted by $\otimes$, is an operation on two matrices of arbitrary size resulting in a block matrix.



$$U_h(3)= \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{matrix}$$

Natural ordering of continuous and discrete Walsh functions and matrix made of function values

There are also other transformations with harmonic basis functions. Here belong (except Discrete Fourier transformation) *discrete cosines transformation* (**DCT**), *discrete sinus transformation* (**DST**), *discrete Hartley transformation* (**DHYT**).

3 Analog and Digital Technologies

3.1 Multiplexing

Multiplex combines several analog or digital signals and forwards the selected input into a single line, in telecommunication into telecommunication link. Multiplexers (or mux) are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth.

On the other hand, a demultiplexer (or demux) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. A multiplexer is often used with a complementary demultiplexer on the receiving end.

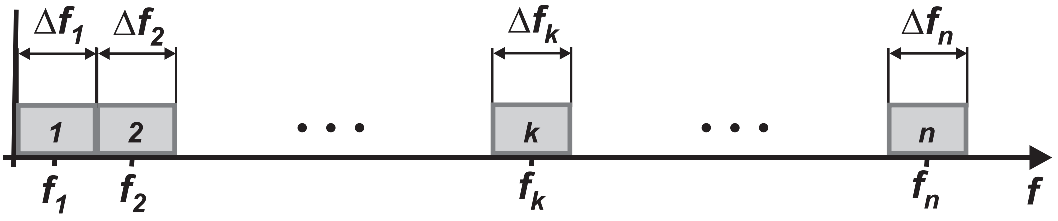
Frequency division multiplex

Frequency division multiplexing (**FDM**) is mainly an analog technology. Frequency division multiplex combines several digital signals into one medium by sending signals in several distinct frequency ranges over that medium.

+ Advantage of systems where channels are divided by frequency (a frequency multiplex) is that multiple signals belonging to multiple transmit sources and channels can be transmitted simultaneously. Another advantage is that signals can be transmitted via great distances.

− On the other hand, disadvantage of frequency multiplex is channel interference, which is a result of using real (not ideal) filters, and both channel and electric circuits crosstalk.

Frequency multiplex works by assigning each signal mutually non-overlapping frequency bands $\Delta f1$, $\Delta f2$, ..., Δfn. It is usually the case, that $\Delta f1 = \Delta f2$ ... $= \Delta fk = \Delta fn$. Of course, the frequency spectrum of transmitted signal has to fit the width provided by frequency band Δfn.



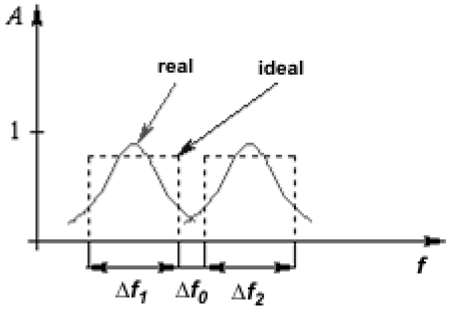
Frequency spectrum for multiple channels

The frequency range of the transmission path F determines frequency boundaries of channels, e.g. minimum frequency of interval $\Delta f1$ ($\Delta f1_{min}$) and maximum frequency of interval Δfn ($\Delta f\text{n}_{max}$).

$$F = \Delta f_{n\max} - \Delta f_{1\min}$$

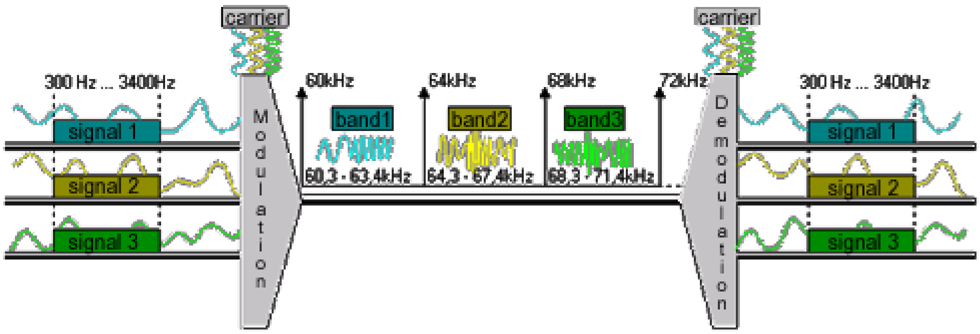
A crucial part of frequency multiplex system on both the transmitter and receiver side is a band-pass filter. A band-pass filter is a device that passes frequencies within a certain range and rejects (attenuates) frequencies outside that range. The bandwidth of the filter (Δfn) is simply the difference between the upper and lower cutoff frequencies. Selection of a proper band-pass filter depends mainly on the transmission speed.

Real band-pass filters do not have ideal characteristics and do not eliminate all frequencies above or below the cutoff frequency. Therefore a protective band Δf_0 must be in place. The ratio $\Delta\ f_0/\Delta f = kf$ depends on the amplitude-frequency characteristics of the band-pass filter in use (see image below) (more about filters in chapter Filters).



Amplitude frequency characteristic of ideal and real filter

The remaining question is, how do signals, which we want to transmit move to desired subband. It is via modulation on transmitter side and demodulation on receiver side. Each signal has a different carrier signal (see part about modulation in chapter Modulation), one that fits middle of the desired subband. It is later demodulated to its original form.



Frequency multiplex principle

Frequency multiplexing was mainly used in analog Telco systems, its current usage is mainly cable TV providers (each channel is on slightly different frequency – by switching a channel on a TV we tune to a different frequency band) and in optical communications.

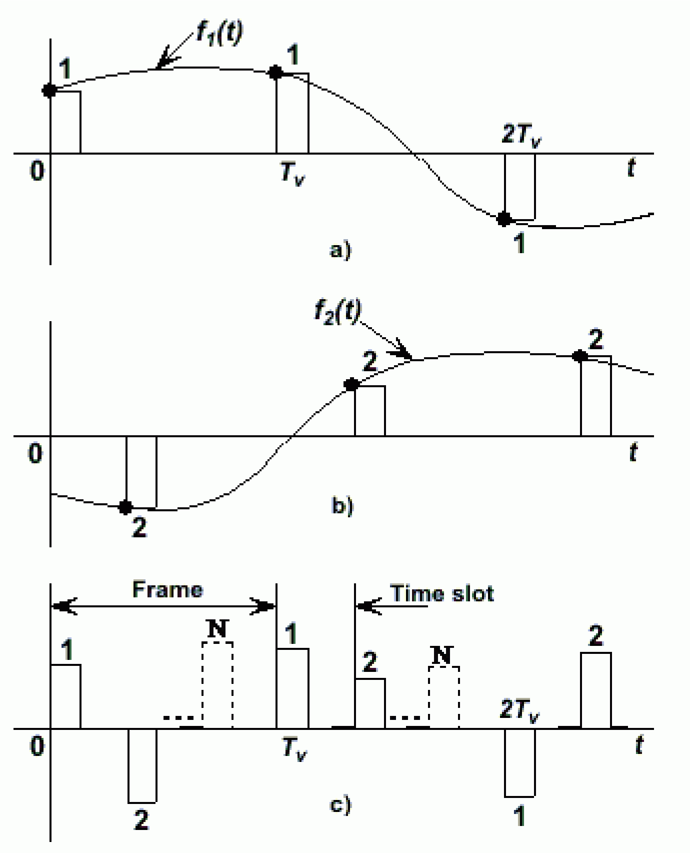
Time division multiplex

Time division multiplex (**TDM**) is used both in networking and phone systems and does exactly what the name says, e.g. takes samples from several slower speed signals, transmits it through one fast channel and restores the original signals.

The input device, also called multiplexer, selects one by one different source and takes a portion of its data and places it on the wire next. In this manner several "samplings" from several sources can be interleaved on the high-speed communications channel. This can be accomplished because the individual sources are sending their data at a relatively slow speed (i.e. 300 baud), while the outgoing channel has significant speed to accommodate a sampling from each source (i.e. 1200 baud). When the data reaches its transmit destination, another multiplexer disassembles the transmitted data and sends it to its destination, once again at the slower speed at which it entered the TDM system.

This technology is used by phone companies which have to put a large number of conversations over limited numbers of wires. If the conversations are broken up and put back together faster than human ear can detect, no one notices it. For this reason, high speed trunks use time-division multiplexing to carry several conversations at once.

One disadvantage of multiplexers that use time division multiplex is that they allocate a time slot even though the source is not sending any data or signal, which creates inefficiency.



Time division multiplex a) the first information b) the second information c) time slot multiplexing

Synchronization

The synchronization of the sampling process is crucial at both ends of the channel. TDM devices must synchronize with one another so that the time moment required for each sampling matches. Otherwise, the demultiplexer would not be able to determine which source signal goes with what destination channel. Timing is therefore an extremely important element to a time-based methodology like TDM.

The time synchronization is based on transmitting a reference time impulses by a transmitter. The receiver must synchronize itself based on the received impulses. There is also frame synchronization – each frame has a special position, usually designating beginning of a frame with a special character.

3.2 Linear discrete time-invariant systems

Linear time-invariant system (**LTI**) has direct applications in seismology, circuits, signal processing, control theory, and other technical areas. The analysis of the continuous-time LTI and discrete-time LTI are rather similar, but the discrete-time case involves fewer technicalities, so we concentrate on it.

LDTI (*Linear discrete time-invariant system*) takes one discrete input signal and produce one discrete output signal with the following properties:

- The system is linear. It means that if the input signals $x_1(n)$ and $x_2(n)$ generate output signals $y_1(n)$ and $y_2(n)$ and if a_1 and a_2 are constants, then the input signal $a_1x_1(n)+ a_2x_2(n)$ generate output signal $a_1y_1(n)+ a_2y_2(n)$.
- The system is time invariant. This means that if the input signal $x(n)$ generates the output signal $y(n)$, then, for each real number s, the time shifted input signal $\hat{x}(t) = x(t-s)$ generates the time shifted output $\hat{y}(t) = y(t-s)$.

The difference equation

The first important fact concerning the behavior of linear discrete time-invariant system is that the response of the system to any input is completely determined by its response to one special input, the Kronecker delta impulse (defined in chapter Important signals) at time 0. Let's denote the output by $h(n)$ that results from the Kronecker delta impulse at time 0. It is called the impulse response of the system. We'll now define a formula that expresses the output generated by any input $x(n)$.

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

This equation describes recursive LDTI. Samples of output signal are linear combination of input signal and weighing coefficients a_k and b_k. Systems described by this equation are with *infinite impulse response* (**IIR**).

Special case is differential equation for non recursive system. In this case the output signal is dependent only on input signal, not on previous samples of output signal. This system is with *finite impulse response* (**FIR**) and mathematically is described as:

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

Convolution

Convolution is other possibility how to describe LDTI. If the impulse response $h(n)$ of LDTI is known and input signal is $x(n)$, than the output signal is given as:

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

where operator * is convolution product. The length of the output signal is defined as $D_y = D_x + D_h - 1$ where D_x is the length of the input signal and D_h is length of the impulse response.

The principle is based on superposition in linear systems. The output signal is given as sum of weighted and shifted impulse responses. For easy understanding please read following example.

Let's have FIR system with impulse response h(n) = {1, 2, 3}. Input signal is given as *x(n)* = {*x*(0), *x*(1), *x*(2), *x*(3)}.

Output signal is calculated by using convolution.

The length of *h(n)* is *Dh* =3 and length of input signal *x(n)* is *Dx* =4, then length of output signal is *Dy* =6. Convolution product can by calculated in table.

Convolution

n	**0**	**1**	**2**	**3**	**4**	**5**	**6**
x(0)	x(0)	2 x(0)	3 x(0)	0	0	0	0
x(1)		x(1)	2 x(1)	3 x(1)	0	0	0
x(2)			x(2)	2 x(2)	3 x(2)	0	0
x(3)				x(3)	2 x(3)	3x(3)	0
y(n)	y(0)	y(1)	y(2)	y(3)	y(4)	y(5)	0

In each table row are samples of input signal weighted with samples of impulse response. Rows are shifted right so it corresponds to delay of input signal. In last row are samples of output signal given as superposition of values in columns for *n* = 0, 1, 2,... For example :

y(1) = 2.x(0) + x(1)

y(2) = 3.x(0) + 2.x(1) + x(2), etc.

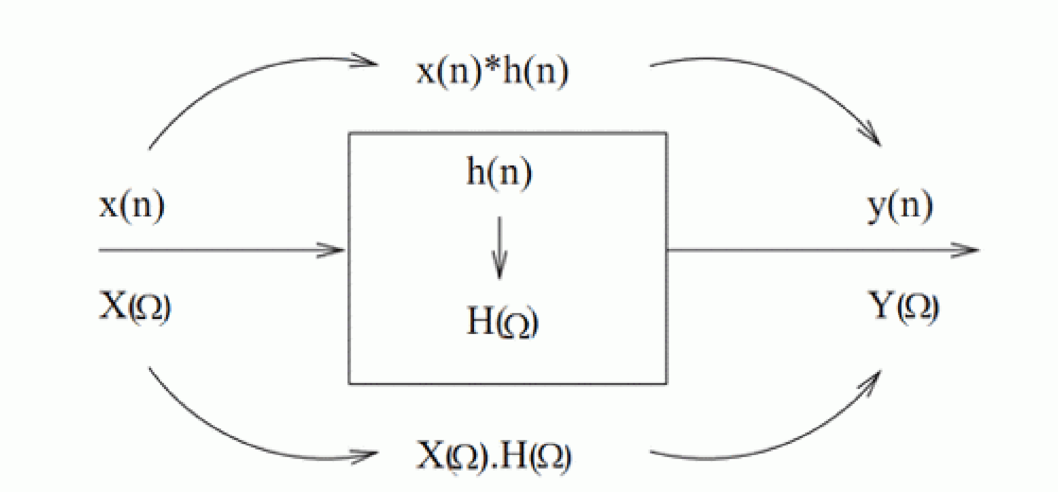
To get convolution of two signals, one signal is sort in reverse order and the other one is shifted from left to right. For each step the sum of products is calculated.

Transfer function

Transfer function $H(\Omega)$ represents the relation between the input and output of a LDTI system with zero initial conditions in frequency domain. Transfer function can be derivate from difference equation or impulse response. In both cases

Discrete Fourier transformation is used. It was defined before $y(n)=h(n)*x(n)$. After DFT of each component we get

$$Y(\Omega)=H(\Omega).X(\Omega)=>H(\Omega)=\frac{Y(\Omega)}{X(\Omega)}$$



Relation between input and output signal in time and frequency domain

It's obvious that *h(n)* and $H(\Omega)$ characterize the same system in different domains. In case DFT is applied to difference equation, transfer function is described by using weighted coefficient a_k and b_k which are the same as in difference equation:

$$H(\Omega)=\frac{a_0+a_1e^{-j\Omega}+a_2e^{-j2\Omega}+\ldots+a_Ne^{-jN\Omega}}{1+b_0+b_1e^{-j\Omega}+b_2e^{-j2\Omega}+\ldots+b_Me^{-jM\Omega}}.$$

Formula in fraction form is more convenient because the result of dividing numerator by denominator is samples of impulse response. In case of IIR system the number of samples is infinite. For FIR systems the formula is not fraction, because denominator is equal 1.

$$H(\Omega)=a_0+a_1e^{-j\Omega}+a_2e^{-j2\Omega}+\ldots+a_Ne^{-jN\Omega}$$

The transfer function is very important and based on $H(\Omega)$ frequency response is estimated. This is used mainly in filter theory.

Frequency response

Frequency response is used to characterize the dynamics of the system.

It is a measure of amplitude (or magnitude) and phase of the output as a function of frequency, in comparison to the input.

In simplest terms, if a sine wave is injected into a system at a given frequency, a linear system will respond at that same frequency with a certain magnitude and a certain phase angle relative to the input.

So for frequency response of transfer function $H(\Omega)$ is defined:

$$H(\Omega)=\left|H(\Omega)\right|.e^{j\varphi(\Omega)}=Re\left\{H(\Omega)\right\}+jIm\left\{H(\Omega)\right\}$$

where $\left|H(\Omega)\right|=\sqrt{Re\left\{H(\Omega)\right\}^{2}+Im\left\{H(\Omega)\right\}^{2}}$ and $\varphi(\Omega)=arctan\dfrac{Im\left\{H(\Omega)\right\}}{Re\left\{H(\Omega)\right\}}$.

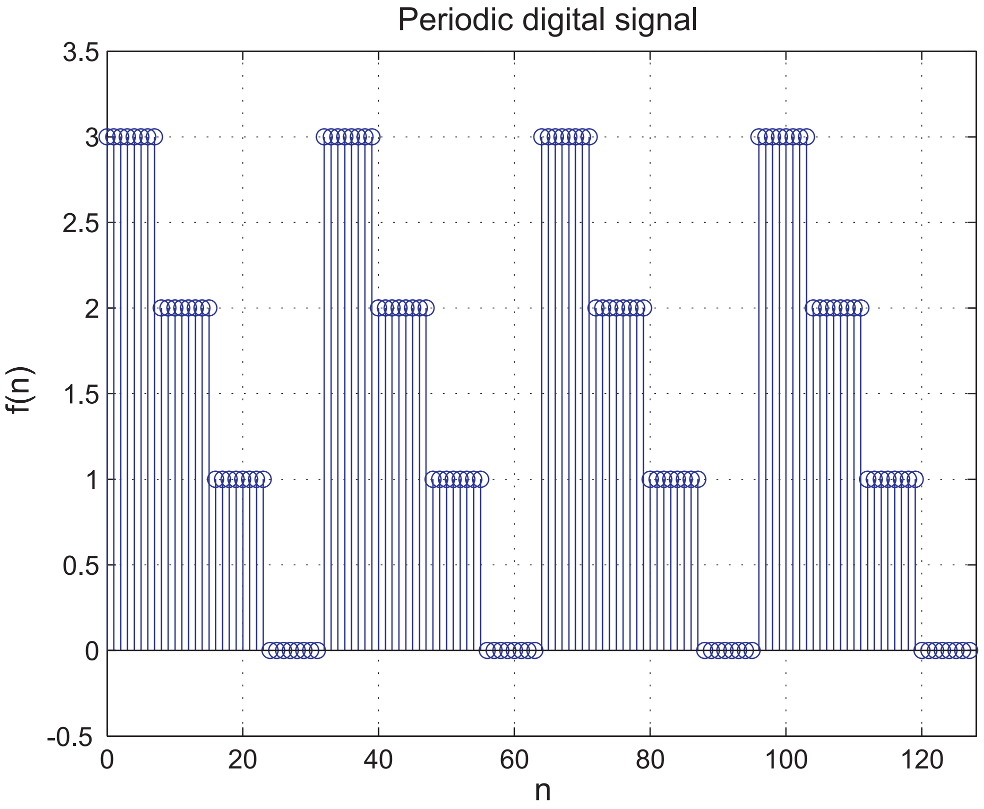
The absolute value of transfer function $\left|H(\Omega)\right|$ is called magnitude frequency characteristic and $\varphi(\Omega)$ is phase frequency characteristic. $\varphi(\Omega)$ is not continuous function but function with step changes about 180 degrees. These steps can be removed by change of mark of magnitude frequency characteristic "–" from "+" to – and vice versa. This change is done for each step change in $\varphi(\Omega)$. New phase frequency characteristic $\Theta(\Omega)$ is continuous function. The relation between amplitude and magnitude frequency characteristics following:

$$A(\Omega)=\pm M(\Omega).$$

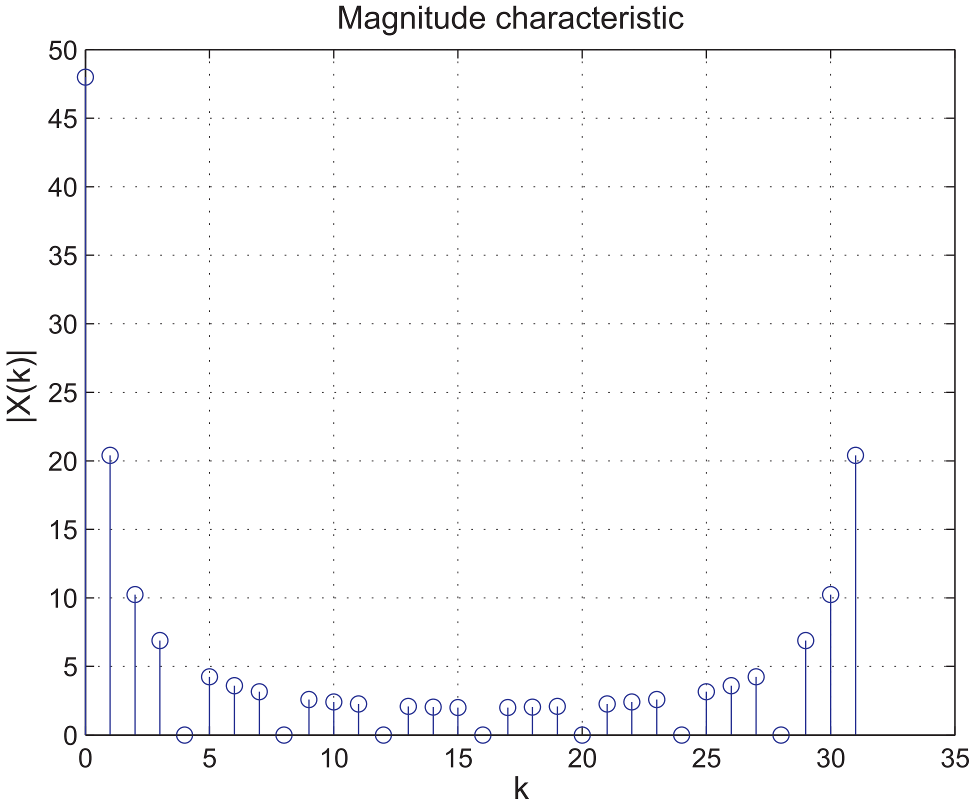
Based on information above following is defined for frequency characteristics:

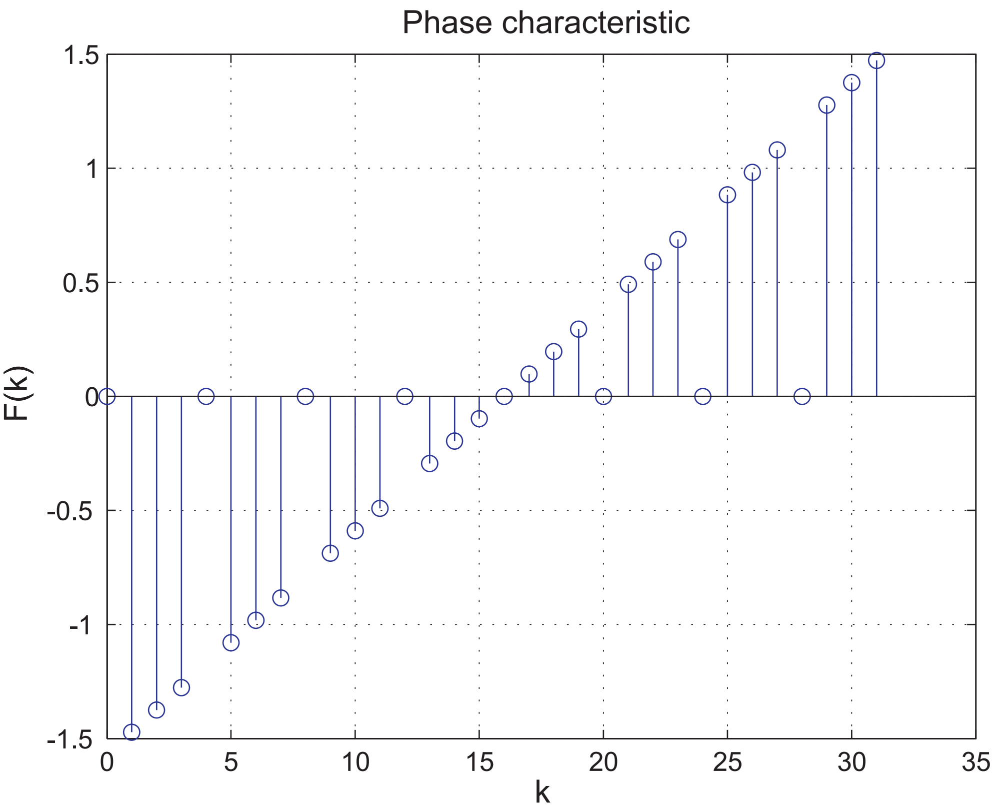
$$H(\Omega)=A(\Omega).e^{j\blacklozenge(\Omega)}$$

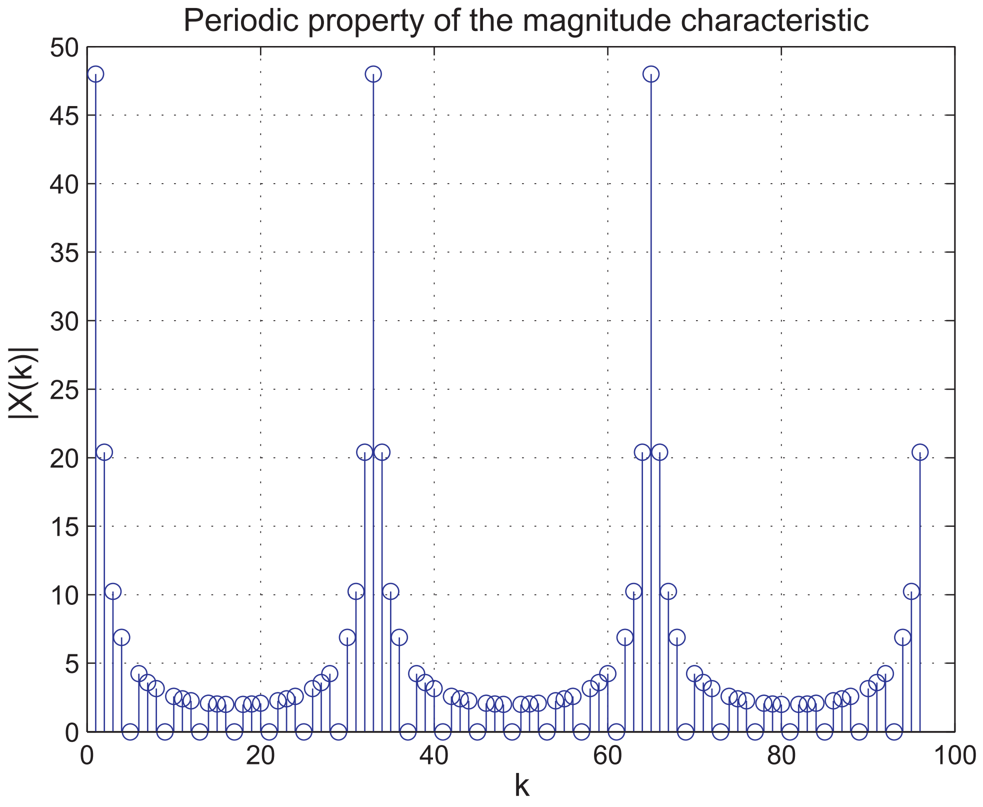
$$H(\Omega)=M(\Omega).e^{j\varphi(\Omega)}$$

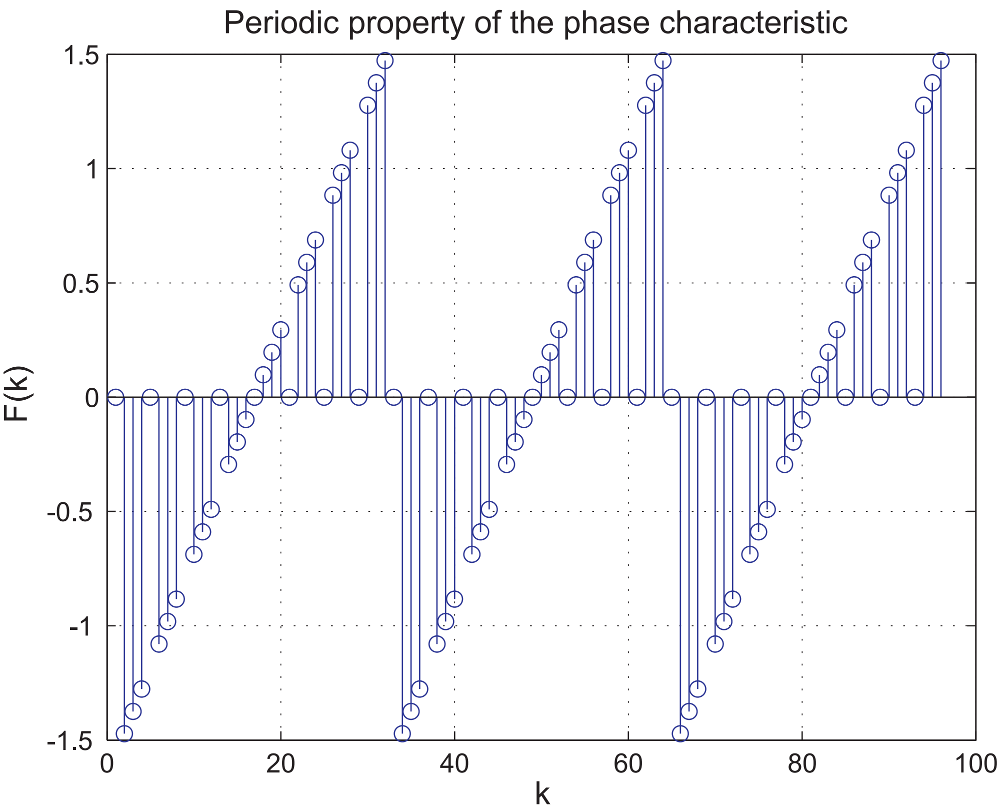


Examples of transfer function of LDTI, magnitude and phase frequency characteristic (one and more periods)




Phase characteristic
F(k)
k
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35


Periodic property of the magnitude characteristic
|X(k)|
k
50
45
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100


Periodic property of the phase characteristic
F(k)
k
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100

3.3 Filters

In signal processing, a filter is a device or process that removes from a signal some unwanted component or feature.

Most often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise. However, filters do not exclusively act in the frequency domain; especially in the field of image processing many other targets for filtering exist.

The drawback of filtering is the loss of information associated with it. Signal combination in Fourier space is an alternative approach for removal of certain frequencies from the recorded signal.

There are many different types of classifying filters and these overlap in many different ways; there is no simple hierarchical classification. Filters may be:

- analog or digital
- discrete-time (sampled) or continuous-time
- linear or non-linear
- infinite impulse response (IIR) or finite impulse response (FIR) type of discrete-time or digital filter.

Some terms used to describe and classify linear filters:

- Low-pass filter – low frequencies are passed, high frequencies are attenuated.
- High-pass filter – high frequencies are passed, low frequencies are attenuated.
- Band-pass filter – only frequencies in a frequency band are passed.
- Band-stop filter or band-reject filter – only frequencies in a frequency band are attenuated.
- Notch filter – rejects just one specific frequency – an extreme band-stop filter.

Filters can be built in a number of different technologies. The same transfer function can be realized in several different ways, that is the mathematical properties of the filter are the same but the physical properties are quite different. Often the components in different technologies are directly analogous to each other and fulfill the same role in their respective filters.

- Electronic filters were originally entirely passive consisting of resistance, inductance and capacitance. Active technology makes design easier and opens up new possibilities in filter specifications.
- Digital filters operate on signals represented in digital form. The essence of a digital filter is that it directly implements a mathematical algorithm,

corresponding to the desired filter transfer function, in its programming or microcode.

Linear analogue filters

Linear continuous-time circuit is perhaps the most common meaning for filter in the signal processing world, and simply "filter" is often taken to be synonymous. These are filters that are designed to remove certain frequencies and allow others to pass.

Such a filter is, of necessity, a linear filter. Any non-linearity will result in the output signal containing components of frequency which were not present in the input signal.

The modern design methodology for linear continuous-time filters is called network synthesis. Some important filter families designed in this way are:

- Chebyshev filter, has the best approximation to the ideal response of any filter for a specified order and ripple.
- Butterworth filter, has a maximally flat frequency response.
- Bessel filter, has a maximally flat phase delay.

The difference between these filter families is that they all use a different polynomial function to approximate to the ideal filter response. This results in each having a different transfer function.

Especially in the field of telecommunications, filters have been of crucial importance in a number of technological breakthroughs and have been the source of enormous profits for telecommunications companies. It should come as no surprise, therefore, that the early development of filters was intimately connected with transmission lines.

Digital filters

In electronics, computer science and mathematics, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal.

This is in contrast to the other major type of electronic filter, the analog filter, which is an electronic circuit operating on continuous-time analog signals. An analog signal may be processed by a digital filter by first being digitized and represented as a sequence of numbers, then manipulated mathematically, and then reconstructed as a new analog signal. In an analog filter, the input signal is directly manipulated by the circuit.

A digital filter is characterized by its transfer function, or equivalently, its difference equation. Mathematical analysis of the transfer function can describe how it will respond to any input.

Finite impulse response filter

Finite impulse response filter, called FIR filter – is a filter whose impulse response is of finite duration, because it settles to zero in finite time.

FIR filters have no feedback so output signal depends only on samples of input signal. In case we have N samples, $N-1$ is filter level.

Difference equation is defined as:

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

Transfer function is defined as:

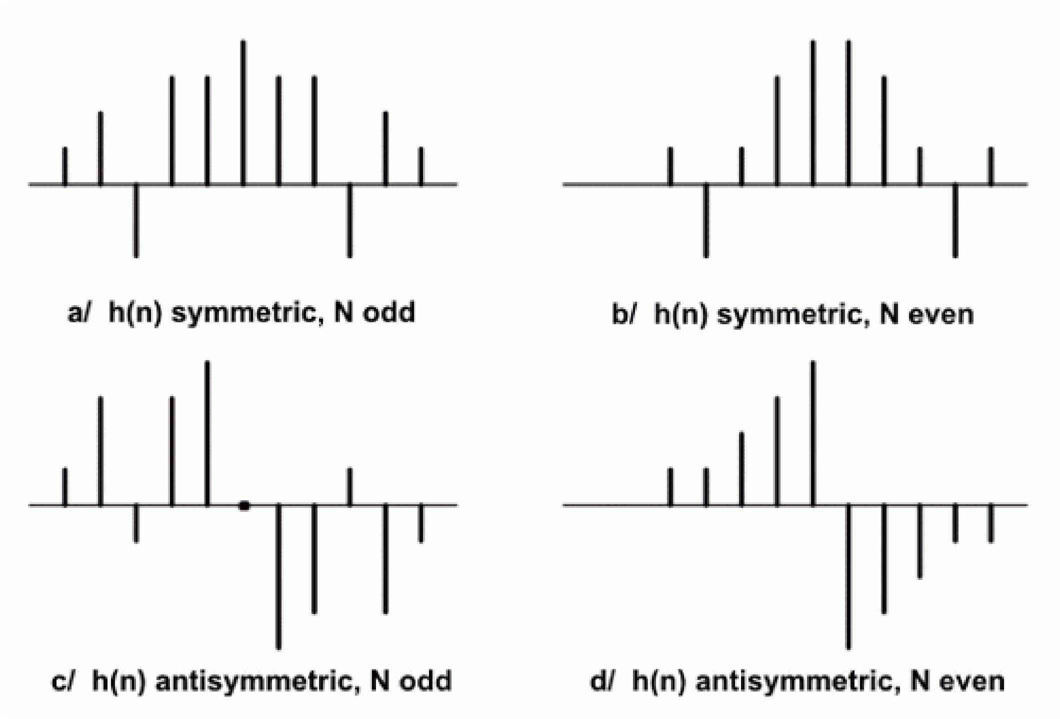
$$H(\Omega)=a_0+a_1e^{-j\Omega}+a_2e^{-j2\Omega}+\ldots+a_Ne^{-jN\Omega}$$

+

The main advantages of FIR filters are:

- Because there is no feedback, errors are no compounded by summed iterations.
- Simpler implementation.
- Stability. This is because there is no feedback so all poles* are located at the origin. The system is stable if the absolute value of each pole (poles are roots of denominators of filter transfer function) is less than one.
- It's easily designed to have linear phase. This is guaranteed by making impulse response symmetric or anti symmetric.

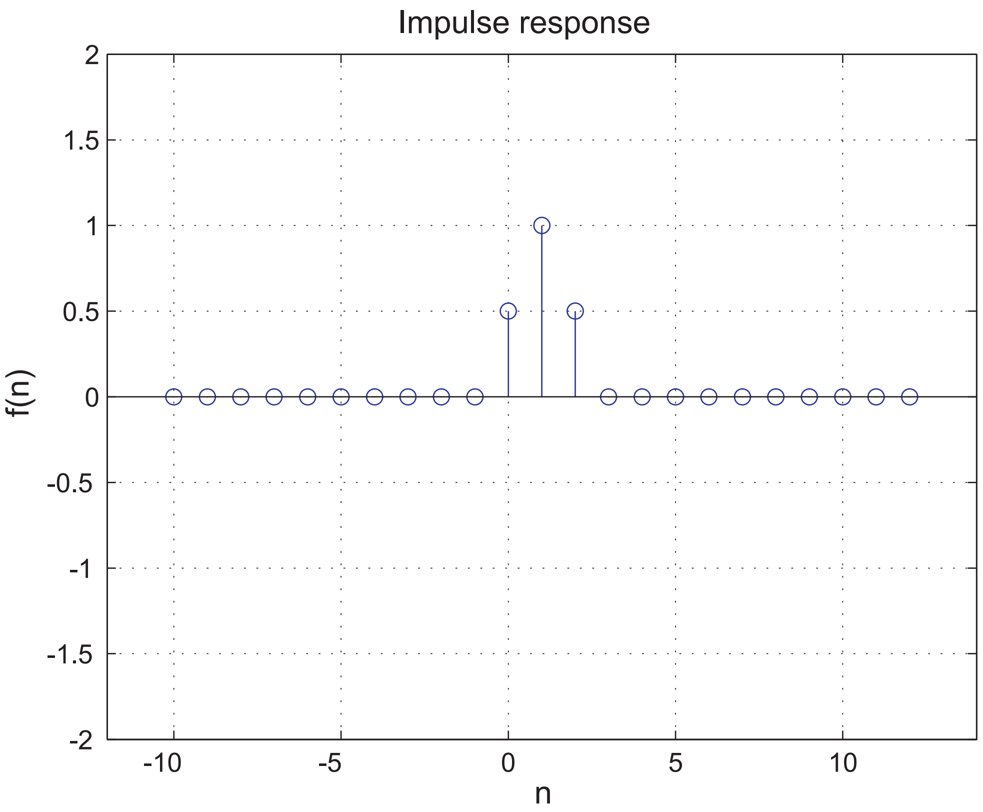
*Note: poles are roots of denominators of filter transfer function.


a/ h(n) symmetric, N odd
b/ h(n) symmetric, N even
c/ h(n) antisymmetric, N odd
d/ h(n) antisymmetric, N even

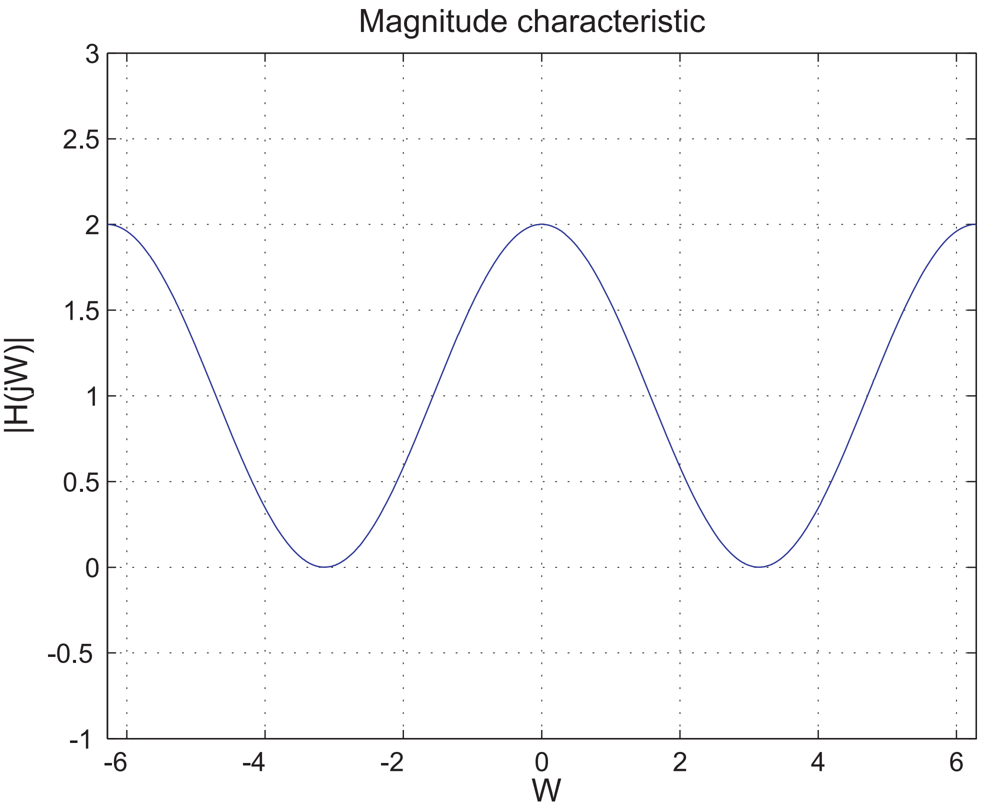
Impulse response (linear phase frequency characteristic)

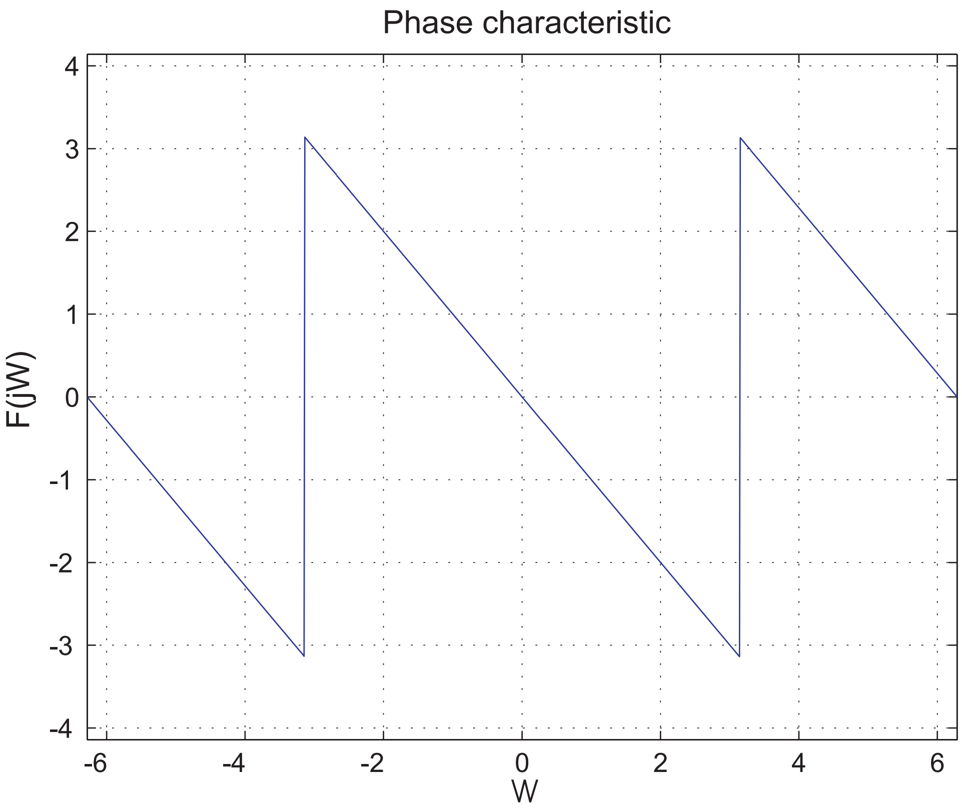
To design a filter means to select the coefficients such that the system has specific characteristics. The required characteristics are stated in filter specifications. Most of the time filter specifications refer to the frequency response of the filter. There are different methods to find the coefficients from frequency specifications:

- Intuitive method
- Frequency sampling method
- Window design method
- Weighted least square design



Impulse response, magnitude and phase frequency characteristic of FIR filter





Infinite impulse response filter

Infinite impulse response filter, called IIR filter – is a filter whose impulse response is function that is non-zero over an infinite length of time.

Filter output sample is given as a sum of N samples of the input signal weighted with a_k coefficients, and samples of output signal weighted with b_k coefficients. This is obvious also from difference equation defining IIR filter:

Difference equation defining IIR filter is:

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

and transfer function is defined as:

$$H(\Omega)=\frac{a_0+a_1e^{-j\Omega}+a_2e^{-j2\Omega}+\ldots+a_Ne^{-jN\Omega}}{1+b_0+b_1e^{-j\Omega}+b_2e^{-j2\Omega}+\ldots+b_Me^{-jM\Omega}}$$

Transfer function is fraction of two polynomials and so stability is not always guaranteed. As was mentioned before, the system is stable if the absolute value of each pole is less than one. There are several methods how to stabilize non stabile filter, as PLSI algorithm or using all-pass filter.

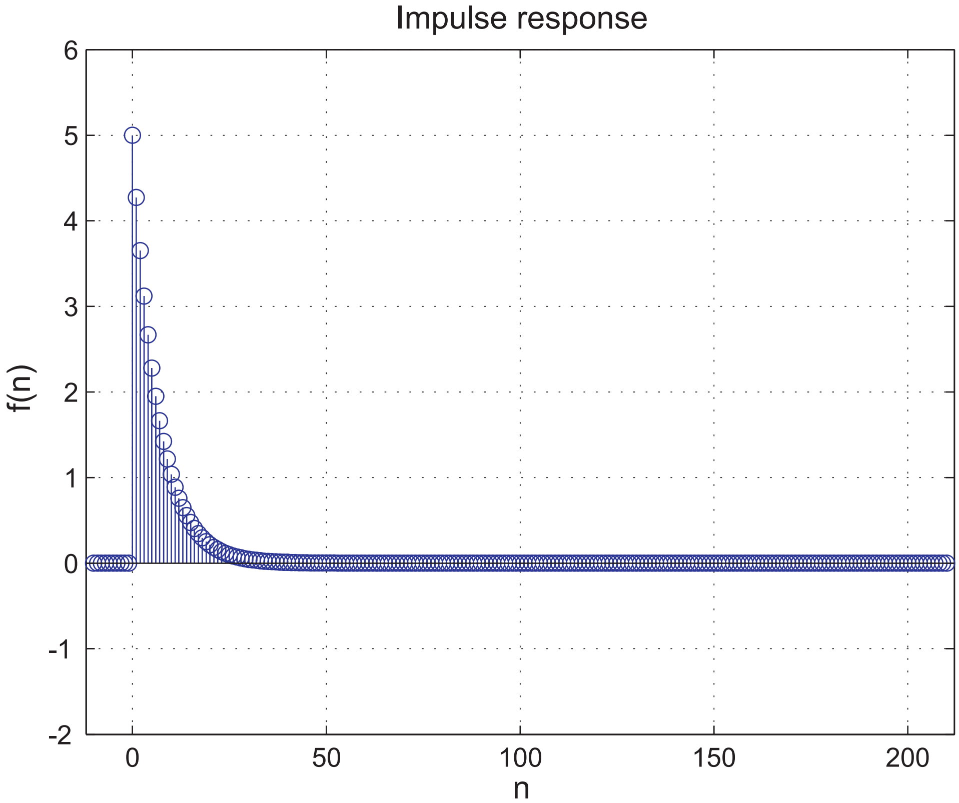
To design a filter means to select the coefficients such that the system has specific frequency characteristic. It means for IIR filters to determine level of numerator and denominators and coefficients a_k and b_k. The design methods of IIR filters can be divided into two groups.

Into first group belong straightforward methods:

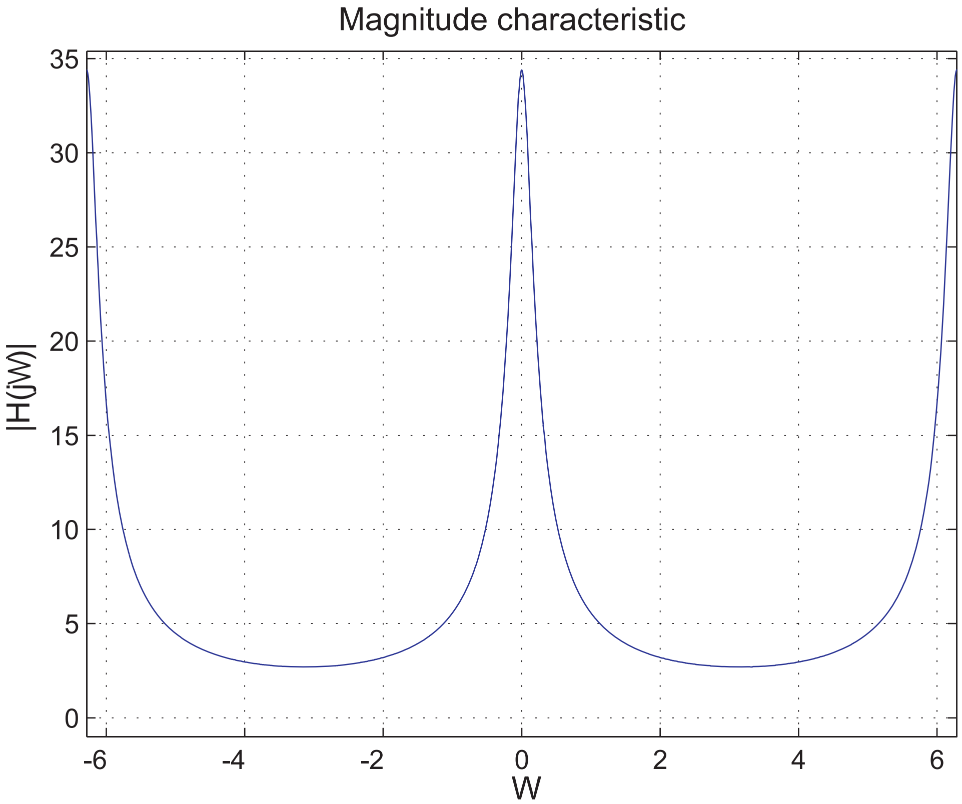
- Intuitive method
- Frequency sampling method
- Prony method

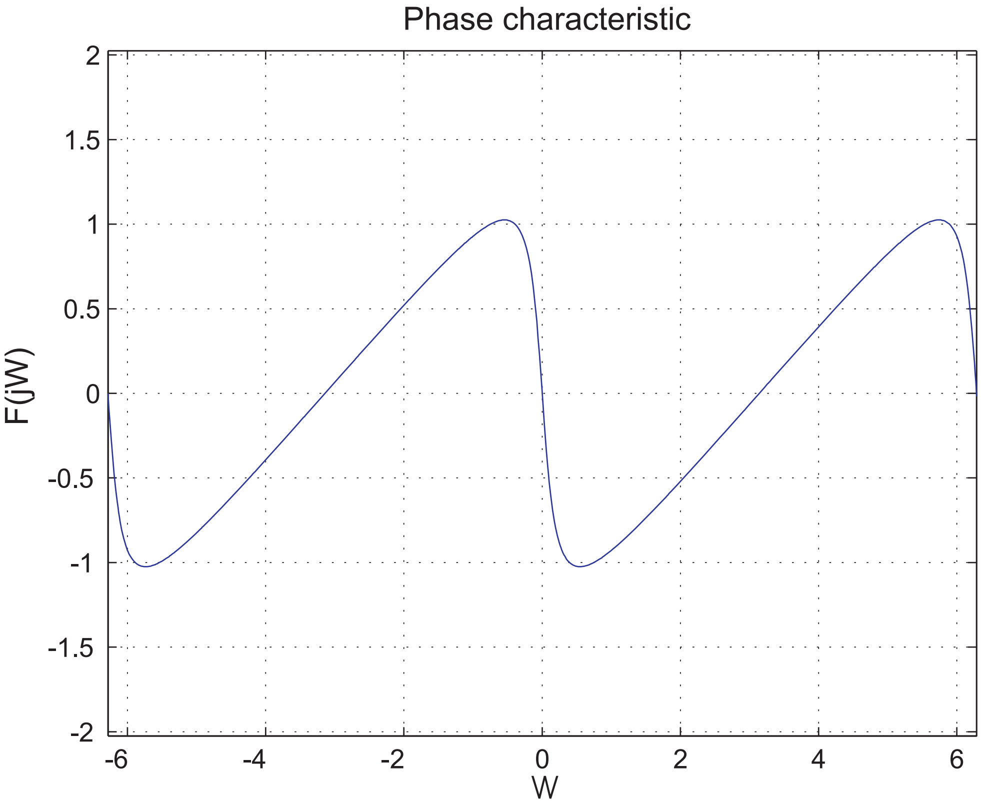
To the other group is based on analog filters design method. When a digital IIR filter is going to be implemented, an analog filter (e.g. Chebyshev filter, Butterworth filter, Elliptic filter) is first designed and then is converted to a digital filter by applying discretization techniques. Here belong methods as:

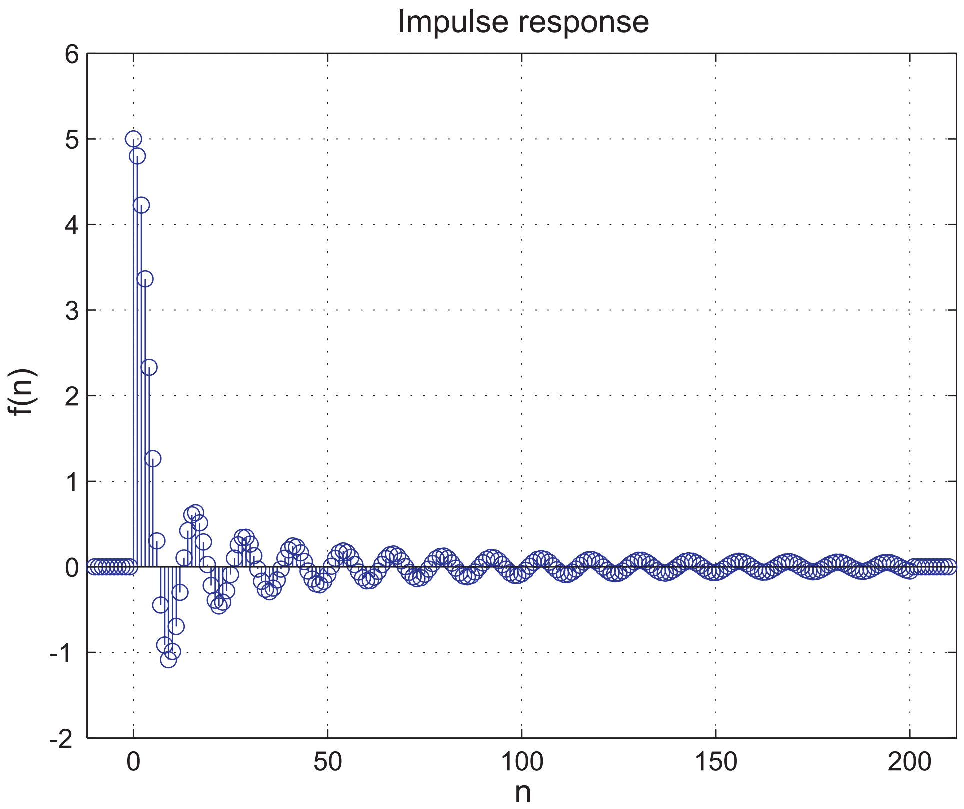
- Bilinear transform
- Impulse invariance.



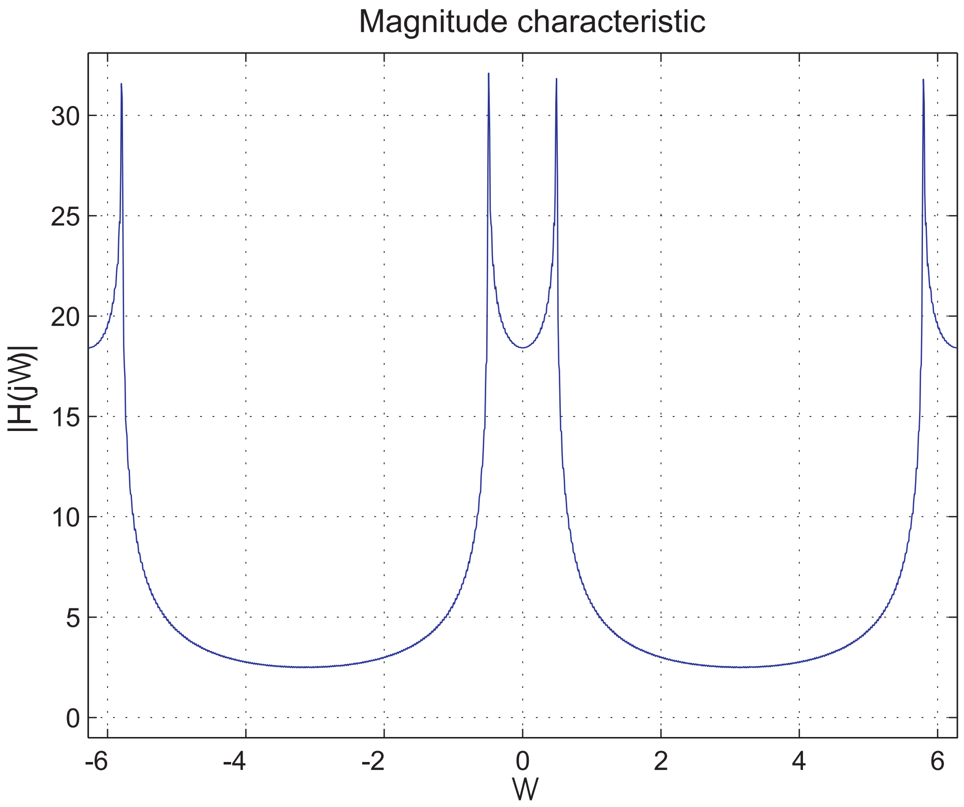
Impulse response, magnitude and phase frequency characteristic of IIR filter (exponential function)

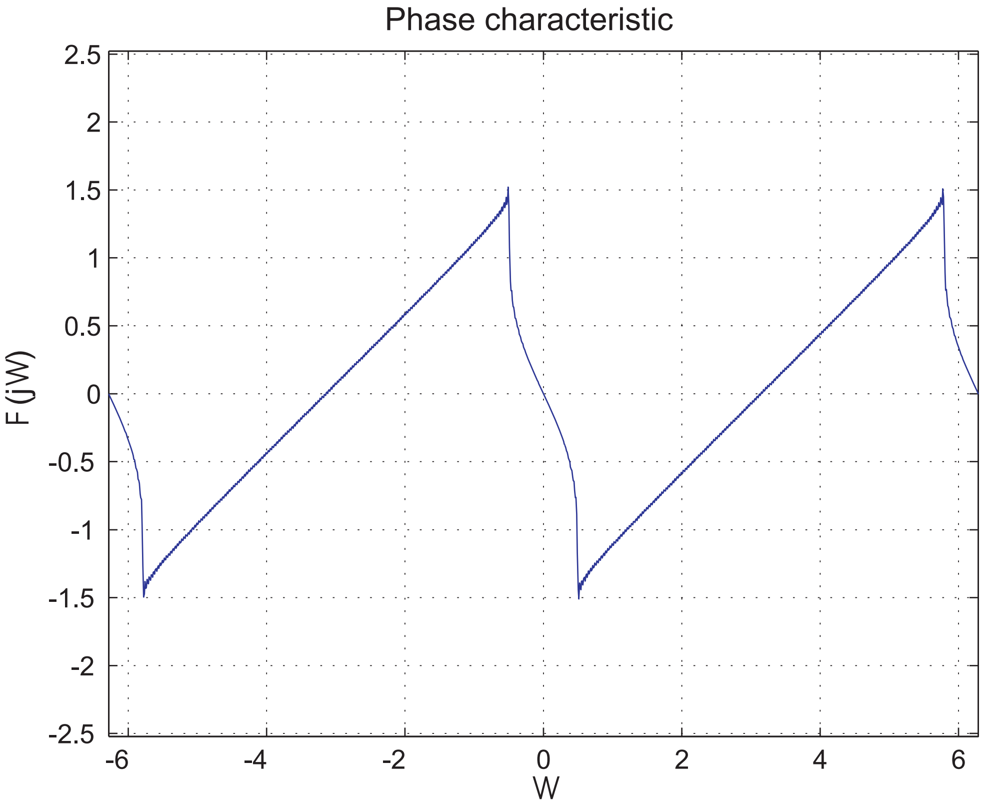

Magnitude characteristic
|H(jW)|
35
30
25
20
15
10
5
0
-6
-4
-2
0
2
4
6
W


Phase characteristic
F(jW)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-6
-4
-2
0
2
4
6
W



Impulse response, magnitude and phase frequency characteristic of IIR filter (sinc function)





+ The main advantage IIR filters have over FIR filters is that through recursion they use fewer taps. Therefore in digital signal processing applications IIR filters use fewer computing resources than an equivalent FIR filter.

– A disadvantage of IIR filters is they can be unstable. The implementation of IIR filters is more complicated than the implementation of FIR filters.

4 Communication Channel

4.1 Communication Channel

Typical communication system contains transmit system (called transmitter), receive system (called receiver) and transmission system, all together called telecommunication channel. Telecommunication channel then represents a set of technical devices allowing only one-way signal transmission between two points regardless of the used devices.

Telecommunication circuit is formed by a pair of channels allowing two-way communication. Communication can be either simplex (signal transmission circuit alternately in one direction or the other) or duplex (signal transmission circuit at the same time in both directions).

Based on the signal in channel two types of communication systems are known: analog and digital. In next chapters analog and digital transmission systems with basic function blocks will be introduced.

Analog communication channel

Analog communication is a data transmitting technique in a format that utilizes continuous signals to transmit data including speech, image, video, electrons etc.

Analog transmission is inexpensive and enables information to be transmitted from point-to-point or from one point to many. Data in analog signal is generally carried by use of modulation. Therefore it is necessary to have that transmitter and receiver have compatible devices – a suitable modulator on the transmitter side to the demodulator on the receiver side.

A device used to both to send and receive signal that contains both modulator and demodulator is called MODEM (*MOdulator+DEModulator*).

–

Analog circuits do not involve quantization of information unlike the digital circuits and consequently have a primary disadvantage of random variation and signal degradation, particularly resulting in adding noise to the audio or video quality over a distance. Data is represented by physical quantities that are added or removed to alter data.

Digital communication channel

Transmitter

Even digital communication system can work with analog signal. For this *analog-digital convertor* (**ADC**) is required and used. An analog signal is converted to the digital form in the analog-digital convertor. The detail process of signal digitalization is described in chapter Analog signal digitalization.

The next block working with digital signal is source encoder. The general description of source encoder is that it very effectively converts every number value into the digital representation (mainly the binary representation). Usually the coding process is already done in ADC. In the final digital form there is a big redundancy.

The main advantage of source encoder is reducing of the redundancy, it means less binary values (zeros and ones) to transmitting. Very simple example how to reduce redundancy: instead of “for example” we simply use “i.e.”

In many systems without a channel encoder is the output signal transmitted directly into communication channel. It is important to notice that output must be in correct form, one that is acceptable by the channel. As described in source encoding, one purpose of the source encoder is to eliminate redundant binary digits from the digitized signal. The strategy of the channel encoder, on the other hand, is to add intentional redundancy to the transmitted signal—in this case so that errors caused by noise during transmission can be detected and corrected at the receiver. This process is done by adding redundant information (we can say adding security information) to digital information of source encoder. The simplest forms of redundant information are repeating and parity.

Last device on transmitter side is modulator.

In digital modulation, an analog carrier signal is modulated by a discrete signal.

The type of modulation in use depends on channel and data that is transmitted. Some modulators are intended for noisy channels (such as radio, or WIFI), some are intended for clear channels (such as optical cable).

Receiver

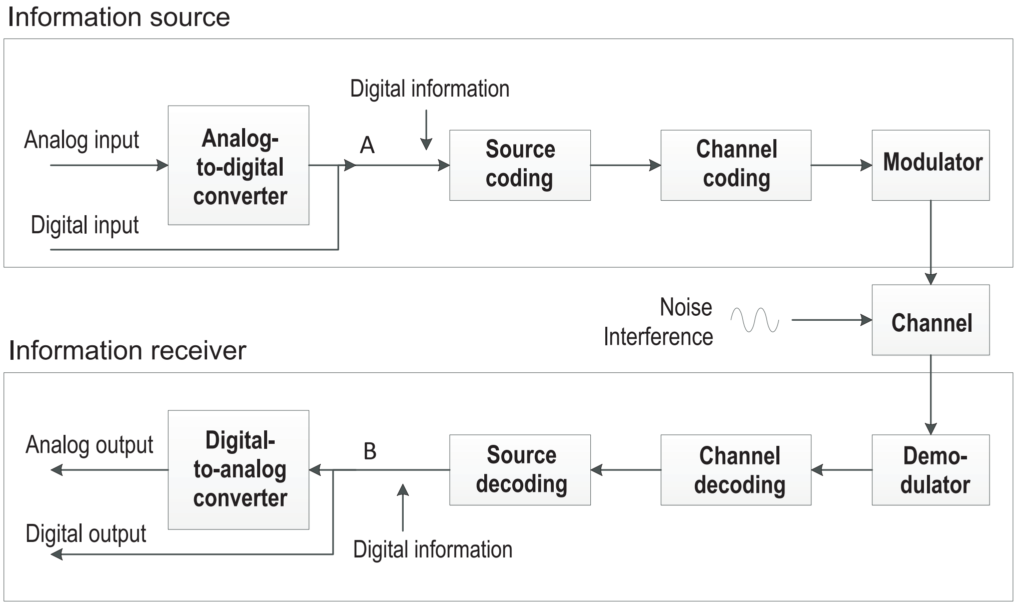
A receiver is basically the same device as transmitter in reverse order. Very important task in design and optimization of digital communication system is minimizing failures in transmission.

Since the signal is usually corrupted because of noise in the communication channel, it has to be repaired by the receiver. This can be done with a device called demodulator which turns a numerical value into actual signal (e.g. number 1 represents the value 5 of the signal; the value 5 can be amperes, volt, etc). There are several ways of demodulation depending on how parameters of the base-band signal are transmitted in the carrier signal, such as amplitude, frequency or phase.

For example, for a signal modulated with a linear modulation, like AM (Amplitude Modulation), we can use a synchronous detector. On the other hand, for a signal modulated with an angular modulation, we must use an FM (Frequency Modulation) demodulator or a PM (Phase Modulation) demodulator. Different kinds of circuits perform these functions.

The main task of channel decoder is to decode (or reconstruct) the output signal of channel encoder. In ideal condition the result is the same signal as was on the side of the transmitter. In real life there are always failures caused by noise in the communication channel. But because of the redundancy added by channel encoder the failures can be detected and corrected and we can get the accurate signal (which cannot be done using analog transmission).

The source decoder performs exactly an inverse function as source encoder. It means removed redundancy is added back to the signal. In case the output from receiver is digital, than the output from source decoder is output of the receiver. In other case, when an analog output is required, digital values are sent to *digital-analog convertor* (**DAC**). Digital-analog convertor converts the values to analog signal according the theory of sampling and quantization. The original signal is the result of these operations.



Block diagram of digital communication channel

5 Compression Techniques

The aim of compression is to remove irrelevant and redundant information from the original data in order to transmit or store it more effectively, e.g. use less bandwidth or storage space. It is done by encoding the information using fewer bits than are used in the original representation.

The amount of data which is created by today's multimedia devices is huge. If we wanted to store an uncompressed 6 megapixel image (3000x2000 pixels with 8 bits/colour) to a USB stick we would need more than 17 megabytes of space (exactly 18 million of bytes). The exactly same image compressed using the JPEG coder would only require around 3 megabytes of storage space (about 3 200 000 bytes), and you would not notice any difference. The same applies not only to pictures but also to audio and video signals, both for storage and for transmission across digital networks.

The compression algorithms can be divided into:

- Lossless coding
- Lossy coding

Lossless compression reduces bit by finding and reducing statistical redundancy, i. e. information which is repeating or can be determined from other information in the signal. Lossless compression (also known under the term *compaction*) allows us to reconstruct the original signal without any modification. It is therefore suitable for archival purposes.

Lossy compression, on the other hand, attempts to reduce the amount of data by eliminating information which cannot be perceived due to imperfections of human senses. When this type of compression is used, the reconstructed signal is never the same as the original. It is suitable for applications where limited bandwidth or storage occurs.

Generally, lossy algorithms are far more effective than lossless algorithms, however, the reconstructed signal is always different from the original one. Also, the quality of a compressed signal is the trade-off between the amount of information removed from the original signal and the desired file size of the compressed signal.

There is a measure of changes in the signal which are still acceptable without perceivable difference from the original signal, called the *just-noticeable distortion* (for speech and audio) or *just-noticeable difference* (for pictures and video), **jnd**.

There is a compression ratio to express the efficiency of a compression method, given by formula:

$$C_r = \frac{N_n}{N_k},$$

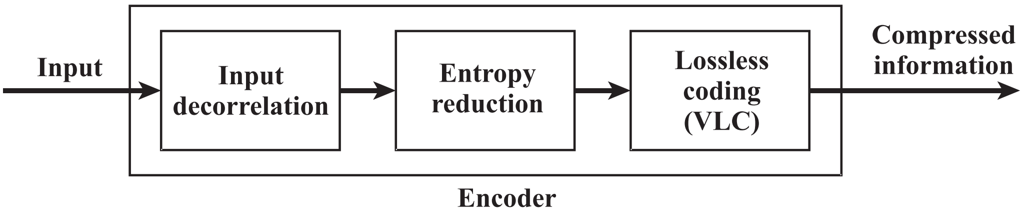
where N_n is the count of bits of an uncompressed signal and N_k is the count of bits of a compressed signal.

The software which performs coding of the original signal to the compressed version and decoding the compressed signal into the original signal's reconstruction is called the **codec** (a short for en**co**der-**dec**oder).

Compression model

There are 3 basic phases in the process of compression:

1. Input data decorrelation
2. Entropy reduction
3. Lossless coding



General compression scheme

Input data decorrelation removes duplicate information from the signal. It can be performed in various ways, or different domains, respectively:

- Time (or space) domain
- Parametric domain
- Signal decomposition

In time (or space) domain, the methods are based on linear prediction which assumes that the neighbouring samples in the time (or space) domain are correlated. The signal decomposition approach splits the signal into subbands allowing monitoring of the energy in each subband separately. The parametric domain applies a specific transform of the input data into the parametric space in order to extract characteristic parameters for coding and transport of the input data.

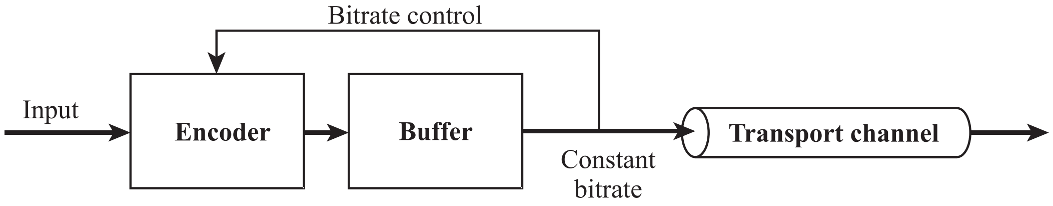
Entropy reduction is performed using quantization. Even though quantization always brings the loss of information sufficiently detailed quantization makes the loss irrelevant.

The final step in the compression process is lossless coding, or entropy coding where statistically frequent combinations of bits (symbols) are coded with a shorter code word and statistically less common bit combinations are coded with longer code words. This coding is also called the variable length coding (VLC), because the code words are of different length, and is lossless. The most efficient algorithm to perform VLC entropy coding is Huffman coding.

Two modes exist when transporting the compressed information through the transport channel:

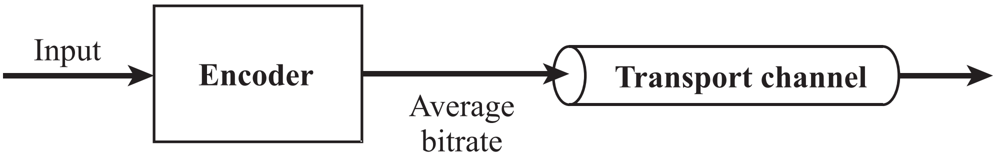
- Constant bitrate
- Variable bitrate

When using constant bitrate the coder output is stored in a buffer to maintain constant output bitrate into the transport channel. The bitrate influences the coder in a way that it produces the same bitrate. In this way, the coder can change the quantization steps and thus the quality of the reconstructed signal varies.



Encoder with constant bitrate output

When variable bitrate is used there is no need for the buffer and the coder may use quantization steps according to the necessities of the coded input signal. Quality of the reconstructed signal is constant.



Encoder with variable bitrate output

It may seem that variable bitrate is the best solution. This is only true if the transport channel's capacity is sufficient for the highest bitrates the variable bitrate coder may produce. On the other hand, if only limited capacity transport channel is available constant bitrate may be more suitable to maintain continuous data stream.

5.1 Compression of audio signals

When we compress a generic audio signal, there are numerous coding standards and compression approaches to choose from. Many of them focus on specific types of audio (i.e. speech) or parameters (computational complexity, delay, etc.).

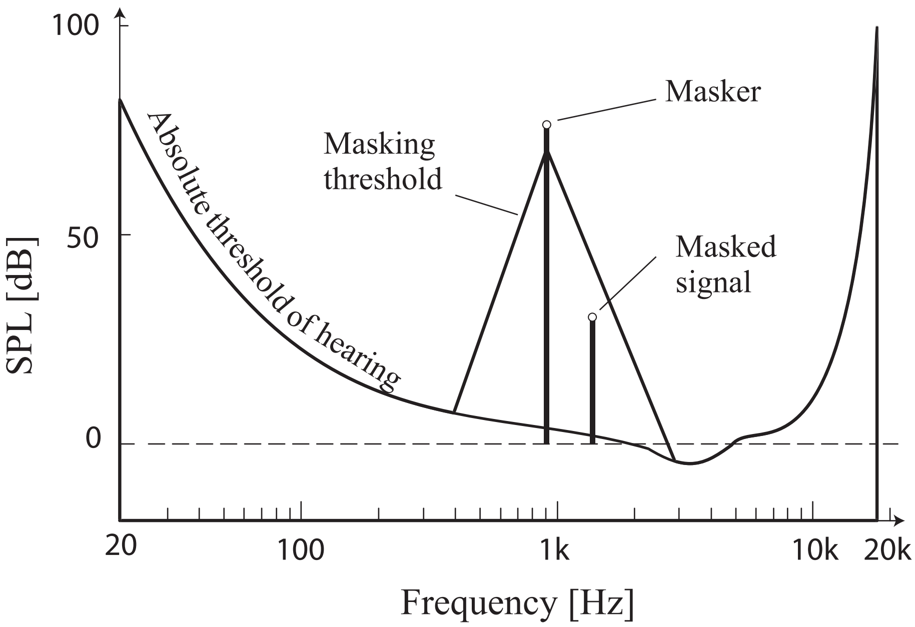
Sampling frequency describes how many signal samples are obtained every second. Generally, the higher the sampling frequency the higher the precision and quality of the recording can be obtained. The following are most used sampling frequencies: 8 kHz, 16 kHz, 22,5 kHz, 32 kHz, 44,1 kHz or 48 kHz for each audio channel.

Auditory masking

Auditory masking is a phenomenon observed as an imperfection in the human auditory system. The ears have limited possibility to hear all sounds, which is described by the absolute threshold of hearing. Additionally, loud sounds often cover more quiet sounds which occur close to them. This can happen in both time and frequency domain, dividing the masking into:

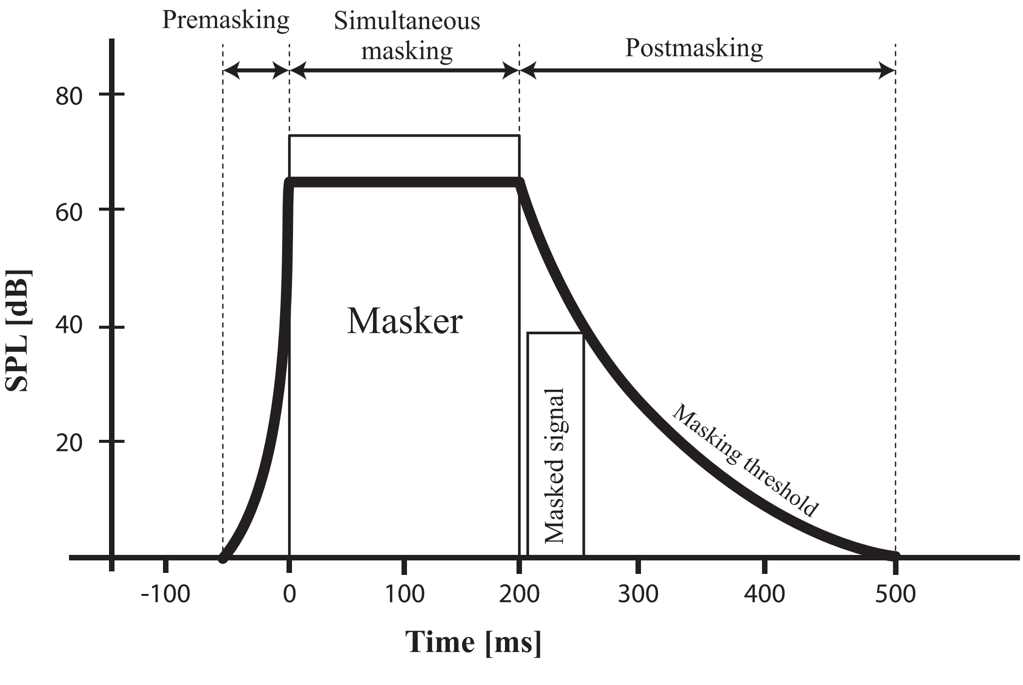
- temporal (non-simultaneous) masking
- frequency (simultaneous) masking

The loud sound is called the masker. When two sounds occur in the same time simultaneous masking may occur. The masker creates the masking threshold below which no other sounds could be heard. If the signal is close to the masker and falls below the threshold it will be masked. The following image shows how the masker can hide a silent signal in the frequency domain. Combination of maskers’ masking thresholds and the absolute threshold of hearing leads to creation of the global masking threshold which may change over time.



Masking in the frequency domain. If the masker's intensity is sufficiently higher than the signal's intensity and the two signals are close enough to each other the signal will not be audible and only the masker will be heard.

In the temporal, or non-simultaneous, masking the masker can mask a signal which occurs closely before (premasking) or after (postmasking) the masker. Again, the intensity of the masker must be higher than the signal's intensity.



Masking in the time domain

Frequency masking has been explored in fairly good detail and is widely used in plenty of audio codecs (as will be shown later on). Temporal masking, on the other hand, has not been examined with the same precision. This is mainly due to the duration of pre- and postmasking. Postmasking takes place up to 300 ms after the end of masker while premasking only lasts some 50 ms or less. These times are too short to be explored precisely because codecs usually work with 20 ms frames or larger, making use of only 2 or 3 frames for premasking.

MPEG

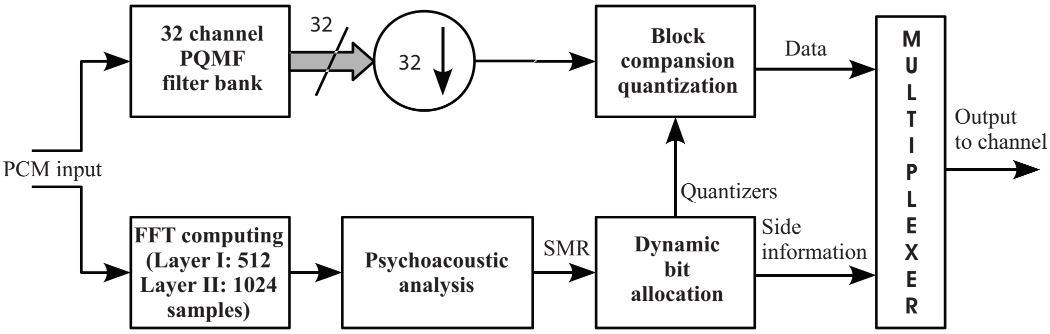
Currently, the most used audio codecs are based on the work of the *Motion Picture Experts Group* (**MPEG**) which is a part of the *International Standards Organization* (**ISO**). During its existence the group introduced several audio formats which gained worldwide usage.

As it will be obvious later on, these codecs are based on lossy coding which means they modify the original audio signal and the output is never the same as the original.

MPEG-1

The MPEG-1 standard represents a flexible coding technique, employing several methods, i.e. subband coding, filter bank analysis, transform coding, entropy coding and psychoacoustic analysis. It works with sampling frequencies of 32, 44.1 or 48 kHz with 16 bits/sample and the output bitrate varies from 32 up to 192 kbit/s per channel. The standard offers 4 channel modes, namely mono, stereo, dual mono and joint stereo (layer III only).

The standard's architecture contains 3 levels which differ in computational complexity, delay and output quality. Layers I (mp1) and II (mp2) are similar and only differ in several details. While both layers employ *fast Fourier transform* (**FFT**), the window size is 512 samples for layer I and 1024 samples for layer II. Maximum subband quantization's resolution is 15 bits/sample in layer I and 16 bits/sample in layer II. Even though these differences seem minimal it has been shown that layer II provides same or even higher quality output at 128 kbit/s than layer I at 192 kbit/s per audio channel.



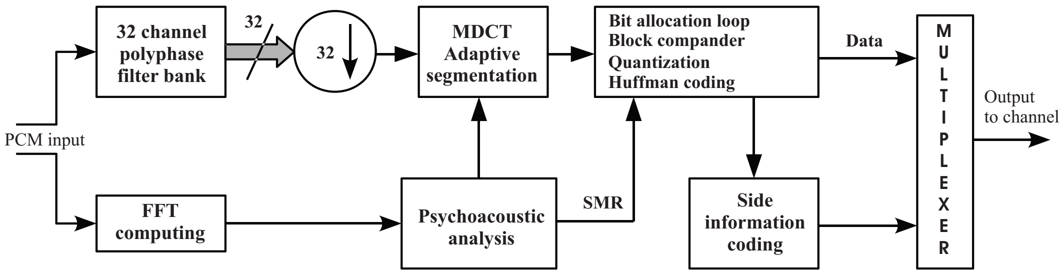
General scheme of the MPEG-1 layer I and II encoder

The compression in both layers I and II works with a PCM input signal which is divided into 32 subbands. During division, FFT is performed in order to perform psychoacoustic analysis and determine the jnd. Based on the masking threshold of each subband suitable quantization steps are decided so that required bitrate and masking level is maintained. The output is then coded using Huffman's entropy coding.

Although MPEG-1 layer II provides acceptable results, the still ruling format is the *MPEG-1 layer III*, commonly known by its shortcut **mp3**. It is based on layers I and II, however, it adds a few techniques which lead to lower bitrate (64 kbit/s per channel) while maintaining the same quality as its predecessors.

The algorithm takes 1152 samples and divides them into 2 granules (576 samples each). Each of the granules passes through the hybrid filter bank (a set of band-pass filters used to split the input into subbands: each subband may then be processed individually) in order to improve frequency resolution. Each subband is then transformed into frequency spectrum using *Modified discreet cosine transform* (**MDCT**). Then, bit allocation and quantization are performed iteratively, when, during each iteration, a process of analysis-by-synthesis is performed to determine the quantization noise level.

Modified discreet cosine transform (MDCT) is derived from the *discreet Fourier transform* (**DFT**) but is specifically designed for signals with overlapping blocks of samples. It decomposes, or transforms, the input signal into a set of cosine functions. Compared to the Fourier transform whose output is a set of complex numbers, the MDCT output is set of real numbers representing the cosine functions. Moreover, DFT outputs the same number of coefficients as was the number of input samples, while MDCT, due to its overlap feature, outputs half the number of coefficients than there are input samples.



General scheme of the MPEG-1 layer III encoder

There are 2 extensions to the original layer III format, namely the **MP3pro** and **mp3 surround**. MP3pro adds a technique called *Spectral Band Replication* (**SBR**) which is used in lower bitrates to remove the original higher frequencies. They are then reconstructed from the compressed signal using side information. mp3 surround allows to compress a 5.1 channel audio (five full range channels and one low frequency - bass - channel) into mp3's 2 channels from which the 5.1 channels can be reconstructed using side information added to the file. Side information is ignored by a non-supporting decoder and the file is played back as an ordinary mp3 file.

MPEG-2

The MPEG-2 is a formal successor of MPEG-1. It comprises 2 modes, one being Backward Compatible with MPEG-1 (MPEG-2 BC) while the other, MPEG-2 Non-Backward Compatible (MPEG-2 NBC) leaves backward compatibility in favour of new methods and coding techniques.

The MPEG-2 BC only brings support for lower sampling frequencies (LSF) and multi-channel coding and is quite similar to mp3 surround. The MPEG-2 NBC is also known as Advanced Audio Coding (AAC) and is created as a set of tools for effective coding. The more tools are used the better compression is achieved while keeping the same quality, at the cost of higher complexity and delay. Unlike MPEG-1, it no longer uses hybrid filter bank to analyse signals, only MDCT is used and transform function uses different window functions. The MPEG-2 AAC became part of the MPEG-4 family of standards.

MPEG-4 AAC

The MPEG-4 AAC attempts to conquer the rule of mp3 format. It provides support for sampling frequencies from 8 up to 96 kHz, 1 to 48 audio channels plus 15 bass channels and additional 15 data channels with 8, 16, 24 or 32 bits per sample. The Low Complexity (LC) AAC represents the original MPEG-2 AAC codec and is suitable for speech coding at 8-12 kbit/s. The High Efficiency (HE) AAC adds the SBR technology (v1) and parametric stereo (v2) which is based on joint stereo profile of MPEG-1 layer III.

Ogg Vorbis

The Vorbis audio codec is one of the most successful open source codecs. Since 2000 when it was standardized it became the direct competitor to MPEG's mp3. It supports sampling frequencies from 8 up to 192 kHz, up to 255 channels and its output bitrate is by default variable.

Its coding process is different from MPEG: first, the signal is transformed using MDCT. Then, a so-called floor is calculated as a rough approximation of the spectral envelope (a curve which connects all amplitude bins in the frequency spectrum) using split linear function. The difference between the spectrum and the floor is then coded using multi-pass vector quantization.

Ogg Vorbis has higher memory requirements than mp3 because its header contains also entropic code table (mp3 uses fixed table) and other settings for the decoder. Nevertheless, it is highly suitable for compression of generic audio signals and provides similar or higher audio quality at the same bitrate as the mp3 codec.

Windows Media Audio

Windows Media Audio (WMA) is a proprietary codec created by Microsoft in response to mp3's licensing requirements. There are several variants of the codec:

the WMA 9 is a direct mp3 competitor with support for sampling frequencies up to 48 kHz with 16 bits per sample and output bitrates ranging from 64 to 192 kbit/s, supporting both CBR and VBR.

The WMA 10 Professional extends the codec's possibilities to compete with MPEG-4 AAC by adding sampling frequency of 96 kHz with 24 bits per sample for 7.1 channels. If the device is not capable of 7.1 playback the signal is automatically degraded to parameters (sampling frequency, bits per sample and channel downmix) suitable for the device. This suggests that the codec utilizes a technique similar to that used in mp3 surround.

The WMA 10 also provides modes for speech compression called WMA 10 Voice, which has output bitrate from 4 up to 20 kbit/s. Its speciality, however, is the ability to dynamically switch between the voice and standard codec if the audio signal is too complex. Additionally, WMA 10 provides a Lossless version which is claimed to be able to reduce the file size of the original PCM signal to half or even third of its size.

The WMA 10 Professional codec provides higher quality at 64 kbit/s when compared to MPEG-4 AAC v2 in 70% of comparisons.

5.2 Compression of speech

Even though speech is in its nature audio signal it has some specific characteristics which allow us to use more radical compression techniques than with generic audio signal. Firstly, speech signal is meant as a medium to express information. The information doesn't have to be articulated in the exact same manner as the original in order to be understandable. This implies audio characteristics can be reduced. For example, standard phone call is sampled with sampling frequency of 8 kHz (compared to 44.1 kHz with standard audio sampling frequency), which means only 4 kHz bandwidth can be acquired. The bandwidth contains most of the speech's energy, and information.

Secondly, speech signal is relatively simple compared to a recording of a rock band, when usually there is only one speaker at a time and no other instruments. Moreover, to obtain the clearest speech we apply algorithms to suppress background noise.

Techniques used in speech compression can be divided into following groups:

- Waveform coding
    - Time domain
    - Frequency domain
- Vocoders
    - Linear predictive coding
    - Formant coding

For the purposes of quality assessment of the various speech processing algorithms a measure exists that is called intelligibility. It describes how comprehensible and understandable the speech is. There are various aspects of speech that are taken into account, for example speech level, non-linear distortions, background noise level, echoes and reverberations and others. There are two main scales for the measure: the Speech Transmission Index (STI) and Common Intelligibility Scale (CIS), both ranging from 0 (worst) to 1 (best), or 0% to 100%. Generally, it is desirable to achieve at least 0,5 (or 50%) for the speech to be understandable.

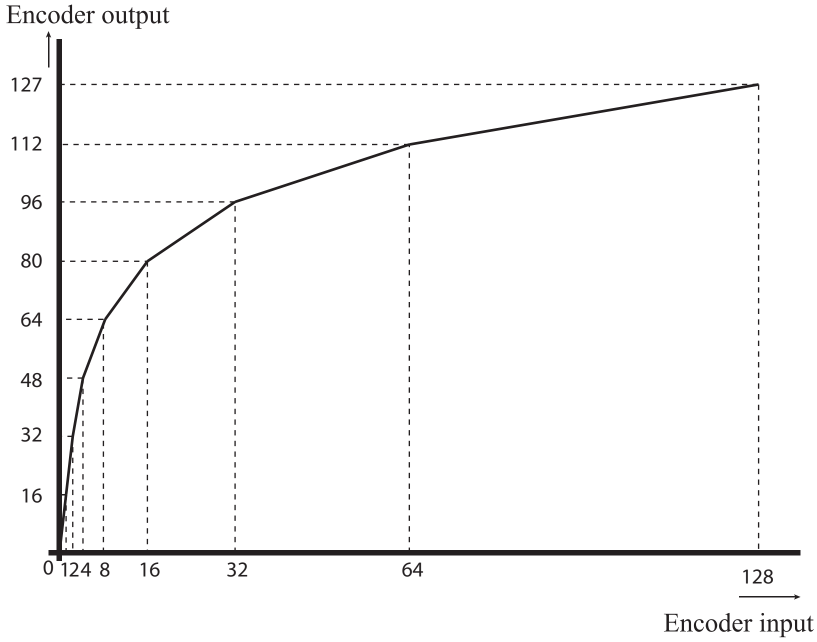
Time Domain

Waveform coding in the time domain is represented by the PCM technique. While linear PCM uses equal distances between quantization levels, non-linear PCM uses non-linear quantization steps, or, as a modification, the dynamics of the input signal is compressed (companded) by the transmitter and expanded by the receiver.

There are two compansion characteristics defined in the G.711 recommendation, the µ-law (USA and Japan) and the A-law (Europe). For example, the A-law characteristic is given by:

$$F(x) = \mathrm{sgn}(x)\begin{cases} \dfrac{A\left|x\right|}{1+\ln\left(A\right)}, & \left|x\right| < \dfrac{1}{A} \\ \dfrac{1+\ln\left(A\left|x\right|\right)}{1+\ln\left(A\right)}, & \dfrac{1}{A} \le \left|x\right| \le 1 \end{cases}$$

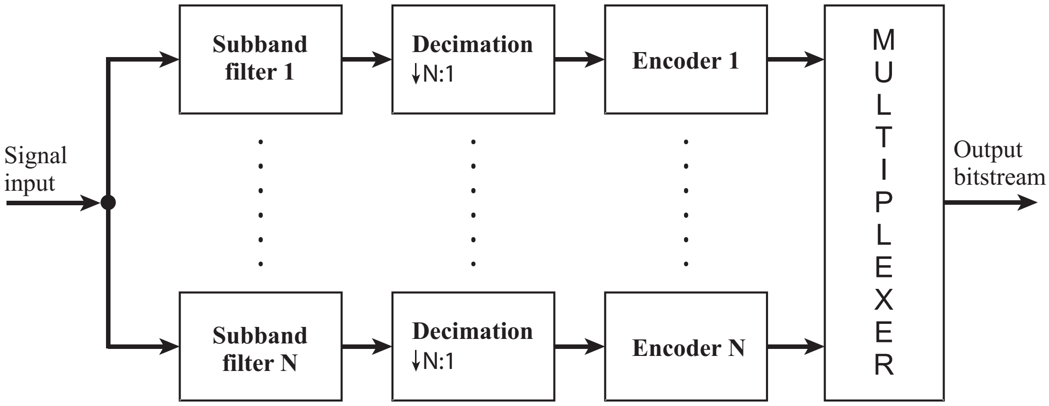
where sgn(x) is ±1 for positive or negative value of x and A is the compression parameter. Usually, A=87,7.



An example of the A-law compansion curve. Note that higher frequencies (represented on the horizontal axis with higher numbers) are encoded with less values than lower frequencies.

Frequency Domain

In the frequency domain, subband coding and adaptive transform coding are used. In Subband Coding (SBC), the speech signal is split into several frequency bands using a set of band frequency filters (filter bank) and the signal is decimated to reduce the number of samples. Then, each subband is coded, most often using the ADPCM method which allows flexible quantization and bit assignment.

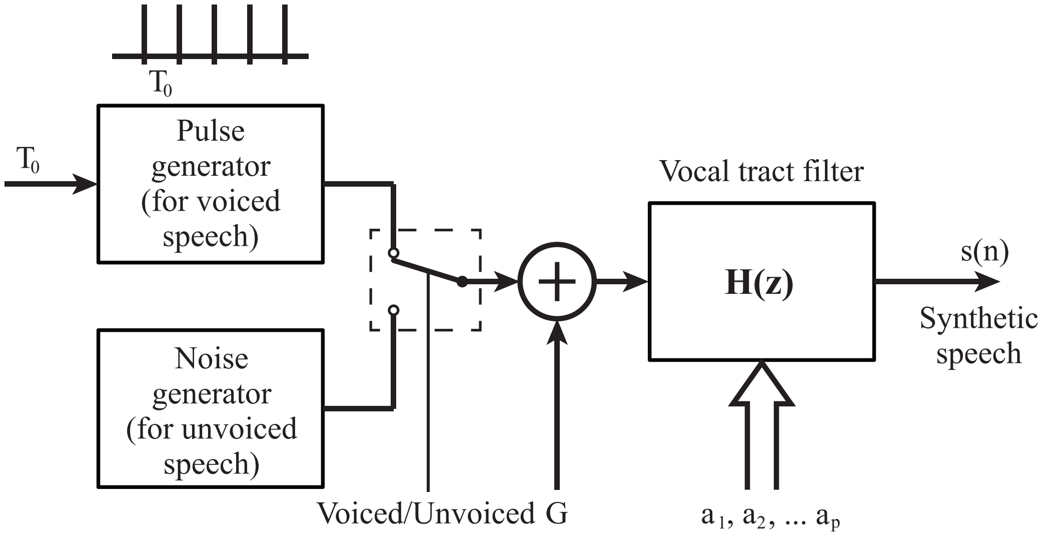


An example of the filter bank

Alternatively to ADPCM, other methods based on *Adaptive Transform Coding* (**ATC**) may be used. In these, the signal is transformed into frequency domain by applying FFT or other transformation and split into subbands. Then, bits are assigned dynamically to the samples in the subbands according to each band's need.

Linear Predictive Coding

Natural human speech can be understood as a response of the vocal tract of the speaker to excitation signal, in our case the air exhaled from lungs. The output signal is then modelled by changing the properties of the vocal tract (vocal cord, oral cavity, teeth, etc.). If we looked at the process from the signal analysis point of view, we could represent the output signal using the excitation signal and a filter representing the vocal tract, with time-varying parameters, the coefficients which are re-calculated from frames of 10 to 30 ms. Even though there are various methods to describe the coefficients of the vocal tract function the most used method is based on linear prediction, hence the name *Linear Prediction Coding* (**LPC**).



General scheme of the LPC decoder

The LPC coefficients minimize the quadratic variation between original and predicted speech samples. As it can be seen, the model of the LPC speech generator consists of two parts, the same as mentioned before:

- Vocal tract excitation
- Vocal tract filter

The vocal tract excitation is represented by the pulse generator and noise generator, which are switchable depending on voice of the piece of speech. The excitation is further amplified by gain (G) to the required level.

The vocal chords vibrate with a specific base frequency f_0 which leads to the base period T_0. The higher the frequency the higher the pitch of speech.

Depending on the use of vocal chords, the speech sounds can be divided into:

- **voiced** – sounds are pronounced using the vibrations of the vocal chords, i.e. ‘a’, ‘v’, ‘z’. All vowels are voiced.
- **unvoiced** – sounds are pronounced only using the noise-like flow of air with no vocal chord vibrations, i.e. ‘s’, ‘c’, ‘f’.

The vocal tract filter is given by a linear digital filter with finite response (FIR filter) whose transfer function is given by:

$$H(z)=\frac{G}{1+\sum_{i=1}^{p} a_i \cdot z^{-1}}=\frac{S(z)}{E(z)} \ ,$$

where a_i are the filter coefficients and p is the order of the filter. When $S(z)$ represents the output sample and $E(z)$ is the excitation, we get the next sample $s(n)$ as a linear combination of the previous samples with the excitation $G.e(n)$:

$$s(n)=G\cdot e(n)-\sum_{i=1}^{p}a_i\cdot z^{-1}=G\cdot e(n)-a_1\cdot s(n-1)-\ldots-a_p\cdot s(n-p)$$

To be able to use the LPC speech generator the following information has to be determined for each frame:

- Segment voice
- Base period T_0
- Filter parameters (the G amplification and a_i coefficients)

The bitrate of the LPC encoded speech signal varies from 1.2 to 2.4 kbit/s and its intelligibility is around 80-85%.

However, the reconstructed signal sounds machine-like, which is given by the two main factors:

1. It is difficult to segment the speech exactly to voiced and unvoiced frames as there are more types and combinations of the two in natural speech.
2. The base period in natural speech (which is the characteristic of the speaker's voice) changes more often than is the frame length and the change is not periodic.

There are methods which suppress the imperfections of the LPC method by coding the residue between the original signal and the LPC-predicted one.

The *Residually Excited Linear Prediction* (**RELP**) transfers the difference between the original signal and the LPC-reconstructed signal and transfers the residuum directly. At the receiver side the LPC coefficients are used to generate the reconstruction and then the residuum is added to form more precise reconstruction.

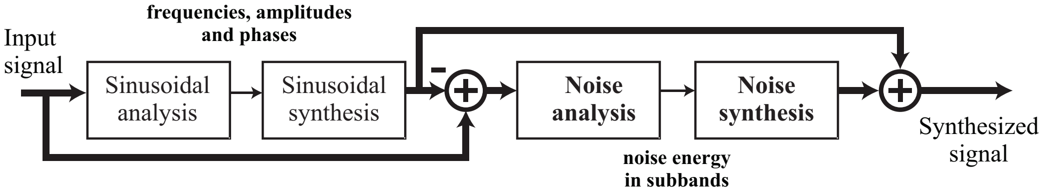
A successor to RELP algorithm is the *Code Excited Linear Prediction* (**CELP**). The algorithm is based on the analysis-by-synthesis principle and performs perceptual optimization of the synthesis signal in a closed loop. Then, a fixed codebook is searched for the most suitable excitation function and only the position in the codebook is transmitted along the LPC coefficients. Alternatively, the excitation function can be encoded using vector quantization.

The method achieves bitrates from 4 to 8 kbit/s. Its disadvantages are relative computational demands and delay at around 35 ms.

The CELP's modification Low Delay CELP (LD-CELP) reduces the delay to 2 ms while using 16 kbit/s bitrate. It is a part of the ITU-T's G.728 standard. Another codec based on CELP technique is Speex, an open source codec from organization Xiph.Org, the author of the Ogg Vorbis.

Sinusoidal coding

Sinusoidal coding derives from the theory that any audio signal is a combination of a deterministic and stochastic signal. The deterministic part can be therefore represented by harmonic functions (sines, cosines) while stochastic part can be modelled by noise or other parameterization. The principal scheme of such coder is given below. Sinusoids are time-changing frequency connections which are believed to form one tone.



General scheme of the sinusoidal coder

This model, however, cannot sufficiently model quickly changing parts of the sound so a third part has been added, the transients which model fast changes in the signal. This leads to the sinusoids+transients+noise (STN) model.

Another extension to the basic SN model is Harmonic + Individual Lines + Noise (HILN) model. In this approach, the sinusoids are split into two groups, harmonic part and individual part. In the harmonic part, the sinusoids are represented as harmonic multiples of the basic frequencies and only the multiples are stored. Then, individual sinusoids are encoded and the residuum is treated as noise.

+ Sinusoidal coding is expected to deal well with simple signals which consist mainly of harmonic sounds, such as speech. Based on the technique was SKYPE's first codec SVOPC which achieved good quality at 20 kbit/s and was robust against packet loss.

– However, its computational demands led to creation of a new, LPC-based SKYPE codec named SILK.

On the basis of the SILK codec, and combined with properties of the *Constrained Energy Lapped Transform* (**CELT**) codec, a new codec has been standardized in September 2012, the Opus codec. The codec is able to utilize SILK's good performance at low frequencies and low delay of the CELT codec at higher frequencies, and switch between the two on request. The codec is highly capable of encoding both speech and audio and is suitable for online applications such as VoIP and live broadcast.

5.3 Compression of static image

The objective of image compression is to reduce the unnecessary information in the image to reduce the required bandwidth or storage space. As with audio signals, there are both lossless and lossy algorithms available depending on the application an image is meant for.

Note that when analysing image, all of the transformations are considered 2-dimensional by default.

In order to understand the way image is stored in a computer, various colour spaces have been defined. Let's mention the most used ones.

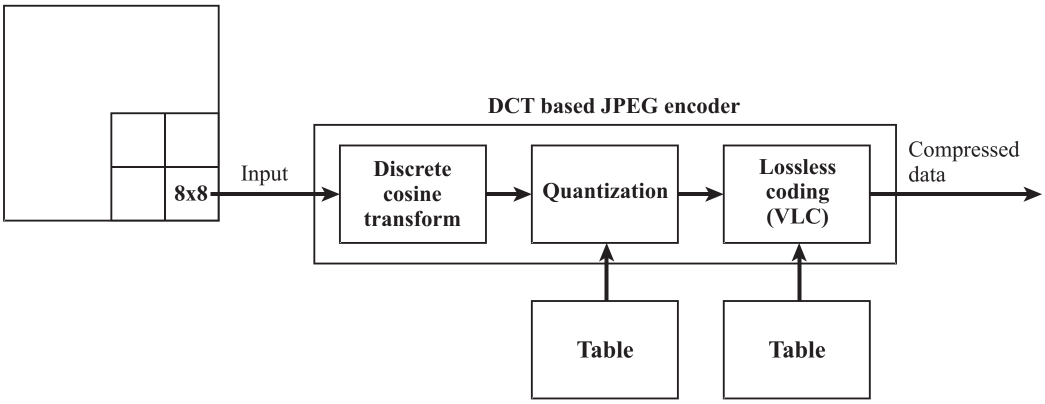
Selected colour spaces

Abbreviation	**Meaning**	**Explanation**
RGB	Red, Green, Blue	Each pixel is given by the combination of the 3 colours of light. Combination of the highest levels of all 3 colours gives white. It is used in light imaging.
RGBA	Red, Green, Blue, Alpha	Applies the same explanation as RGB. The additional Alpha channel describes transparency.
YC_BC_R or YUV	Brightness, Blue chrominance (U), Red chrominance (V)	Brightness scales from black to white. Blue and Red chrominance are calculated from the given RGB source. Although differences in calculation exist between them, YC_BC_R is also referred to as YUV.
CMYK	Cyan, Magenta, Yellow, Black	Each point is given by the combination of the 4 colours. Combination of the highest levels of all 4 colours gives black. It is used in print.

JPEG

One of the most commonly used lossy image formats for acquiring photographs is JPEG. It is named after the Joint Picture Experts Group which created the standard in 1986. It achieves the compression of 10:1 with little perceptible loss in quality.

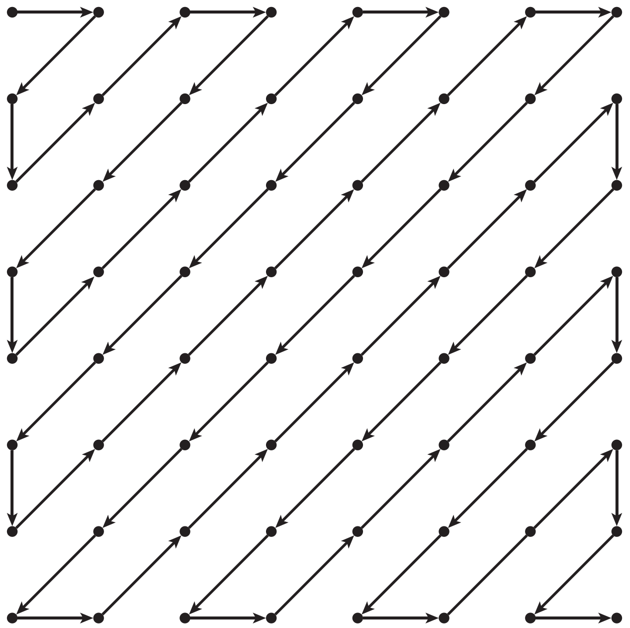
The JPEG algorithm is based on the 2-dimensional Discreet Cosine Transform (DCT). The input image is converted into the YC_BC_R colour space which provides better properties than say RGB colour space. The image is then split into non-overlapping blocks of 8x8 which are transformed using DCT. The obtained coefficients are quantized and insignificant coefficients are removed which is where the lossy compression occurs. The coefficients are then collated into 1-dimensional sequence and lossless coded. Image components (Y, C_B, and CR) are encoded in turn.



General scheme of the JPEG encoder.
The input is an 8x8 block of luminance or chrominance pixels.

The quantization is the key to compression in JPEG algorithm. The quantization is non-linear as human eye is more sensible to changes in lower frequencies. In order to bring scalability to the quality/compression ratio a quality factor q_f is defined, ranging from 1 to 100, which modifies the quantization matrix.

To collate the coefficients, a “zig-zag” reading is used, starting in the upper left corner. If each 8x8 block is encoded together the encoding is called **baseline JPEG**. Other approach is to encode all of the upper left corners of each 8x8 block in a sequence and continue to the next position in each of the blocks. Such approach is called the **progressive JPEG** and has the advantage of gradual image reconstruction during its download. Additionally, JPEG offers **hierarchical** mode where the image is encoded in a layered pyramidal way. Each upper layer’s pixel is acquired by applying certain operation to the 2x2 pixels laying in the layer right below. Each layer is individually separable by the decoder, allowing for multi-resolution images.

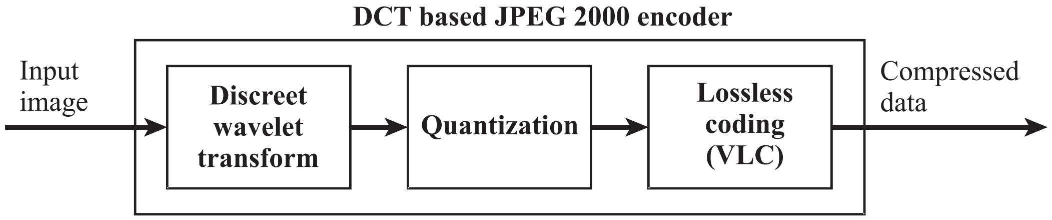


The ordering in the zig-zag reading to queue the coefficients in a 1D vector.

JPEG also supports lossless coding based on prediction coding and lossless VLC coding, omitting the DCT transform and spectral operations. The typical compression ratio is approximately 2:1.

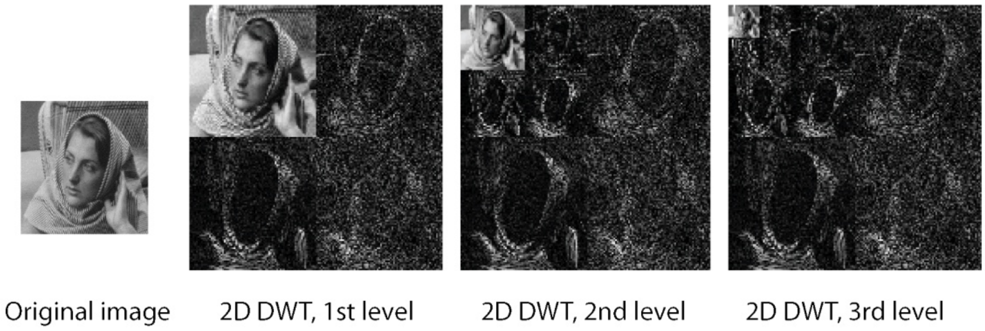
JPEG 2000

The JPEG 2000 format attempts to be the successor of the imperfect but still much more preferred JPEG format. It extends the possibilities of its predecessor, improves quality/compression ratio and allows for scalable lossy to lossless compression. Other improvements include region of interest coding where important parts of an image are coded more precisely than the rest.



General scheme of the JPEG2000 encoder

The transform function changed from DCT to 1D Discreet Wavelet Transform (DWT). The original image is wavelet transformed, quantized and entropy coded. The main difference between DCT and wavelet transform is that DWT divides the blocks of image into subblocks which are then divided into subblocks, etc.



Original image

2D DWT, 1st level

2D DWT, 2nd level

2D DWT, 3rd level

An example of the 2D DWT-transformed image. See how details are distributed in the blocks. Each block is created by DWT-transforming in the directions horizontal, vertical and diagonal. The lower level is divided into higher level blocks.

Wavelet is a wave-like part of a function which, unlike say sine function which continues from infinity to infinity, has its start, amplitude and end. It can be of various shapes that we choose depending on the signal we analyse.

Wavelet transform basically takes the wavelet and compares its similarity to a part of the analysed signal. As the wavelet has beginning and end, it can be stretched to any scale. If the scale is chosen in defined steps multi-resolution wavelet spectrum is obtained.

The DWT creates 2 sets of samples, the low-pass and high-pass samples. To successfully reconstruct the signal only high-pass samples from each level of resolution are needed. They represent the details to be added to the lower level of resolution to construct the higher level.

GIF

The *Graphics Interchange Format* (**GIF**) is still a popular image file format on the Internet. Introduced in 1987, it is a bitmap format with support for 8-bit colour palette, transparency and provides good compression ratio. Due to the limited colour palette (255 colours) it has limited use for high fidelity images, such as photos. It is, however, suitable for limited colour images, such as logos, with sharp edges and minimal colour transitions. The second version (1989) of the file format brings support for transparency.

GIF uses the Lempel-Ziv-Welch (LZW) algorithm to compress the image data, assigning byte sequences in a dictionary to colours in the colour palette.

Even though new, more advanced algorithms exist, such as PNG, GIF still has popularity thanks to its support for animation by placing multiple images on top of another. This feature has been exploited to allow for true colour (24 bit) images and animations by placing three 8-bit frames on top of each other, each containing a part of the 24 bit colour palette.

PNG

Portable Network Graphics is a bitmap image format meant to replace GIF with its licensing issues and technical limitations. It was proposed in 1996 and became international ISO/IEC standard in 2004. **PNG** supports RGB and RGBA colour spaces with 8 bits per colour (24 bit RGB or 32 bit RGBA).

The PNG format is very flexible with its container-like structure. The image is created as a series of "chunks" which allows for distribution of the image information, layer support and data streaming.

The lossless compression works in two stages:

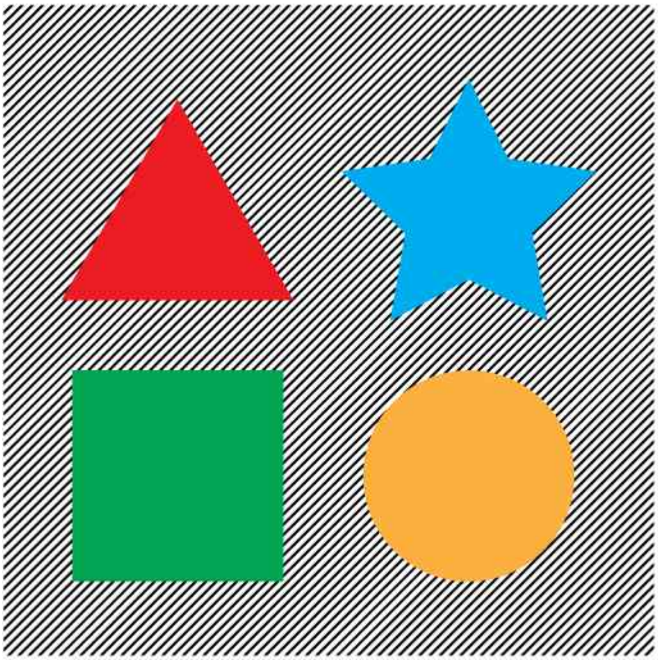
- Pre-compression (filtering)
- Compression

During pre-compression, image data is reduced using a method similar to DPCM when the pixel value is stored as a difference between it and the pixel to the left, above, above left or a combination thereof. For each line of pixels different filter type may be used. Then, the values compressed using the DEFLATE algorithm which eliminates duplicate strings using references and applies Huffman coding scheme to blocks of data rather than to the whole image.

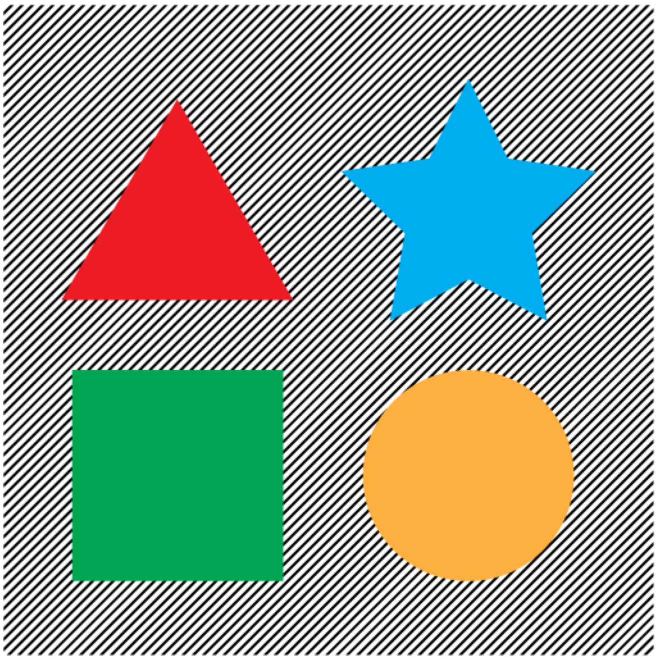
💡 The first name proposal for the format was PING meaning an anagram – "PING is not GIF".

− When compared to JPEG, the PNG algorithm produces larger files of photo-like images with subtle colour transitions.

+ However, JPEG has hard time processing sharp transitions and edges, such as text, line art or graphics, on large area of solid colour, and produces artefacts. PNG is capable of better compression and no artefacts are visible after compression which makes it ideal for web use.



The comparison of JPEG (above) and PNG (below) version of the same image. Note the imperfections at the edges of the JPEG image. The JPEG's file size is 61 kB while PNG's is only 11 kB.



WebP

The youngest of the formats, WebP came from the laboratories of Google in 2010. It is presented as a new open standard to compete the still very popular JPEG. WebP brings the best of the feature of JPEG (good performance with true-colour graphics), JPEG 2000 (both lossy and lossless image compression), PNG (alpha transparency in both lossy and lossless modes) and GIF (animation support).

The lossy algorithm is based on the VP8 video format. The compression is based on the prediction of image blocks from three blocks above and one block to the left, employing one of four modes: horizontal, vertical, DC (one colour) and TrueMotion. Wrongly predicted and non-predicted blocks are then compressed in a 4x4 pixel block with DCT or Walsh-Hadamart transform. The output is then entropy coded.

Apart from standard techniques like dictionary coding and Huffman coding, the lossless algorithm uses advanced techniques like different entropy codes for different colour channels or colour cache of the recently used colours.

When compared to other image formats, JPEG and PNG, WebP seems to outperform them by at least 20% in their focus image types. WebP is currently supported on Linux and Windows via plugins, and Firefox, Chrome and Opera support WebP as well.

5.4 Compression of video

Video, or moving pictures, is the most common part of multimedia these days. The amount of video content rises rapidly and takes by far the most data space compared to other types of media. With the growing possibilities of personal devices in terms of video creation and playback the amount of data stored and transmitted rises every day. Therefore it is necessary to reduce the size of video to lower the transmission and storage costs.

Video sequence consists of individual frames, or images, which can be encoded in fundamentally same way as static images. However, a sequence of rapidly changing images is expected to involve the same object changing its position over time, making two frames following each other similar with only small changes. Thus, if we only encode the changes we can achieve even higher compression.

Video coding methods can be divided into two main categories:

- Time domain (waveform) coding
- Model based image coding

Waveform coding methods employ transform coding with intra-frame image prediction and are widely used in a variety of formats. **Model based image coding** is used in limited bandwidth applications, such as video telephony, with bitrates up to 64 kbit/s. The methods exploit typical video telephony scenes with constant content, minimal movement and reduced frame rate.

Intra-frame prediction and motion compensation

To reduce the time redundancy between two frames the intra-frame prediction with motion compensation which works in two stages:

- Motion estimation
- Motion compensation

During motion estimation a motion vector is determined which covers relative movement of an image block from the previous to the current frame. If the position is predicable then only the motion vector is necessary to transmit. There are two algorithms for motion estimation:

- Pel recursive algorithms
- Block matching algorithms

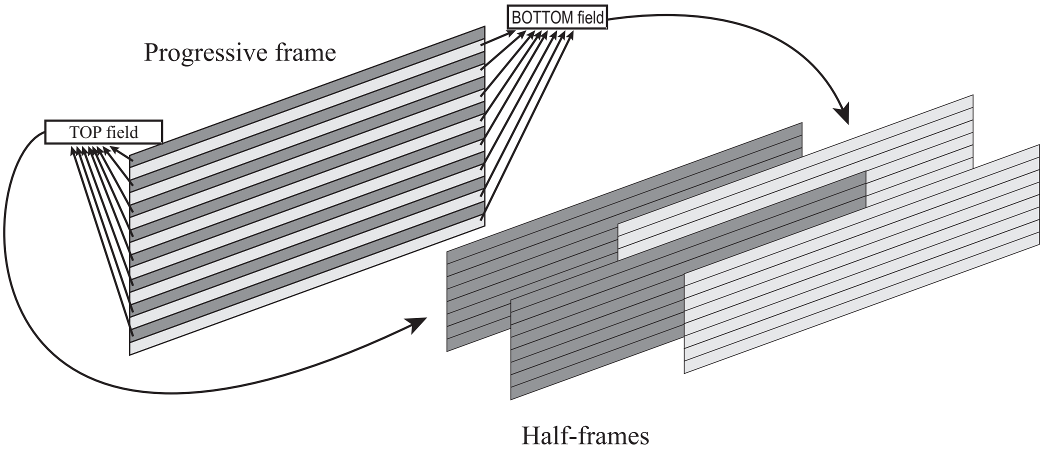
Pel recursive algorithms attempt to iteratively minimize the prediction error. Since they highly depend on local statistical distances they cannot estimate greater distances and are suitable for slow motion video, i.e. video telephony. The block matching algorithms presume that all parts of the image block move the same direction. The current frame is divided into blocks and for each of them the most similar block is searched in the previous frame.

Motion compensation algorithms use the motion vectors to move each block from the previous frame into the new position in the current frame. Thus, the prediction frame is created which is coded and transmitted.

Video compression techniques

Interlaced video

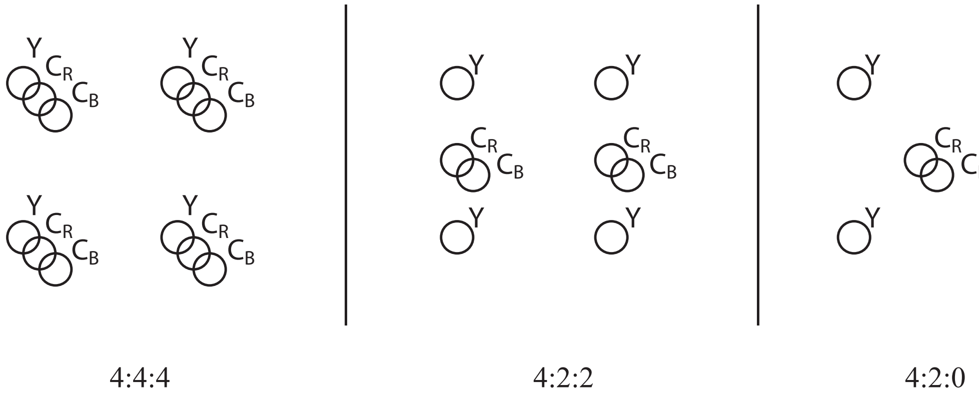
Standard (progressive or non-interlaced) video consists of 25 (Europe) or 29.97 (USA) frames per second. Interlaced video consists of double the amount of half-frames per second (50 for Europe or 59.94 for USA). Each half-frame contains either odd (top field) or even (bottom field) lines of the standard video frames which are decoded and played back in interlacing order.



Difference between progressive and interlaced video. A full frame is decomposed into top and bottom fields, creating half-frames which are alternately put in a sequence. The video has then double the frame rate per second.

Colour subsampling

Another coding technique is colour subsampling. Each pixel of the video frame encoded in YUV colour space consists of three subpixels: one luminance (brightness) and two chrominance (colour) pixels. For the set of four pixels in square this is denoted as 4:4:4. However, the human eye is more sensitive to brightness than to colour change. This allows us to reduce the number of chrominance pixels to half or even three quarters. In the first case the four luminance pixels are covered by four chrominance pixels (4:2:2), and in the second case four luminance pixels are covered with only two chrominance pixels (4:2:0 or 4:1:1).



Examples of colour subsampling. The left image shows that each luma subpixel has its own pair of chroma subpixels. In the middle, each two luma subpixels share a pair of chroma subpixels. On the right, four luma subpixels share one pair of chroma subpixels.

Similar to static images, the video compression techniques involve mainly hybrid methods combining time domain coding with transform coding (such as discreet cosine transform (DCT) or discreet wavelet transform (DWT)). The prediction frame created in the process of motion compensation is subtracted from the current frame creating the error frame. The error frame is block coded using DCT and the transformation coefficients are quantized and "zig-zag" coded using variable length coding (VLC). The VLC sequence is then transmitted.

On the receiver side, the reconstruction is performed the inverse way, performing inverse VLC, inverse quantization and inverse DCT.

The described approach is used with modifications in both MPEG and H.26x coding standards.

MPEG

The Motion Picture Experts Group (MPEG) was assigned by the ISO and IEC organizations to create standards for audio and video compression. As a result, following standards have been proposed, each focusing on different application:

- MPEG-1
- MPEG-2
- MPEG-4

The MPEG-1 focuses on interactive systems based on CD-ROM media. MPEG-2 extends the capabilities of MPEG-1 for digital television and High Definition TV (HDTV). MPEG-4 focuses on multimedia applications with very low bitrate.

MPEG-1

The standard was created to encode the video signal with sufficient quality at bitrate of 1.4 Mbit/s. It supports fast forward and backward playback and image freeze. While typical frame size is 352x288 px (CIF) the codec supports frame size up to 720x576 px at 30 fps and 1.86 Mbit/s.

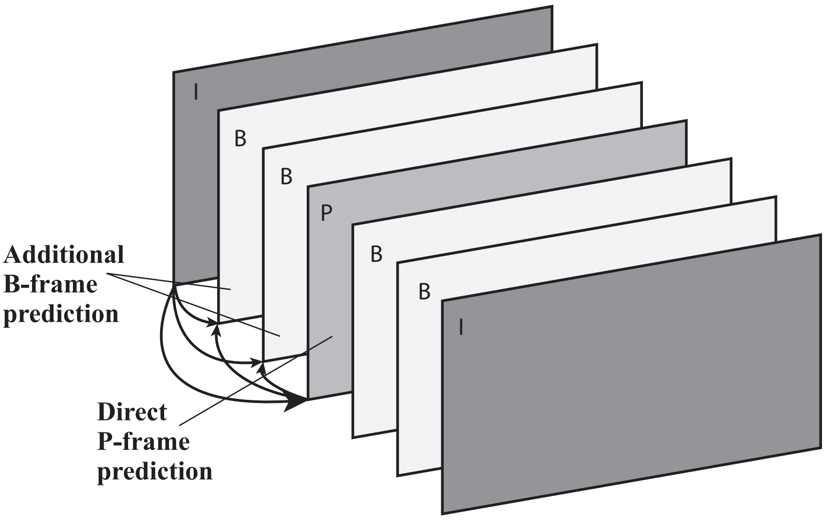
The intra-frame coding in MPEG-1 is based on intra-frame prediction and DCT coding. There are 3 types of macroblocks (sets of four luminance blocks and two chrominance blocks) leading to three types of frames:

- **I frames** – Intra frames – frame coding
- **P frames** – Predicted frames – direct prediction intra-frame coding
- **B frames** – Bidirectional frames – 2-way prediction/interpolation

The I frame, in blocks of 8x8, is only DCT coded, coefficients are quantized and "zig-zag" ordered and finally entropy coded. No motion estimation is done so the frame appears as a screenshot, is independent from other frames and serves as a stop point during fast forward or backward playback.

The P frame is coded with intra-frame prediction coding and compared to the previous frame. Thus, the error frame is created (represented by the motion vector) which is split into 16x16 pixel macroblocks and DCT coded, quantized and entropy coded similarly to the I frame. However, these frames do not contain the whole image information since they depend from the previous frames and only serve as the reference frames for prediction, not for fast forward or backward playback.

The B frames are only acquired using forward or backward prediction from I or P frames. B frames usually serve as padding or added details in fast scenes between I and P frames which contain the same information as P frames, anyway. No prediction is done out of B frames.



The ordering of I, P and B frames and their dependencies

The frames may be combined flexibly to fulfil the application requirements. The IIIIIIIII sequence offers high frame access and fast forward and backward playback, however, low compression applied to I frames puts high demands on the necessary bandwidth. Typically, a IBBPBBPBBPBB(I) sequence is used (called the Group of pictures (GOP)), applying the I frame approximately every half a second.

MPEG-2

MPEG-2 is an extension of the previous standard (with backward compatibility) with the possibility of interlaced video, improved maximum image resolution, TV video quality at bitrates of 4 to 8 Mbit/s and HDTV video quality at 20 Mbit/s.

Through profiles and levels various properties of decoders are specified. Each level defines a set of parameters specifying the target application of the video while profiles specify complexity of the used algorithms. The following tables describe levels and profiles in detail.

List of MPEG-2 levels

Level	HIGH	HIGH 1440	MAIN	LOW
Parameters	1920x1152 px 60 fps 80 Mbit/s	1440x1152 px 60 fps 60 Mbit/s	720x576 px 30 fps 15 Mbit/s	352x288 px 30 fps 4 Mbit/s

List of MPEG-2 profiles

Profile	**Algorithms**
High	All Spatial Scalable profile functions plus 3-layer spatial scaling and SNR scaling modes Colour model YUV 4:2:2 for demanding tasks
Spatial Scalable	All SNR Scalable profile functions plus 2-layer spatial scaling coding mode Colour model YUV 4:2:0
SNR Scalable	All Main profile functions plus 2-layer SNR (Signal-to-Noise ration) scaling coding mode Colour model YUV 4:2:0
Main	No scaling, interlaced video coding Random frame access, prediction mode with B frames Colour model YUV 4:2:0
Simple	All Main profile functions supported except for prediction mode with B frames Colour model YUV 4:2:0

Video scaling

Scaling enables decoders to play back low-bitrate video in the case they are not able to play back the high-bitrate video. The decoder receives the low-quality video and additional information allowing scaling the quality up. Using SNR scaling, the DCT coefficients are quantized roughly for the low-bitrate video. Then, the difference between the rough quantization values and the true value is quantized again using finer quantization and the information is sent separately to enable on demand video quality upscaling. With spatial scaling, the video is first encoded with lower resolution and the additional data are added to enable higher resolution. If the decoding device doesn't support higher resolution it omits the additional data to decode only the low resolution video. Temporal scaling works in a similar way. The video with reduced frame rate is created and additional data are added to allow reconstruction of higher frame rate video. Spatial and temporal scaling can be combined to support variability in video coding, i.e. to support both HDTV and SDTV systems.

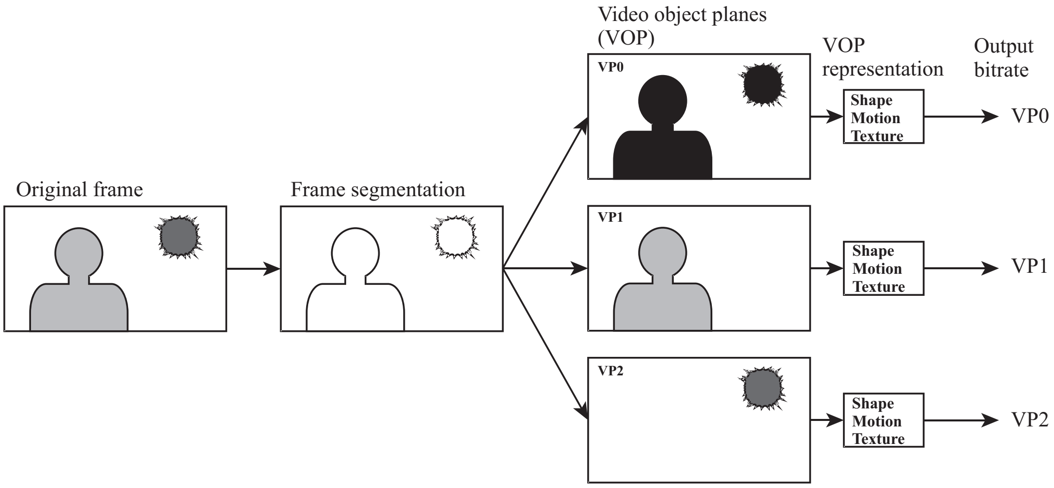
Note that MPEG-2 standard has been developed together by organisations ISO and ITU-T who named it H.262.

MPEG-4

MPEG-4 standard was developed in order to support very low bitrates up to 64 kbit/s. Its target was to enable video over the Internet, in mobile devices and networks, and to support interactivity with the objects in the scene. This required improvement of the compression methods which now utilize video object coding in both natural (standard) and synthetic (wire-frame objects coding) video.

There are two versions of the MPEG-4 video standard. The first is referred to as Part 2, which is used by variety of codecs including DivX, XviD, Nero Digital and others. The second version is referred to as Part 10, also known as MPEG-4 AVC/H.264 Advanced Video Coding, and is used in x264, Quicktime or in HD video media, such as Blu-ray Disc.

Natural video coding is performed by detection and coding of video object planes (VOP). Each VOP contains information about shape and texture of the object in the scene. A sequence of VOPs representing the same object is called the video object (VO). Each video object can then be coded using different bitrate, allowing for flexible bitrate allocation and additional object manipulation (scaling, rotation, brightness and colour variation).



Demonstration of the usage of video objects

The video object is given by its shape given by either binary or grayscale shape mask. Motion coding is based on similar principles which are used in earlier

MPEG standards but are applied to the video object planes, creating IVOP, PVOP or BVOP frames. The spatial redundancy is removed using DCT transform while temporal redundancy is dealt with by applying motion compensation. The texture of the video objects is, again, coded using a modification of DCT, the Shape adaptive (SA) DCT. Additionally, an alternative coding is permitted using SA-DWT (Discreet Wavelet Transform).

Synthetic video allows for creation of artificial objects and mix them with existing video objects in the scene. Main target is to enable facial animation for multimedia applications.

H.261 and H.263 standards

The H.261 standard was published in 1990 to enable video telephony and video conference with low bitrates 64 to 1920 kbit/s and low delay. It unites the various television standards using different row count and half-frame frequency (PAL and SECAM using 625 rows at 50 Hz, NTSC using 525 rows at 60 Hz). As a basis, it uses the CIF (352x288) and QCIF (176x144) video formats where one is used for video conferences with more participants and the other for video telephony transferring usually head and shoulders of one person.

Both CIF and QCIF are split into groups of blocks (GOBs), the CIF into GOB 1-12 and QCIF into GOB 1, 3 and 5. Each GOB is then split into 33 macroblocks, each containing 6 blocks: 4 brightness (luminance) and two colour (chrominance – C_R, C_B), each comprising 8x8 pixels.

The H.261 codec uses only I-frames (called the keyframes) and P-frames which are obtained using the motion prediction from I-frames or the previous P-frames. The standard doesn't use B-frames.

The coding algorithm uses hybrid block coding involving intra-frame prediction with motion compensation and DCT-based transform coding, which is pretty similar to MPEG-1 coding. After spatial and temporal redundancy is removed, each block is DCT transformed, quantized and then put in sequence using zig-zag algorithm and Huffman (lossless) coded. Additionally, a loop filter is used to smooth out the differences between blocks of the predicted image, improving the intra-frame prediction.

Loop filter works with a sequence of frames and removes the blocking artefacts introduced by the DCT transform of each block. Its task is to smooth out the hard edges between the blocks of the frame. The smoothing is performed repeatedly in a loop until the threshold is reached. Even though the loop may take more processing time than the decoder but in the end, the motion estimation may produce smaller motion vectors to be encoded.

The H.263 standard brings higher efficiency coding compared to H.261. Thanks to using some techniques from MPEG-1 it provides bitrate reduction up to 50% while maintaining the same subjective quality. In comparison to H.261, the H.263 standard brings wider video format support (SQCIF, 4CIF, 16CIF), improved motion vector estimation, modified VLC coding and introduction of PB-frames.

Motion estimation in H.263 works with half-pel (half-pixel) prediction. While H.261 motion vectors only work with integers, vectors in H.263 are represented with 0.5 precision. Moreover, the motion vector of a macroblock is estimated based on comparison (median calculation) with motion vectors of the nearby macroblocks and only difference between the estimated and real motion vector is transmitted (this is called median prediction).

The PB-frame mode works in a similar way the MPEG-1 codec. The P-frame is obtained from the previous I-frame or P-frame. The B-frame is obtained by 2-way prediction from the surrounding frames. The difference between MPEG-1 and H.263 is that the H.263 B-frame is contained within the P-frame, forming the PB-frame. This is beneficial for low-bitrate videos.

An extension to the H.263 standard, the H.263+ brings robustness against transfer errors, dynamic scene resolution and frame scaling.

MPEG-4 AVC/H.264

The other (and newer) version of MPEG-4, known as the advanced video coding, is the most used standard today. It is maintained jointly by the ISO and ITU-T organizations and particularly suitable for high definition video compression.

The standard brings many improvements to the previous standards of both organizations, such as higher resolution colour information, scalable video coding, and multi-view video coding which enables coding of more angles of video and allows for stereoscopic (3D) video.

Variable block size allows precise precise segmentation of moving regions with sizes ranging from 16x16 pixels to 4x4 pixels. Multiple motion vectors can be derived from one macroblock pointing to different reference pictures. The motion compensation algorithm works with quarter-pixel precision (compared to H.263's half-pixel precision) enabling higher motion vector accuracy. The DCT transform is improved and adjusted in a way which allows exactly specified decoding. Additionally, a secondary Hadamard transform can be used in smooth regions to further improve the compression ratio.

Moreover, lossless macroblock coding is introduced allowing perfect representation of specific areas of the image, working in two modes, the PCM macroblock or enhanced lossless macroblock (with greater efficiency). Entropy coding introduces new algorithms, the Context-Adaptive Binary Arithmetic Coding and Context-Adaptive Variable-Length Coding, to encode syntax elements and quantized transform coefficient values more effectively than in previous standards.

There are plenty of other improvements leading to maintaining the same subjective quality as the older standards at half or even less bitrate, especially on high bitrate and high resolution videos.

Similarly to MPEG-2, the MPEG-4 AVC/H.264 standard supports coding profiles to be used in different target applications and levels defining the required decoder performance.

WebM

WebM is an open source audio and video codec by Google to be used with HTML5 video. It is a multimedia container based on Matroska which contains Ogg Vorbis-coded audio and VP8-coded video.

The VP8 codec was developed by On2 Technologies and was released under open source license by Google who bought the company in 2010. Even though the VP8 codec uses many of the techniques introduced by both MPEG and H.26x standards it brings more improvements with the aim to keep the high subjective quality while reducing the computing complexity. Some of the improvements follow.

The codec uses the so-called constructed reference frame which serves as a reference frame for motion compensation of more than one predicted frame. The look of the constructed reference frame is not specified so it is up to the designers. The loop filtering procedure which removes the blocking artefacts from the spatial redundancy removal (DCT transform) can use different number of blocks in sequence per each block. The entropy coding uses mostly binary arithmetic coding which is adaptive to each frame individually.

Apart from the coding standards mentioned before, there are plenty of other video formats, for example Ogg Theora, which is based on an older VP3 codec by On2 Technologies, the Windows Media Video (WMV) by Microsoft and many more standards not covered by ISO or ITU-T organizations.

6 Multimedia processing

6.1 Speech synthesis

Speech synthesis means creating human-like speech using a machine, which is known as speech synthesizer.

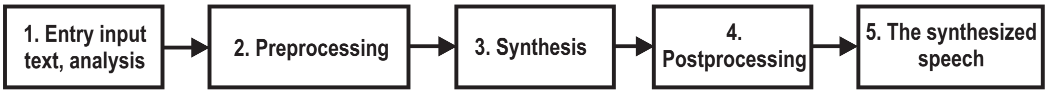
There are several types of these synthesizers, but each is made to do the same: to reproduce the given text in the clearest and most understandable manner.

There are four basic approaches:

- Synthesis using units (a.k.a. diphone synthesis, human samples are concatenated together to form words, described below)
- Formant synthesis (computer generated speech instead of human samples, easy to generate, but sounds artificial)
- Articulation synthesis (is model based on a human vocal tract model and articulation processes, it is not commonly used)
- HMM synthesis (mathematical model generating speech based on maximal likelihood criterion)

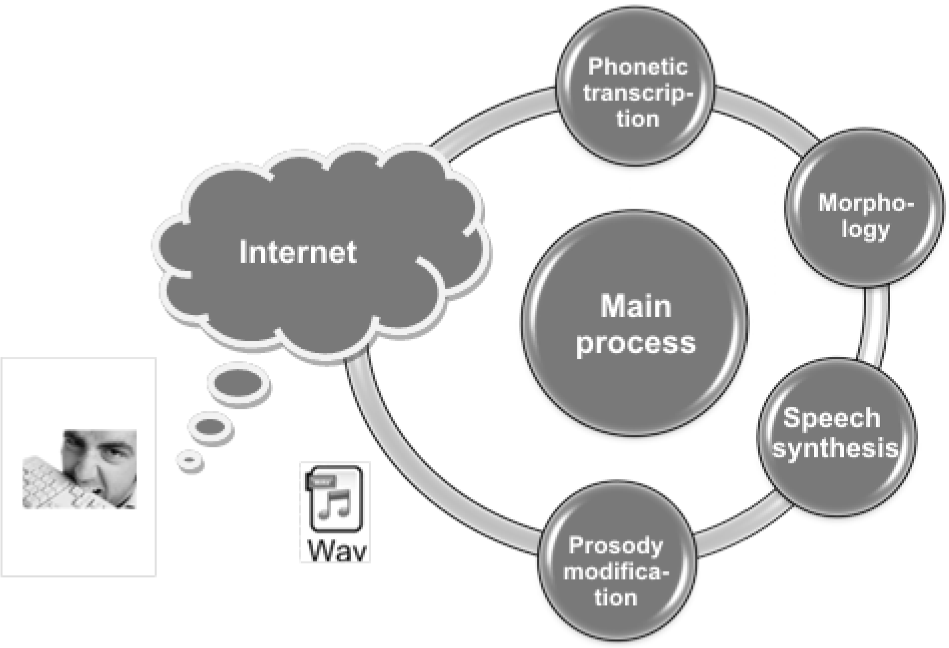
On the picture is block diagram of a general synthesizer. Of course this diagram is simplified to our needs and some elements (such as feedback found in some learning synthesizers, etc.) are omitted. However, virtually every synthesizer consists of following parts:

- Entry text input
- Preprocessing – words or sentences are „translated“ into form which can be understood by computer: numbers and all abbreviations are rewritten as words in correct form, all letters are written in special alphabet for speech synthesis called Sampa alphabet. SAMPA (Speech Assessment Methods Phonetic Alphabet) is a phonetic translation which uses only printable ASCII characters
- Synthesis
- Post processing – final synthesized speech is modified to be more natural, this means mainly prosody modification.
- The synthesized speech



Block diagram of general synthesizer

To achieve natural synthesized speech the synthesizers should fulfill more complex tasks like preprocessing and post-processing. Ideally, to be as humanly as possible, to system should be adaptive and able to learn. Such system would consist of four basic modules: phonetic transcription of words, word class identification (mainly for German and Slavic languages which use inflection), phonetic transcription of abbreviations and prosody modification module (such as different accent and speed when asking, when commanding etc.).



Synthesizer’s modular architecture

In the following example the focus in only on synthesis process.

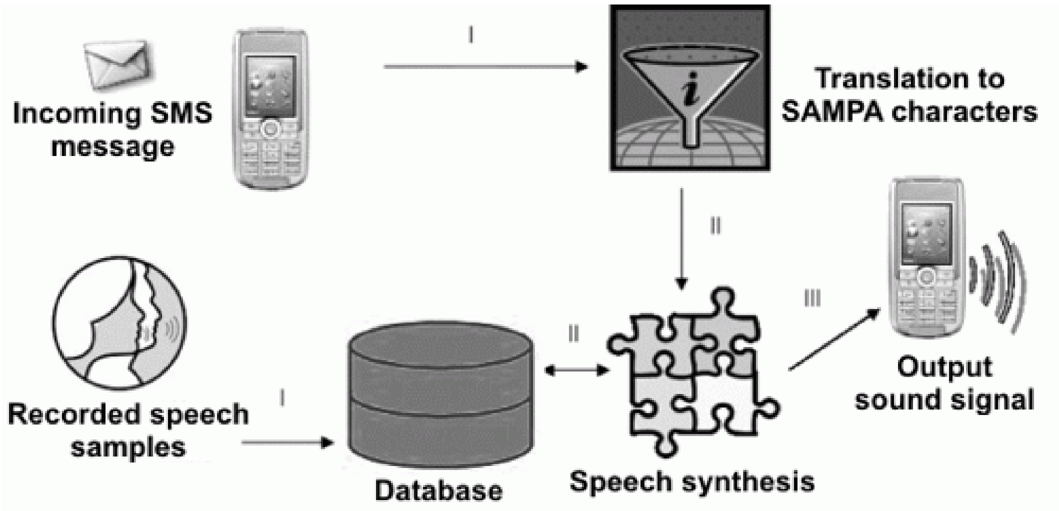
Diphone synthesis example

This is example where can be speech synthesizer used. The major advantage of this solution is naturally sounding voice and a small database size. Slovak language has only 1550 diphones and this makes the size of the solution very reasonable (especially compared to other approaches).

A diphone, together with phoneme, is one of major speech elements. It consists of two neighboring phonemes. The boundaries of diphone are in the middle of these sounds. This means, that a diphone length is not double, as one might suspect, but approximately the same as length of one phoneme. The advantage of using diphones and not phonemes is that they better represent change between sounds, because they boundaries are in the middle of sounds where the characteristic time curve is most stable.

In theory, the number of diphones is the square of number of phonemes (all combinations of two phonemes is a square). However, the real number is lower, because the particular language does not use, or does not utilize all of them. We can get the real number of diphones by closely studying the language. Diphone database consists of real speech recordings which are broken into small parts – diphones. There are two options how to create and record this database. Either to choose words, which will cover all diphones from a dictionary, or use some other approach. These words in dictionary don't have to have a meaning; the aim is to have smallest possible set of recordings.

Design of a diphone synthesizer is on picture below. It describes in very simple form how the synthesizer works.



Synthesizer design

The input text has to be synthesized into speech. But at first it has to be broken down into SAMPA. In the first step all characters are retyped to SAMPA. In the second step result from the first step is retyped according all rules for pronunciation for each language (in our case Slovak language). For each diphone is checked if it is in the database and the corresponding units are selected and concatenated together. The output is synthesized text.

Some examples of speech enabled applications are: personal speech assistants, mobile assistants for the blind, working city guides, timetables and navigation systems, web-based multimodal services, applications for documenting traffic accident reports, or speech-based inventory and time management services. Nowadays very popular are book readers with implementing speech synthesis, mainly for English language.

6.2 Image recognition

A common problem with images is determining whether an image contains some specific object, or feature. This problem is currently solved only for specific object (we are looking for faces), but not for arbitrary objects on a image (e.g. list what you see on this particular image). Therefore we recognize multiple areas of recognition, such as:

- Barcodes – barcode consists of bars of varying width representing digits. These widths are of course relative, therefore bars have to be scaled and approximate matches have to be found.
- 2D codes – also known as QR codes, or matrix codes are basically a 2D barcodes, which can store much more data than simple barcodes. They can store various types of data, such as URLs, numbers, or text. They can be of various size and density and contain RS codes with embedded data correction capability
- Optical Character Recognition (OCR) – identifying characters in images of printed or handwritten text, which basically is specific pattern recognition.
- Fingerprint identification – a pattern based algorithm compares basic fingerprint patterns (arches, loops and whorls) in an image with predefined template (original finger scan)
- Specific detection – image is scanned for certain conditions, such as movements in digital security cameras
- Face recognition – an image is searched face patterns
- Object recognition – one or several pre-specified or learned objects or object classes can be recognized, such as augmented reality programs like Nokia Lens, or Google Goggles

Each recognition type uses its own algorithm, which can be simple, or advanced – such as use of advanced statistics and neural networks. As an example we provide a simple face detection algorithm with forehead and chin detection on images with defined pixel sizes.

Face points detection

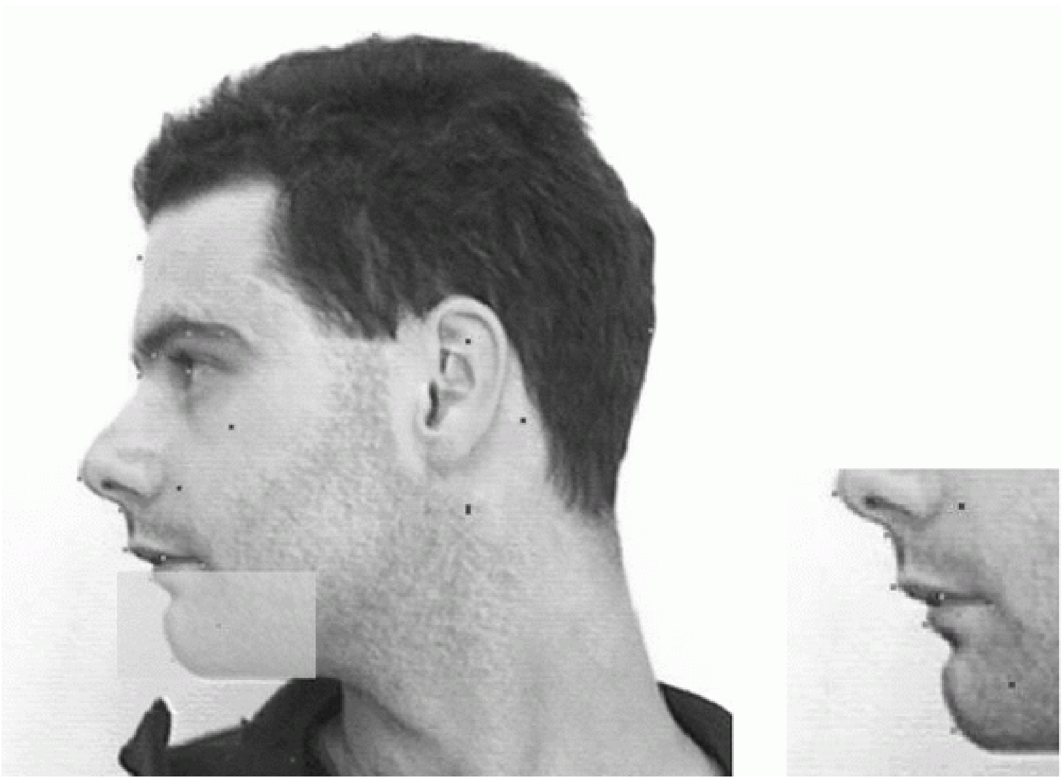
Facial feature detection is based on human skin chromacity and morphological characteristics of the human head. Output of the skin detection is used to isolate further important points of the analyzed face. As the first step of detection, noise has to be removed, with use of low band-pass filter. This is necessary in order to avoid unwanted (and incorrect) detections. Next step would be edge detection, via a special Sobel filter. After this necessary „preparation“, the preprocessed picture can be used to start detection of major face elements (eye, nose, mouth, chin…) based on known characteristics of these elements.

Chin detection starts in the area below the lower lip, with the actual search proceeding down. The boundary of horizontal search is approximately in the third of the image (this assumption is based on the image size and image position criteria). We look for image area where the edge enters the image followed by sharp rise, which is the end of chin.

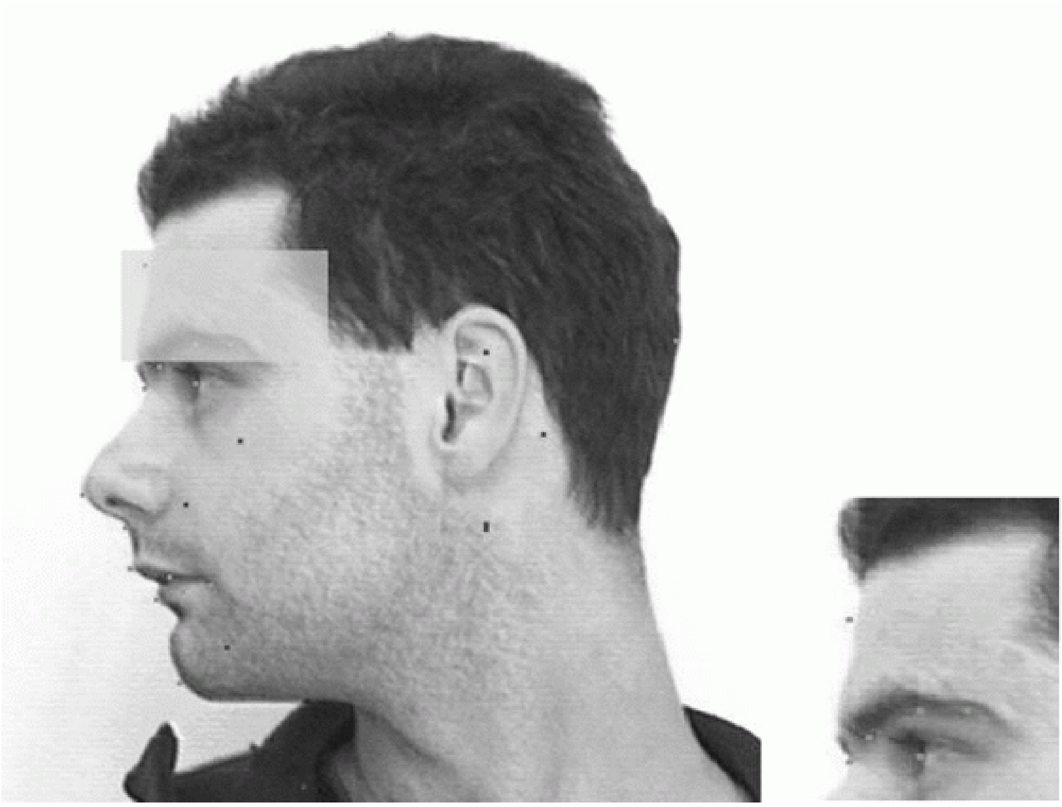
The detection is comparing the horizontal distance of two points, both of which are on the edge. The Y position of the second point is less by 10 pixels (this value is valid only for our algorithm with specific size of picture). If the horizontal distance of these points is bigger than expected, the point is designated as the chin.

The area of forehead detection is in the first third in horizontal direction, and in the first quarter in vertical direction of the image. We want to find the point where hair meets the skin.

For this detection the output of skin detection is used, too. We are looking for a point on the edge. Than we start to search the "Skin detect" image, right direction of this point. When the search finds the point with color value 255 (hair color in RGB in the output picture for skin detection) it looks on the difference of its X position and X position of the edge point. If the difference is lower than the given value, we found the point of forehead.



Chin detection



Forehead detection

Based on face points detection a general 3D model of human head can be deformed and personalized.

6.3 Face animation

The human face can be modeled and animated by many methods. For choosing the best method we need to define requirements for our own module of animation. The list of requirements for module of animation is:

- The animation has to by continuous?
- The performance has to be optimized for PC or mobile phone?
- The animation has to be synchronous with synthesized speech?
- The quality of the animation should be the best, final animation should be as similar to reality as possible.

There are two basic approaches for face animation: two-dimensional and three-dimensional and real time animation and forward calculated animation.

The real time animation allows users an interactive intervention to animation and reduces the time necessary for the preparation of the animation. The disadvantage of this method is its quality of an image. The calculation of each image should not be longer than approximately 0.05s, because there should be at least 20 images per second (20 FPS – frames per second). Quality of three-dimensional modeling is better than the quality of two-dimensional approach. This animation models the reality more naturally.

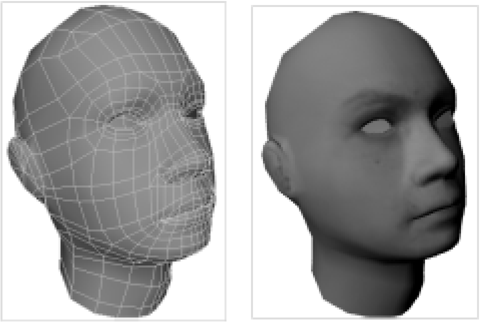
For the face animation are the following methods known: interpolation, parameterization, simulation of muscles, etc.

There are many various techniques for modeling human face in space, for example polygonal modelling, modelling by parametric areas, sub-division modeling.

Talking head on mobile phone

This chapter introduce multimedia speech synthesis project. It describes an application of the speech synthesizer together with the face animation for a mobile cell phone. The ultimate goal of this project was to develop a multimedia communication system without the need to transfer any video and audio data. The results concern a mobile phone Java application for reading of *short messages* (**SMS**). After receiving SMS (short message service), a talking head based on a sender's photo appears on the screen and animate the reading while the speech will be synthesized in parallel.

In our example both the model and the visemes needed for face animation are in the object (OBJ) format, which means the files include not only the model, but also the texture which belongs to it.



Neutral face model and face texture

The viseme, we are referring to is a deformed model of the face. This is not just any kind of deformation; it is the deformation as if the face was saying the given phoneme. The model of the visemes still has the same number of nodes, which have the same numbering scheme and are connected with the same lines as the neutral model. The only change is the position of the nodes. Because of this simplification we are able to perform an easy interpolation of the nodes and the orientation of the subsurfaces defined by the nodes.

The animation itself is realized by the before mentioned viseme interpolation. The neutral model is read from the file (together with the other models and visemes). The interpolation is performed between these models. The animation is also based on the time (this is a real time type of animation). The model is also deformed based on results from face points detection.

Face visemes

Speech synthesis use diphone speech synthesizer described in chapter Speech synthesis. The synthesized text has to be synchronized with the speaking face.

Application on mobile phone: personalized face (first one) and general model (second nd third one)

6.4 Speech recognition

A genuine speech is created by human vocal organs and can be observed as special vibrations of air. It contains except many other lexical information that is vital for speech recognition. This information is coded into the signal as a sequence of different acoustic sounds. Each language consists of set of basic sounds called phonemes that are used in building the whole vocabulary of a particular language. Their number varies based on the sort of a language; its usual number approximately ranges from 40 to 60. Unfortunately adjacent phonemes influence each other in an acoustical way. Moreover, they differ from speaker to speaker (they contain speaker specific information). Further, there is a background noise and distortion inflicted by the environment and recording device. Finally, vocabularies of developed languages may have several hundreds of thousand even millions words including all the cases, times, genders, names, etc. All this may indicate the task is extremely difficult, complex and computational time demanding.

Thus there have been designed many speech recognition systems that are categorized based on distinctive traits as: small, middle and large vocabulary systems, speaker depended and independent systems, speech modelling units (phonemes, syllables, words, phrases, etc.), systems for isolated words and continual speech recognition, etc. Systems working with over hundred thousands of words in real time scenarios have been reported.

For a couple of decades there has been a great effort spent to build and employ automatic speech recognition (ASR) systems in areas like information retrieval systems, dialog systems, etc., but only as the technology further evolved other applications like dictation systems or even automatic transcription of natural speech are emerging. In order these advanced systems are to be wide spread employed, they should be capable to operate on a real time bases, must be speaker independent, reaching high accuracy and support dictionaries containing several hundreds of thousands of words.

Speech Feature Extraction Methods

One of the first steps in the design of an ASR system is to decide which feature extraction technique to use. At the beginning it should be noted that this task is not yet completely solved and a lot of effort is still going on in this area. The aim is to simulate the auditory system of humans, mathematically describe it, simplify for practical handling and optionally adapt it for a correct and simple use with the selected types of recognition and classification methods.

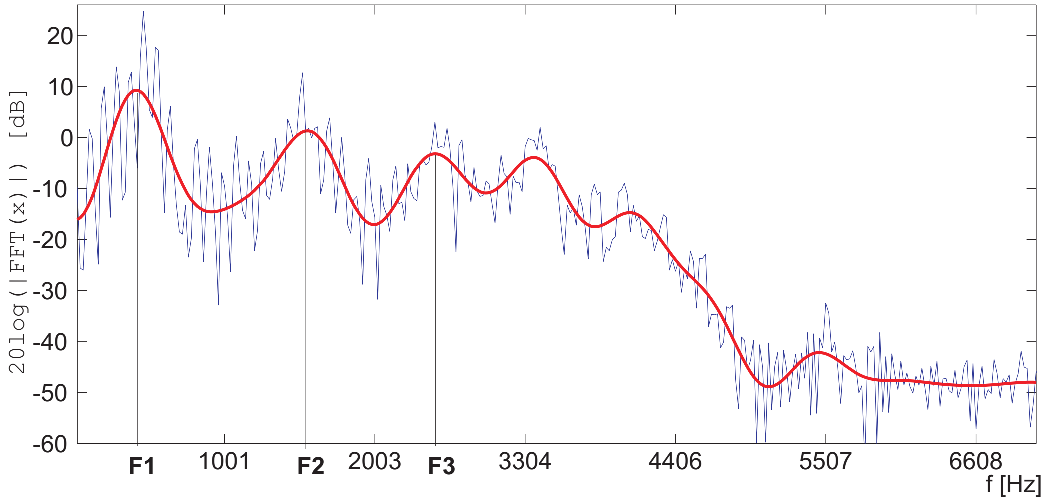
As already said before, speech is a very complex signal produced by vocal organs of human beings. Such signal contains many kinds of information e.g. lexical (what is said), speaker identification part (who speaks), what actual mood he/she is in, where is he/she from (dialect), social status, health condition, speech disorder, and many more. All that information is indirectly encoded into the final speech signal during the speech production process using brain and vocal organs.

The information stream of a speech signal is about 100kb/s that can be approximately estimated using its frequency band and the number of needed quantization levels. However, when the lexical information is tested it was found based on the number of phones per second that it is approximately 10b/s.

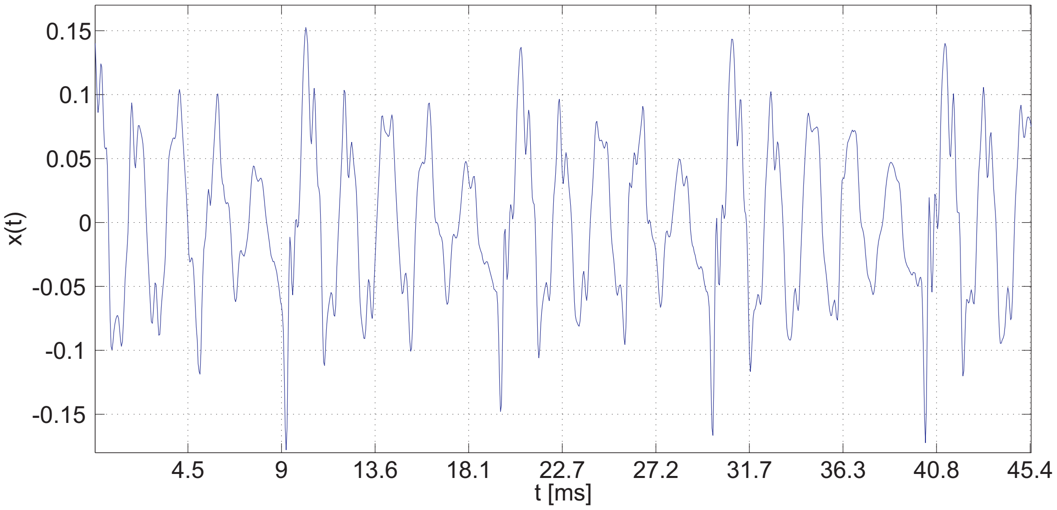
This shows the big redundancy in terms of lexical information. Therefore the extraction method must significantly reduce the information bit rate and keep only the lexical one. However, this task is very complicated as the process of coding all parts of information into a single speech signal is rather complex and not fully reversible.

Naturally there are many speech feature extraction methods that either mimics the speech production process (linear model of speech production) or they simulate the human auditory system (critical bands, EIH) rather than computing anything else. The idea is quite straightforward, i.e. the human auditory system has evolved during several hundred of thousand years to extract “only” the relevant information out of the general audio signal suppressing different sorts of noises and distortions.

The required basic characteristics of extracted features are: massive bit rate reduction, features must be “deaf” to sounds and changes that are difficult to perceive by humans, and on contrary, must be sensitive to variations that are perceivable. Based on an extensive research it was found that good indicators of differences in perception are the so called format frequencies. In the following picture there is a frequency spectrum for a vowel “e”, its magnitude envelop (has a relation to a setting of vocal organs during speech production) and depicted formant frequencies. There is also a time domain of the analyzed sound in the next picture.



Spectra, formant frequencies and a spectral envelope for a vowel “e”.



A time course of a vowel “e”.

To show how formant frequencies are related to the classification of vowels, in the following table first tree formant frequencies for all vowels are listed separately for males and females. From the table the positive shift of formats for females is obvious as well.

Averaged locations of formant frequencies for males and females

vowel	Males			Females		
	F1 [Hz]	F2 [Hz]	F3 [Hz]	F1 [Hz]	F2 [Hz]	F3 [Hz]
a	730	1100	2450	850	1200	2800
e	530	1850	2500	600	2350	3000
i	400	2000	2550	430	2500	3100
o	570	850	2400	590	900	2700
u	440	1000	2250	470	1150	2700

As it was already emphasized a good feature should be sensitive to differences in sounds that are perceived as different in humans and should be “deaf” to those which are unheeded by our auditory system. It was found that the following differences are audible:

- Different number of formant frequencies
- Differences in the position of formant frequencies
- Differences in widths of formant frequencies
- The intensity of signals is perceived non-linearly. Differences in intensities are perceived in relative manners rather than absolute values e.g. lauder parts are attenuated and weaker intensified. It was observed hat this functionality can be mathematically approximated by logarithm function.

On the other hand, following aspects do not play a role in perceiving acoustical differences:

- overall tilt of the spectra of the form $X(\omega)=S(\omega)*\omega^{\alpha}$, where α is tilt factor
- filtering out frequencies laying under the first formant frequency
- removing frequencies above the 3rd format frequency
- a narrow band stop filtering

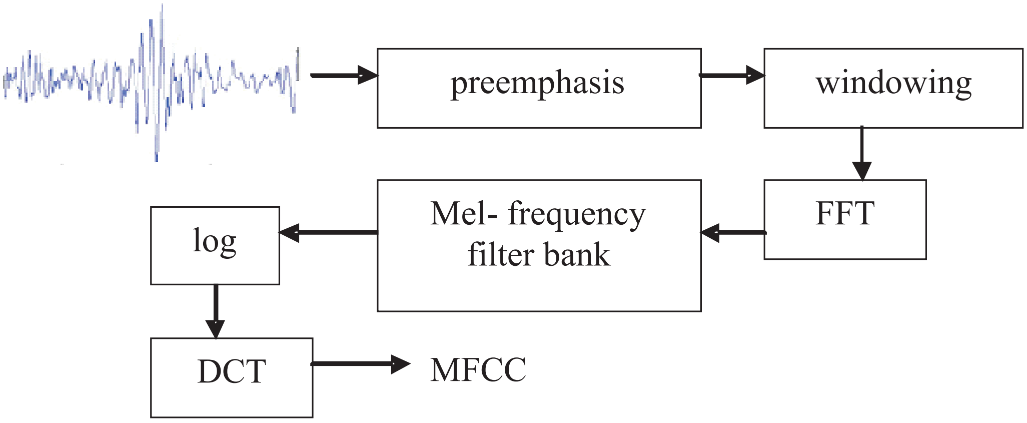
Furthermore, proper features should be insensitive to additive and convolutional noises or at least they should represent them in such a way that these distortions are easy to locate and suppress in the feature space. Next, features must be mathematically tractable, compact and easy to implement in the real time. Finally, when using continuous density hidden Markov models it is required for the feasibility purposes that the elements of feature vectors should be linearly independent so that a single diagonal covariance matrix can be used.

There exist many basic speech features, but currently MFCC and PLP are the most preferable features for speech recognition and speaker identification problems. Both MFCC and PLP were designed to simulate the human auditory system aiming to achieve better scores in speech recognition tasks.

Therefore, both PLP and MFCC are designed to capture positions and widths of formants that are most perceivable. Further, they have an easy interpretation and compact representation (Euklidian distance has a reasonable acoustical meaning). Both features are kind of modified cepstral coefficients, but they differ in the calculation process. In the following a brief overview of MFCC and PLP is given.

MFCC calculation process

MFCC uses preemphasis filter (high pass) to suppress the low pass character of the lip radiation. Prior to DFT computation Hamming window is applied and the spectra is warped into the Mel scale to emulate critical bands. Then, equally spaced triangular windows with 50% overlap are used to simulate a filter bank. Finally the logarithm and DCT transform are invoked. The logarithm not only produces cepstrum but simulates how the intensities are non-linearly perceived by humans. The role of DCT is mainly to decorrelate elements of a vector. The MFCC calculation process is shown in the next figure.



MFCC calculation process.

PLP Calculation

Unlike MFCC PLP calculation engages following steps: Hamming windowing and FFT calculation, frequency warping into Bark scale, smoothing the bark-scaled spectra by a window simulating critical bands, sampling the smoothed bark spectrum in approx. 1 bark intervals to simulate the filter bank, equal loudness weighting (approximates the hearing sensitivity), transformation of energies into loudness (powering magnitudes to 0.33), calculating linear prediction coefficients from the warped and modified spectra, finally cepstral LPC coefficients are derived from LPC. As it can be seen PLP provides more complex human like auditory processing than MFCC, however they usually provide comparable results in the task of speech recognition and laboratory conditions. On the other hand PLP usually shows greater robustness in adverse conditions.

Auxiliary features – time dynamic and energy

As the speech evolve in the time, it is also important to capture these transitions, which may contain additional speech relevant information. Thus delta and acceleration (double delta) coefficients constructed over acoustic features in the time are often derived as well. These can be calculated as simple differences using two consecutive frames or in more general way as a linear combination of differences based on a wider time span (-L, L), making the estimation more robust.

$$\Delta(n) = m\sum_{k=-L}^{L} kx(n+k) \text{ and } \Delta\Delta(n) = m\sum_{k=-L}^{L} k\Delta(n+k)$$

In addition to these features energy information is sometimes extracted too. Obviously the segment’s energy itself does not provide much acoustical information to classify the segment. However; its evolution in the time copies the locations of vowels, fricatives or pauses that may be quite discriminative. That is why the energy dynamic is usually used as well.

Unfortunately, yet there is no feature that would ideally incorporate all the requirements mentioned in this survey. Thus new methods are emerging and some of them achieve even better performance, however it is usually observed mainly in particular sorts of environments and systems thus they may not have general applications.

Recognition Techniques

After the speech feature extraction process each speech signal is presented as a sequence of speech vectors. Therefore it is necessary to compare or asses each unknown sequence to the reference one (model or a sample of a speech sample that we know its lexical content, i.e. they were present in the training phase) that can represent word, phrase or a whole sentence. Based on the samples or models we use i.e. sub word, word or phrases, we distinguish recognition systems. If phrases or word models/ samples are use they can embrace the so called coarticulation effect (phonemes are influenced by adjacent phonemes and their positions in a word or even in a sentence) which plays a role. However sub word models are rather limited in their number thus they are more practical as each sample must exist in several different realizations in the training database prior the recognition process. Next, systems differ according whether they recognize isolated words or continuous speech which is substantially difficult problem as the word boundaries are unknown. The speech recognition process is a specific one because the sequences differ in their lengths. Furthermore, the variations (shortenings or prolongations) are nonlinear within words as some parts have almost constant lengths, but other phonemes are subject to great variability. Thus simple linear length normalization techniques like decimation (shortening) or interpolation (prolongation) are not so effective.

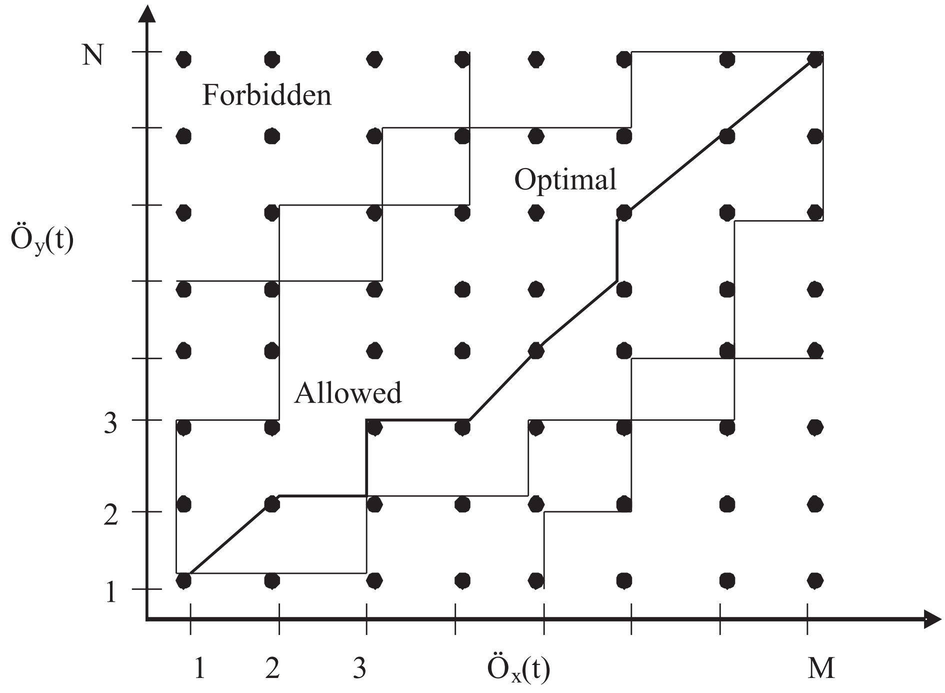
Summing all the requirements and adverse effects involved in the speech recognition, it is clear that more methods have evolved for the classification of speech sequences. However, the most successful ones are DTW and HMM, thus in the following both of them will be briefly introduced.

DTW

DTW (dynamic time warping) is a method that acoustically compares speech feature sequences of two utterances, reference and unknown one. Its main advantage is a nonlinear warping of a time scale during the comparison process. This is necessary to eliminate the differences in the sequences' lengths. Furthermore, it compensates nonlinear length variations within words.

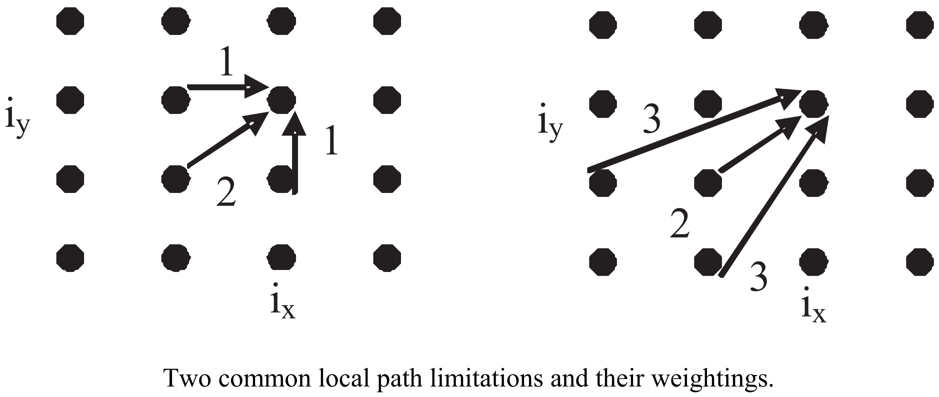
In its basic form the method requires to know word boundaries of both the referenced and test sequences. This involves the usage of additional speech detection algorithm which is also a rather difficult task especially in noisy conditions.

The method is based on a dynamic programming problem and nonlinearly warps time indexes for both reference and test sequences. This nonlinear and constrained mapping of a test sequence to a reference one follows the least acoustical error principle between them. However, this mapping must follow certain logical limitations like: beginning and end points of both sequences must be mapped on each other respectively, the functions warping the time for reference and test sequences must be non decreasing, there must be a natural (maximal allowed) difference threshold between real time indexes of mapped vectors. As a part of DTW calculation process there are used two matrices, local and global one. The local matrix simply contains acoustic distances of reference and test feature vectors among each other. The global one preserves the cumulated minimal distance (and path) between reference and test sequences calculated from the beginning to a certain position in a matrix. However, there are natural limitations on local directions, i.e. how it is possible to move from one point to subsequent ones. An example of a global matrix with depicted limitations on a global path and an optimal path is shown in the following picture.



A global distance matrix with the optimal path and limitations on a global path.

In the following picture the most common direction limitations and weighting of local paths are shown.



Two common local path limitations and their weightings.

This method had significant position in the area of speech recognition at the beginning of the research period, especially for isolated word recognition. However, as the requirements were growing, especially the speaker independence one and continual speech recognition, it lost its position to HMM method, which solves both problems in a mathematically elegant way.

Hidden Markov Models (HMM)

The method of hidden Markov Model is based on a statistical modeling of speech, rather than direct comparison of reference and unknown signals. Speech is decomposed into relevant units (from language and signal processing point of view), like phonemes, syllables, words, phrases, etc. As all statistical models are trained from multiple examples the speaker independence is relatively easy to implement having a large database. Further, the construction of HMM models allows for a simple concatenation of basic units (models) to form whole phrases, sentences or even a continual speech. The method is based on a first order Markov chain concept used for modeling static processes. By doing so it is mathematically easy to calculate the probability of a time sequence of discrete states. Thus the method is very effective even thought it violates the process of real speech production in time.

The Markov chain is defined by a set of static states $S_1,\ldots, S_N$, transition matrix P_{NxN} that gives transitions probabilities between states, and a vector of initial probabilities π for each state. Then matrix P and vector π are given as follows:

$$P=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}, \pi=\left[P(S_0=1),P(S_0=2),\ldots,P(S_0=N)\right]$$

Having such model the probability of an occurrence of state sequence S_1, S_2, S_3,.. S_N is given by:

$$P(S_1,S_2,\ldots,S_{N-1},S_N)=\pi(s_1)p_{12}\ldots.p_{(N-1)N}$$

However, this model alone cannot describe dynamic processes as speech signals. No matter what speech event a single state is related to, the speech events have many (theoretically infinite) different signal realizations. If not all, at lest the majority of them must be described by a single state. This can be achieved by adding another stochastic process that provides the probability of an observation (feature vector extracted from a speech) in a given state, i.e. P(X/Si). Thus each state will have addition probability process attached to it. However, this process has nothing to do with the time progress rather than with the variability of feature vectors (observations) in a state. It is vital as there are many (theoretically infinite number) signal realizations for each phoneme, etc. Combining these two stochastic processes we obtain Hidden Markov Models (HMM). There are three ways how to model observation probabilities in a state, thus we distinguish 3 types of HMM models:

- Discrete HMM
- Continuous HMM
- Semi continuous HMM

Discrete HMM assumes that the observation vectors (speech features) are drawn from a finite discrete set. This can be achieved by applying vector quantization (VQ) to observation vectors. Then each observation vector is represented by a singe vector from a VQ code book. This code book is constructed over a training set so that the training vectors replaced by limited set of vectors achieve minimal (acoustical) distortions, e.g. if the code book has L vectors then any speech would be presented as vector sequence that consist of only L different vectors. Therefore each state must hold probabilities of these L vectors being observed in the state, i.e. L probabilities.

+ This is a simple and fast method and in the case of a present noise VQ may provide certain sort of noise removing process and thus the recognition process may be more robust to a certain degree.

− However VQ process introduces permanent errors that may be significant especially in the case of small code books and small training databases.

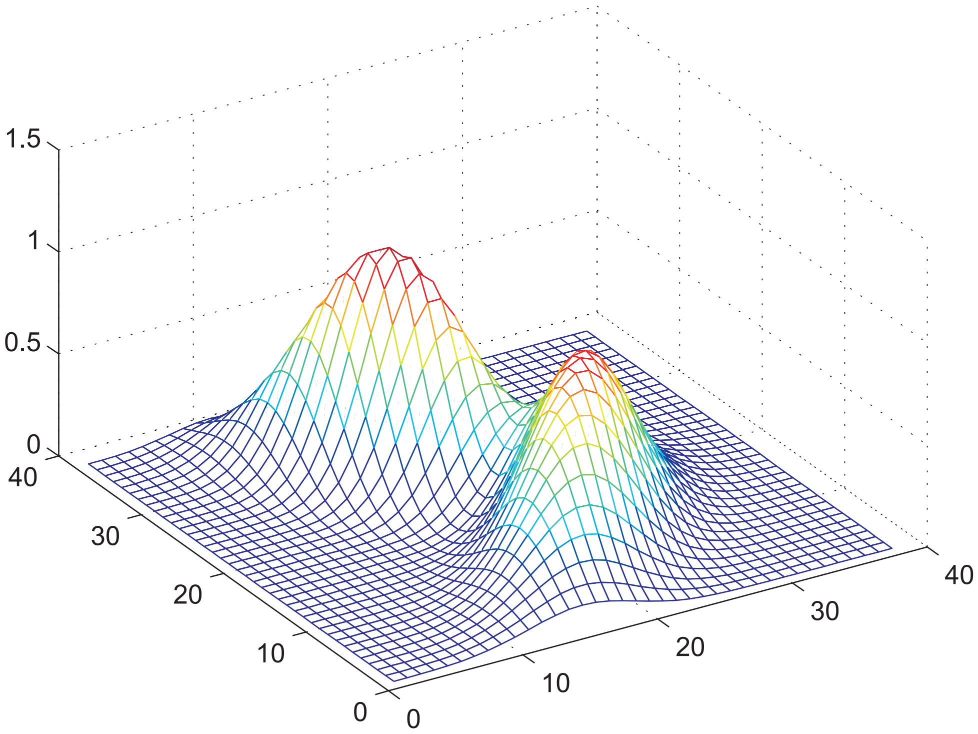
Continuous HMM act as a standard technique in the current speech recognition systems that provide good results for variety of applications.

The probability of an observation in a state (P(x/Si)) is modeled by a weighted mixture of P Gaussian distributions given by:

$$f(x/S_i)=\sum_{i=1}^{P} c_i \frac{1}{\sqrt{(2\pi)^d \left|U_i\right|}} e^{-\frac{1}{2}(x-\mu_i)^t U_i^{-1}(x-\mu_i)}$$

Then such a probability is given by weighting coefficients (c_i), mean vectors (μ) and covariance matrices (Σ).

To see an example of its modeling capability in a 2D feature space see the following figure.



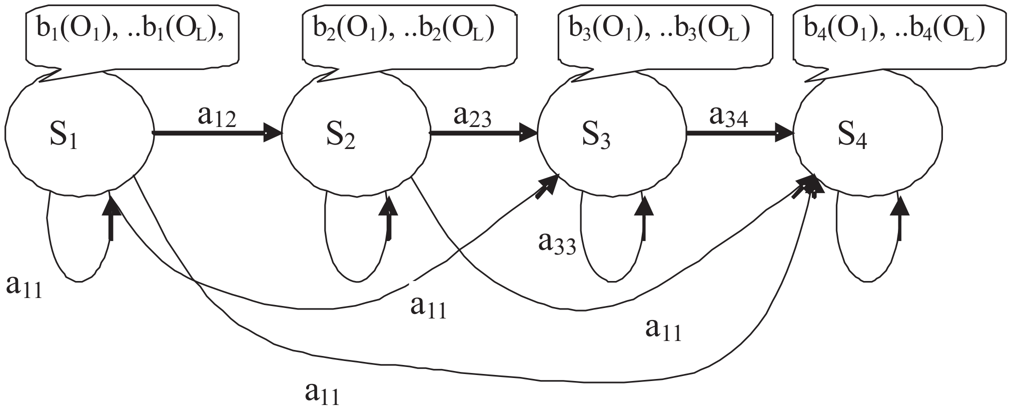
Two Gaussian mixtures in a 2D space.

+ Such model may properly describe observation space having only limited number of training examples.

– Its drawback is the increased computational complexity and memory requirements.

Semi continuous HMMs combine advantages of both discrete and continuous HMM. The process uses the global description of a feature space given by a large number of Gaussian distributions instead of a discrete code book vectors. These mixtures are shared by all states that reduces the amount of required parameters and the training and recognition processes are simplified. Then a state specific description is based on a global space description by weighting the outputs of global Gaussian mixtures, i.e. they use discrete probabilities of generating feature vectors from a particular global mixture given a certain state Si.

An example of a 4 state left-right discrete HMM model is shown in the following picture.



A 4 state left-right discrete HMM model.

Then a probability of observing a sequence of feature vectors (observations) of the length T on a model λ having N states is calculated using an auxiliary variable α as follows:

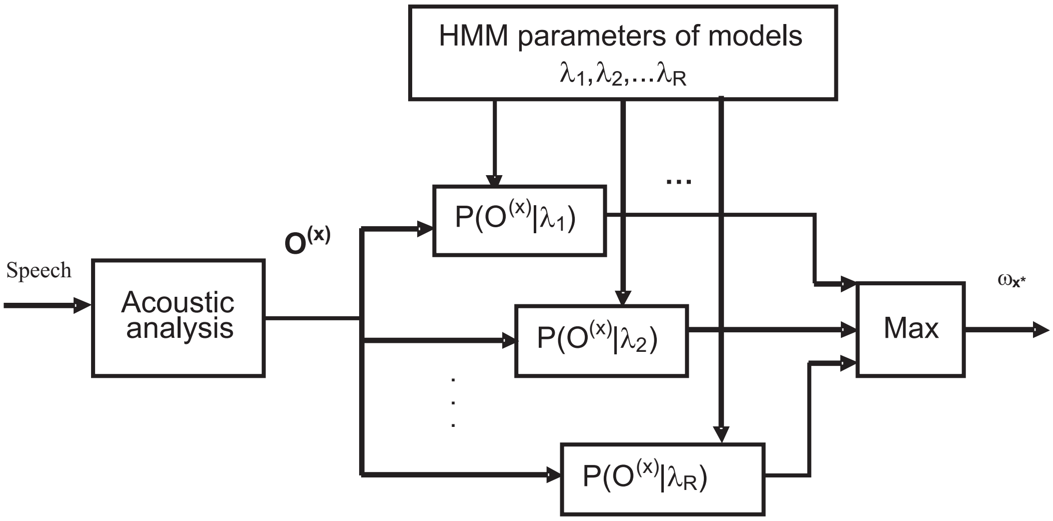
$$a_t(j) = \left[\sum_{i=1}^{N} a_{t-1}(i) * p_{ij}\right] * P(O_t / S_j) \quad \text{where} \quad j = 1,...,N, t = 2,...,T \quad \text{and}$$

$$a_1(j) = \pi(j) * pP(O_1 / S_j)$$

Final probability is given by:

$$P(O / \lambda) = \sum_{i=1}^{N} a_T(i)$$

Then the recognition process calculates the probability of an unknown sequence on all HMM models being in a dictionary and selects the one with the highest probability. The process is schematically depicted in the following picture.



The speech recognition process based on HMM.

Great advantage of HMM is a simple way how to concatenate different models in a single raw. By doing so, we can express any sentence consisting of vocabulary words.

In practical real time applications only the most probable path (sequence of states) is sought and calculated, i.e. $P(x_1,S_{t1},x_2,S_{t2},\ldots x_T,S_{tT})$. Having found this sequence of states it is possible to track back the models that created the words and further deduce upon whole sentences. This is also important from the computational time point of view. To disclose the most probable string of states (they are hidden) that produce the unknown observation the Viterbi algorithm is used.

The most difficult task related to HMM modeling is the training process, i.e. how to set all the unknown parameters (transition matrices, initial probability vectors, Gaussian mixture weights, mean vectors and covariance matrices) having several realizations of recorded speech samples. There are a few techniques that can be used but it was found there is no analytical solution of the problem. Therefore iterative methods are used that leads to local extremes only. Therefore the training process is very complex and undergoes several stages like: threshold settings, initialization of models, training of simple models, gradual enhancement of their complexity interlaced with training loops. There exist more methods how to train HMM models but the most famous and used is the Baum-Welche algorithm that is a maximum likelihood approach. Another question is the structure of models, usually left right transitions are allowed for speech events whereas each stable unit like phoneme is modeled by three states (beginning, middle and end stage). Usually each state has from 16 to 64 Gaussian mixtures with diagonal covariance matrices, but this number varies with the complexity of the task and the amount of available data. As any modeling method where the optimal structure of a model is unknown and using only limited set of data, the so called overtraining phenomenon may take place. Thus it is necessary to use a validation set to verify the performance of final models.

At presence the most advanced HMM systems use discriminative training like MMI, or MCE, and instead of Gaussian mixtures they use discriminative techniques like Support Vector Machines or Neural Networks (Deep belief networks).

6.5 Multimodal Interface

Multimodal interfaces are the current very popular technology. Everyone is talking about multimodal interfaces—how natural they are, and how users like them. Multimodal interfaces offer solutions to many of our user interface problems, as well as enabling new classes of applications.

Multimodal interface represent combination of multiple modalities, it means several ways how to communicate or interact with computer systems. Multimodal interface has been addressed by speaker identification and face recognition. The multimodal interface is responsible for seamless user recognition and authentication using modalities (voice, face detection, etc.). Beside this the multimodal interface serves for commands using gestures or voice to control the *set top box* (**STB**).

Real project where multimodal interface is integrated is HBB-Next. The project seeks to facilitate the convergence of the broadcast and Internet world by researching user-centric technologies for enriching the TV-viewing experience with social networking, multiple device access, group-tailored content recommendations, as well as the seamless mixing of broadcast content, of complementary Internet content and of user-generated content.

HBB-Next is based on modular architecture. The modules in **HBB** (*Hybrid Broadcast Broadband)* project are designed to cooperate together. Here is an example: a user enters the room, the system will recognize the user and system is automatically set according user's requirements. The user opens the AppStore application and the system allows him to choose, open, buy and install a desired application. For each activity or operation of the user, the system may ask multilevel authentication based on secure identification with satisfactory validation and security.

Multi-speaker identification aims to identify possibly more speakers based on a recorded signal that may contain utterances of more individuals. This general task can be divided into several categories based on additional refinements. If the speakers that may appear in given conversation are known in prior, i.e. they were present in some sort of training phase. The task resembles a single speaker identification problem, even though additional algorithms must be applied, tuned and enhanced. However when the set of possible speakers is unknown then the techniques of speaker segmentation and clustering (diarization) must be used. The aim of the most of application is to continuously run and "listen" to an incoming stream of PCM samples (sound waves), *detect voice activity* (**VAD**), silence and background noises, possibly recognize speech overlap, and if substantially long voice period is caught then identify the speaker with certain confidence measure. The aim of a speaker identification system is to decide the identity of a speaker upon an utterance, regardless of what he or she said. A speaker identification system consists of two main parts. The first one is feature extraction from recorded signal and the second one is classification method, which determines the speaker based on extracted paramters. These systems are usually designed for specific task, so the designer has to select proper methods and their modifications for a given application which may differ depending on type and setting of particular task.

In case of voice commands recognition the systems belonging to the group of isolated word recognition would be an option. The most successful and used ones are those based on HMM statistical speech modelling, especially those using tight context dependent phonemes as a basic modelling unit. In case of a fixed set of commands and abundance of test samples whole word models can be used to achieving potentially higher accuracies (better capturing coarticulation effects).

Generally, there are two categories of approaches when tracking person's gestures, appearance- and 3D model-based approach. The 3D model-based approach compares the input parameters of a limb with 2D projection of a 3D limb model. The appearance-based approach uses image features to model the visual appearance of a limb and compare it with extracted image features from the video input. When focusing on the latter approach, the results depend on the capabilities of the capturing device. If an RGB camera is used the methods focus on tracking the skin colour or shape of the gesturing body part. The approach, however, depends highly on the lighting conditions, as well as stability of foreground and background of the tracked subject. Also, no other skin-coloured or limb-shaped objects can appear in the examined area as they would trick the algorithm. An infra-red light depth camera uses its own IR light emitter and is thus much more resilient to lighting conditions of the scene. Moreover, the camera is capable of providing a depth map, a pseudo 3D image of the scene which can be very useful when tracking gesturing body parts, i.e. hands.

Currently in world, there are few methods how eye controlling can be implemented. The simplest and the most natural way was chosen. This system has only one part. It is the static RGB Kinect camera. The person sitting in the front of the Kinect (monitor) will be asked to look with its eyes to the highlighted points on the screen with the still head. Simultaneously the application by Kinect depth camera measure the head distance from the Kinect (monitor). For the purpose of the triangle calculation (Pythagorean Theorem) we will determine the dimensions of the screen. Application will calculate the variance of the pupil movement exactly the same like in the section calibration and also it will calculate the angles by which the pupil must be diverted from straight position to see the edge of the monitor. With knowing the head distance from monitor we can recalculate the variance and angles when the distance is changed to ensure the accuracy of the controlling.

Face recognition methods were mentioned in chapter Image recognition. Usually in real systems a list of requirements is defined for single local user identification based on a human face that the systems must/should/may implement:

- The system shall identify user inside the room by his face, if he belongs to the local users group.
- The system should be able to recognize user and compare him to locally stored user profiles without internet access.

- The system may identify user by his face inside the room unknown to the local system (not listed in the local users group).

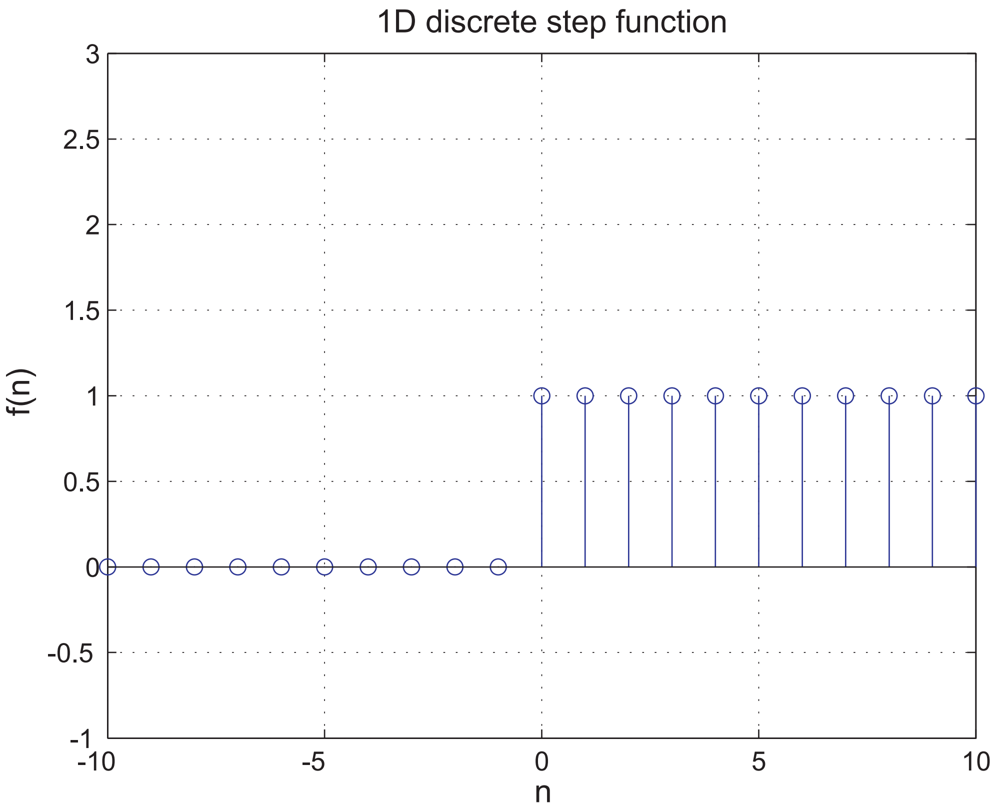
- System may recognize user based on face recognition even in dark.

7 Matlab examples

7.1 Matlab examples

Example 1

Develop a code in MATLAB for one dimensional, discrete unit step function.



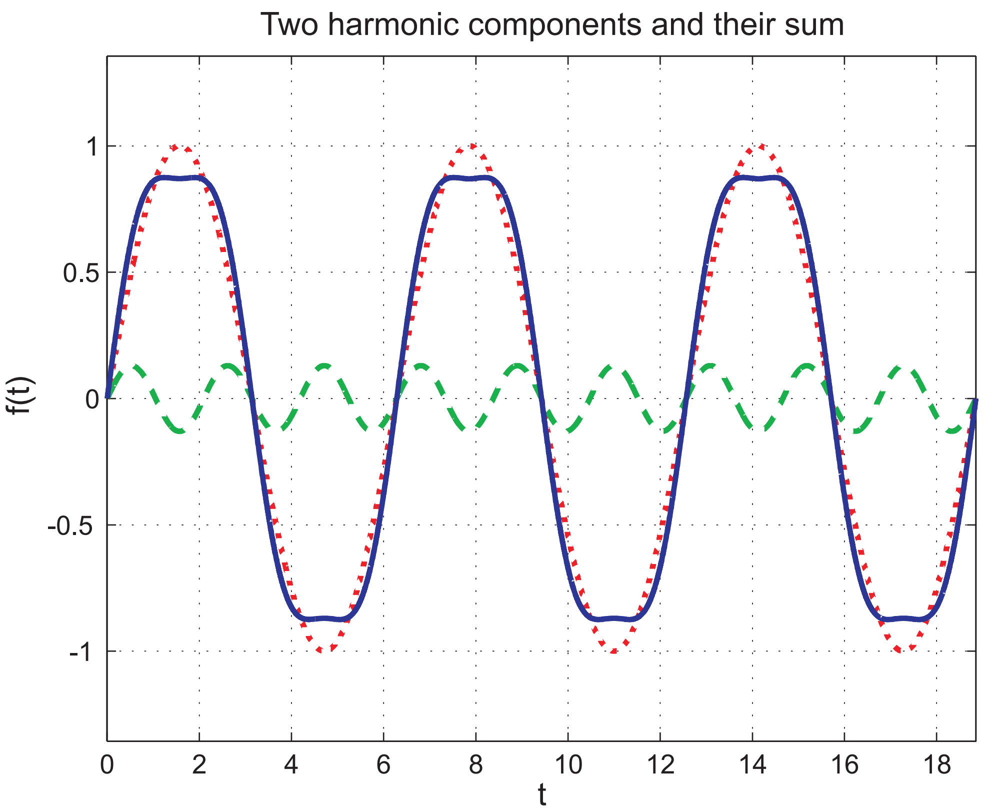
Result: Discrete unit step function

Solution

```
t=-10:20;     %definition of time period
step=heaviside(t);           %heaviside(x)   is   Matlab   function
                       and has the value 0 for x < 0, 1 for x > 0,
                       and 0.5 for x = 0.
step(t==0)=1;
figure;      %command for drawn picture
stem(t,step);
grid on;
xlabel('n');
ylabel('f(n)');
title('1D discrete step function');
axis([-10 10 -1 3]);
```

Example 2

Develop a code in MATLAB for sine wave generation.



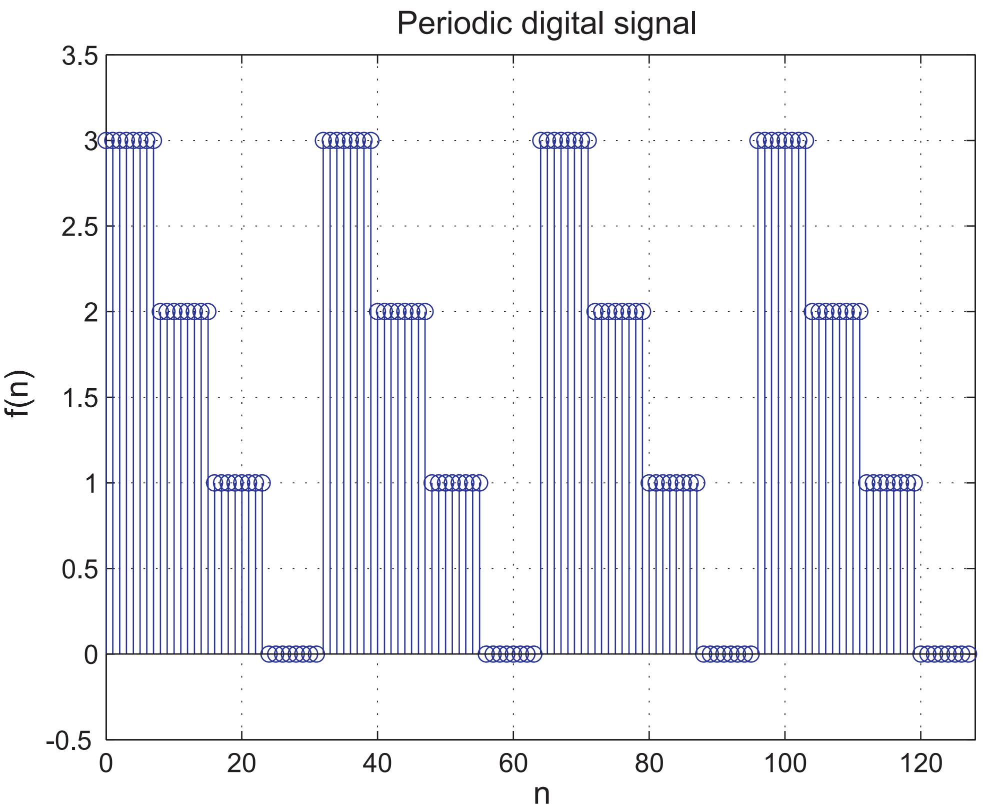
Result: Two harmonic frequencies forming signal

SOLUTION

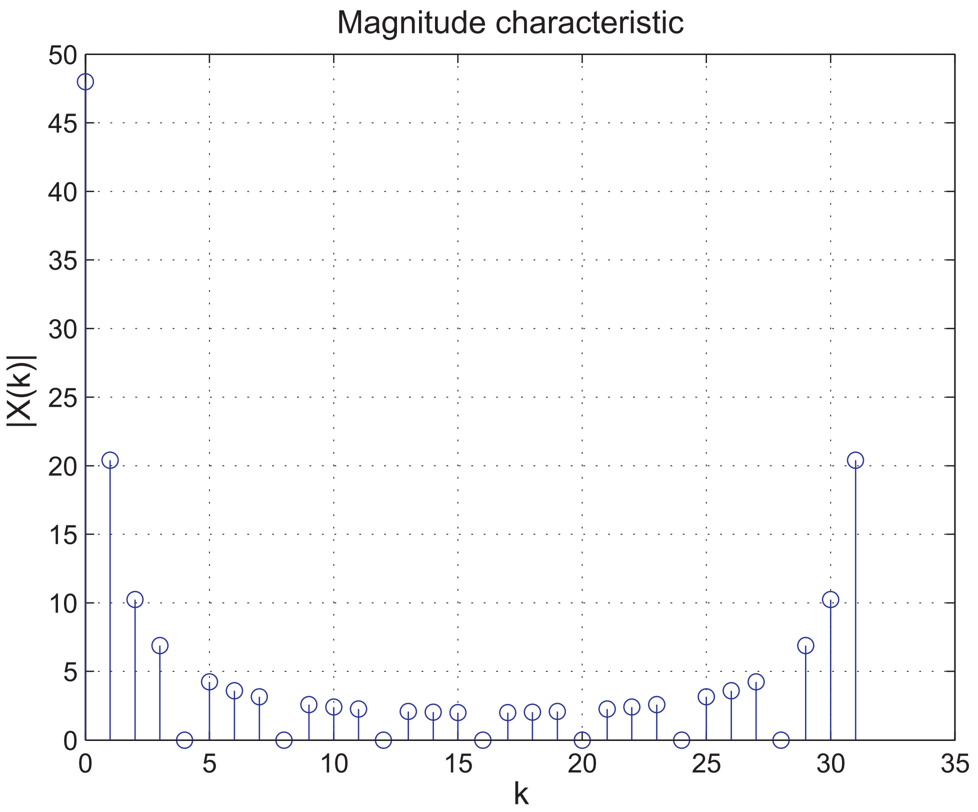
```
range=6*pi;             %max. time of the signal
t=0:0.001:range;        %time points
A=[1 0.13];             %vector of amplitudes
w=[1 3];                %vector of frequencies [Hz]
phi=[0 0];              %vector of phases
sig1=A(1)*sin(w(1)*t+phi(1));  % definition of particular signals
sig2=A(2)*sin(w(2)*t+phi(2));
signal=sig1+sig2;
figure;
plot(t,sig1,':r','LineWidth',2);
hold on;
plot(t,sig2,'--g','LineWidth',2);
hold on;
plot(t,signal,'LineWidth',2);
grid on;
axis([0 rozsah -1.2*sum(A) 1.2*sum(A)]);
xlabel('t [s]');
ylabel('f(t)');
title('Harmonic signals and their sum');
```

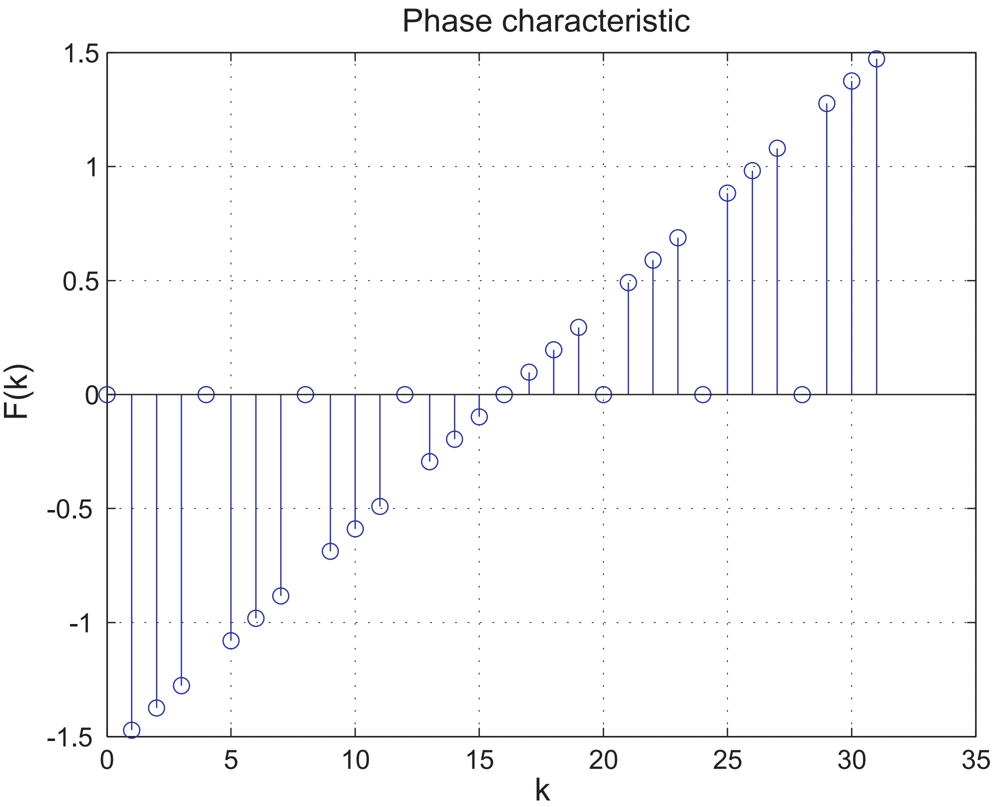
Example 3

Develop a code in MATLAB for Discrete Fourier transformation and frequency characteristics.



Examples of transfer function of LDTI, magnitude and phase frequency characteristic


Magnitude characteristic
|X(k)|
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
35
k


Phase characteristic
F(k)
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35
k

SOLUTION

```
count=32;
Ts=4/count;   %sampling frequency
per=4;        %number of periods
syms k;       %symbol variable
syms n;
signal=[3.*ones(1,count/4) 2.*ones(1, count /4) ones(1, count /4)
zeros(1, count /4)];
fn=[];
for n=1:per
    fn=[fn signal];
end
n=0:count*per-1;

figure;
stem(n,fn);
title('Discrete signal');
axis([0 length(fn) min(abs(fn))-0.5 max(abs(fn))+0.5]);
grid on;

figure;
Xk=fft(signal);     %discrete Fourier transformation
os=0:length(Xk)-1;
stem(os,abs(Xk));   %magnitude frequency characteristic
title('Magnitude frequency characteristic');
grid on;

figure;
stem(os,angle(Xk));   %phase frequency characteristic
title('Phase frequency characteristic');
grid on;
```