Fundamentals of Signal Processing

Networking for Big Data and Laboratory

M.Sc. in Data Science

Paolo Di Lorenzo

Academic Year 2024/2025

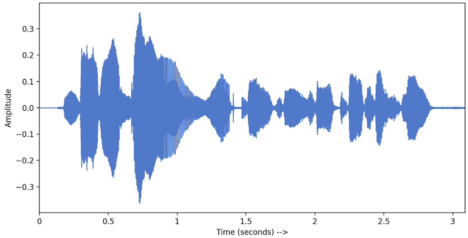
SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

What is Signal Processing?

1 Introduction

- A signal, technically yet generally speaking, is a formal description of a phenomenon evolving over time, space, or more general domains
- **Examples:**



Speech signal

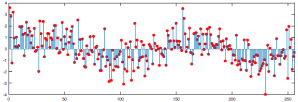
Images

What is Signal Processing?

- **Classical definition:** A signal is the physical support used to carry information
- Some examples of signals:
  - **Audio recording:** an analog pressure wave is sampled and converted to a one-dimensional discrete-time signal.
  - **Communication signals:** An analog continuous-time wave (e.g., electromagnetic, light, acoustic, etc.) carrying information that we are interest to communicate
  - **Photos:** the analog scene of light is sampled using a CCD array and stored as a two-dimensional discrete-space signal.
  - **Text:** messages are represented with collections of characters; each is assigned a standard 16-bit number and those are stored in sequence.
  - **Ratings:** for books (Goodreads), movies (Netflix), vacation rentals (Airbnb) are stored using the integers 0-5

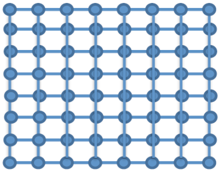
What is Signal Processing?

- In mathematical terms, a signal is a mapping from a domain $\mathcal{D}$ to a co-domain
- **1D discrete-time signals** $\mathcal{D} = \{0, 1, \dots, N-1\}$



Domain Co-domain

- **2D discrete-space signals** $\mathcal{D} = \{(0,0), (0,1), \dots, (N-1, N-1)\}$



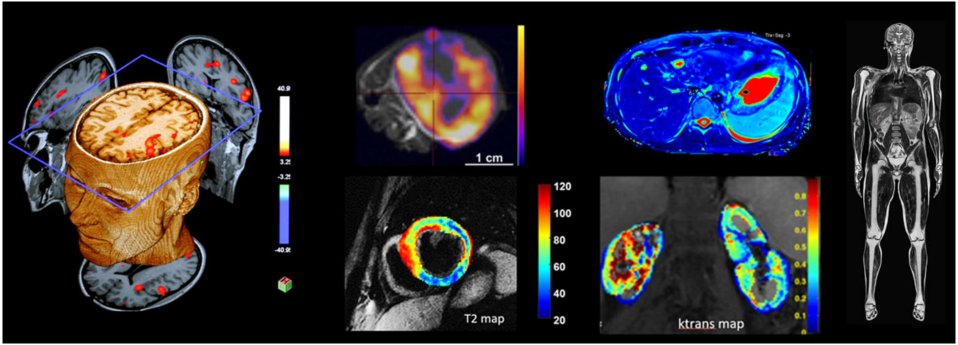
What is Signal Processing?

- Signal processing (SP) denotes any operation that modifies, extract, or otherwise manipulates the information contained in a signal
  - Convert one signal to another e.g. filter, de-noise, interpolate
  - Information extraction and interpretation, e.g., speech recognition, computer vision, remote sensing, deep learning
  - Synthesis and encoding of signals to enable efficient communication

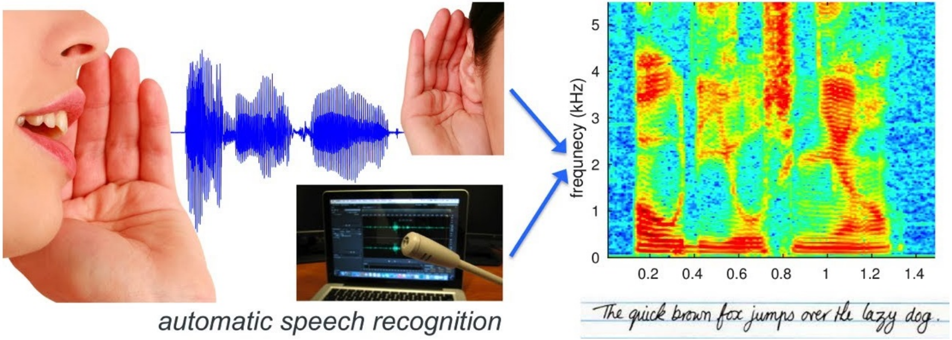
The Goal of Signal Processing

Extend human skills to process and extract information from data, enabling sophisticated sensing, communication, and learning capabilities

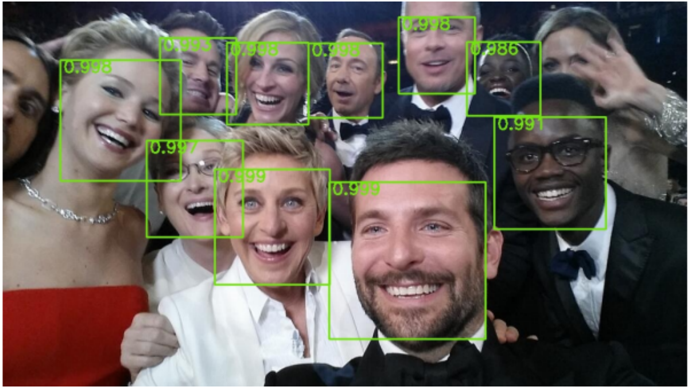
Applications: Magnetic resonance imaging



- Involves exposing the brain to magnetic fields, and recording EM signal responses of hydrogen protons to create high-resolution images of the brain

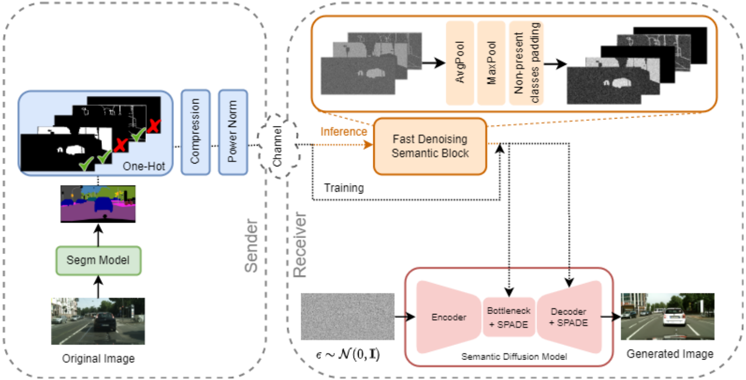


- Signal processing plays a key role in **speech feature extraction**, which is coupled with machine learning models to perform tasks such as recognition, separation, etc.



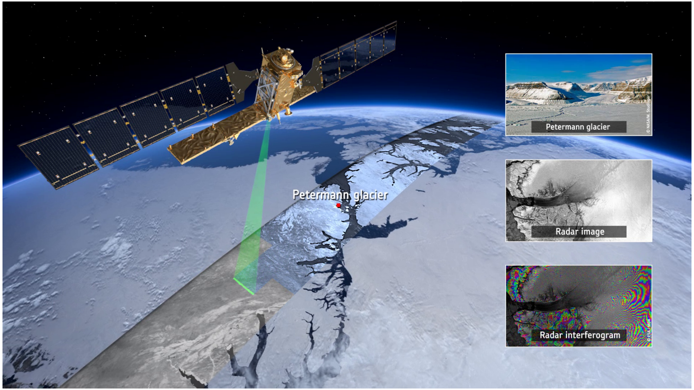
- Signal processing plays a key role to extract **intrinsic facial features**, which are then used to perform face detection and classification

Applications: (Beyond) Digital communications



- SP paved the foundations of digital communication systems, and will play a critical role in the upcoming 6G revolution involving semantic and effectiveness aspects

Applications: Remote sensing



- Signal Processing enables sophisticated remote sensing applications such planetary imaging, meteorology, and deep space exploration

Continuous-time signals

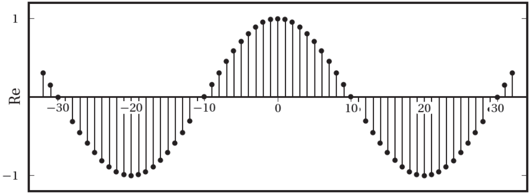
- A continuous-time signal is a complex-valued sequence $x(t)$, i.e., a complex-valued function of a real variable t, with $t \in \mathbb{R}$
- Continuous-time signals are used to model the temporal behavior of physical quantities such as, e.g., temperature, pressure, voltage, light, etc.
- Classical signal processing theory is developed in continuous-time, but real processing is typically performed by computers on discrete-time signals
- A discrete-time signal $x[n]$ can be obtained from the sampling of a continuous-time signal $x(t)$, i.e., observing $x(t)$ at given time instants nT; we say that $x[n] = x(nT)$.
- In the sequel, we will give emphasis to discrete-time signals, while making several connections with the related continuous-time counterparts

Discrete-time signals

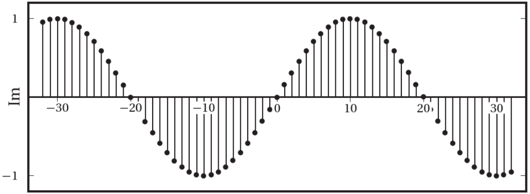
2 Signals

- A discrete-time signal is a complex-valued sequence $x[n]$, i.e., a complex-valued function of an integer index n, with $n \in \mathbb{Z}$

- **Example:** Complex sinusoid $x[n] = e^{j\frac{\pi}{20}n} = \cos(\frac{\pi}{20}n) + j\sin(\frac{\pi}{20}n)$



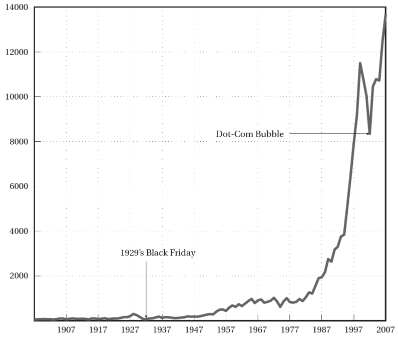
Real part



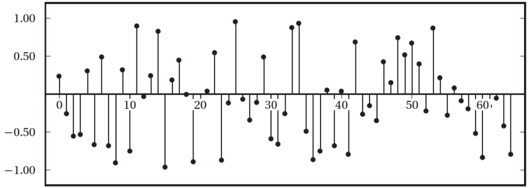
Imaginary part

Discrete-time signals

- **Examples:**



$x[n]$ = average Dow-Jones index in year n



$x[n]$ = n-th output of a random source $\mathcal{U}(-1, 1)$

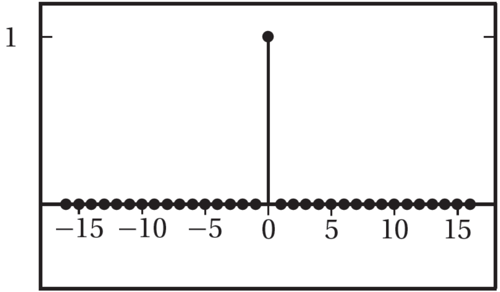
Discrete-time signals

- The dependency of the sequence's values on an integer-valued index n is made explicit by the use of square brackets (standard SP notation)
- The sequence index n is best thought of as a measure of dimensionless time, and is meant to impose a chronological order on the values of the sequences
- We consider complex-valued discrete-time signals, which are very general and useful in several domains, such as data communication systems
- In graphical representations, we resort to stem (or "lollipop") plots. When the discrete-time domain is understood, we will often use a function-like representation, which gives the illusion of continuous-time

Basic signals

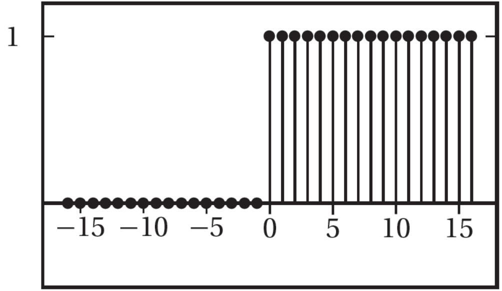
- **Impulse.** The discrete-time impulse (or discrete-time delta function) is defined as

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



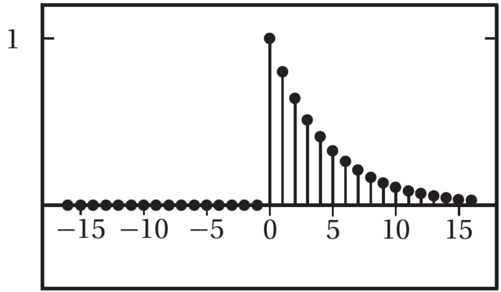
- **Unit step.** The discrete-time unit step is defined by the following expression:

$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n \neq 0 \end{cases}$$



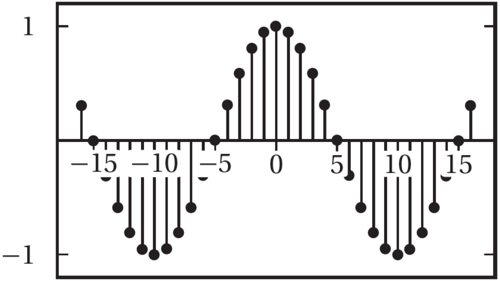
- **Exponential decay.** The discrete-time exponential decay is defined as

$$x[n] = a^n u[n] \qquad a \in \mathbb{C},\ |a| < 1$$



- **Sinusoids.** The discrete-time sinusoid is defined as

$$x[n] = \cos(2\pi kn + \phi) \qquad k \in \mathbb{R},\ \phi \in [0, 2\pi]$$



Elementary operations

- **Shift.** A sequence $x[n]$, shifted by an integer k, is simply:

$$y[n] = x[n-k] \qquad k \in \mathbb{Z}$$

  If k is positive, the signal is shifted "to the right", i.e., it is delayed; if k is negative, the signal is shifted "to the left", meaning that the signal has been advanced.

- **Scaling.** A sequence $x[n]$ scaled by a factor $\alpha \in \mathbb{C}$ is

$$y[n] = \alpha\, x[n]$$

  If α is real, then the scaling represents a simple amplification or attenuation of the signal (when $\alpha > 1$ and $\alpha < 1$, respectively). If α is complex, amplification and attenuation are compounded with a phase shift

Elementary operations

- **Sum.** The sum of two sequences $x[n]$ and $w[n]$ is their term-by-term sum:

  $$y[n] = x[n] + w[n]$$

  Please note that sumand scaling are linear operators. Informally, this means scaling and sum behave “intuitively” as:

  $$\alpha(x[n] + w[n]) = \alpha x[n] + \alpha w[n]$$

- **Product.** The product of two sequences $x[n]$ and $w[n]$ is their term-by term product

  $$y[n] = x[n]w[n]$$

Elementary operations

- **Integration.** The discrete-time integration is expressed by the following running sum:

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

  Integration computes a non-normalized running average of the discrete-time signal

- **Differentiation.** A discrete-time approximation to differentiation is the first order difference:

$$y[n] = x[n] - x[n-1]$$

- Note that the unit step can be obtained by applying the integration operator to the discrete-time impulse; conversely, the impulse can be obtained by applying the differentiation operator to the unit step.

Energy and power

2 Signals

- **Energy.** We define the energy of a discrete-time signal as

$$E_x = \sum_{k=-\infty}^{\infty} |x[k]|^2$$

  The energy is finite only if the above sum converges, i.e. if the sequence $x[n]$ is square summable. A signal with this property is sometimes referred to as a finite energy signal.

- **Power.** We define the power of a signal as the ratio of energy over time, taking the limit over the number of samples considered:

$$P_X = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} |x[k]|^2$$

  Signals whose energy is finite, have zero total power. However, signals whose energy is infinite have finite power, e.g., periodic signals

Classes of discrete-time signals

- **Finite length signals.** A finite-length discrete-time signal of length N is just a collection of N complex values, i.e., it is equivalent to a vector $\mathbf{x} \in \mathbb{C}^N$

$$x[n] \quad n = 0, \ldots, N-1 \quad \Longleftrightarrow \quad \mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]$$

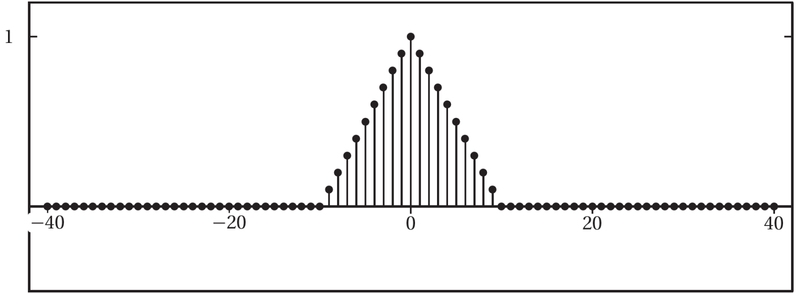
  The vector notation is useful when we want to stress the algorithmic or geometric nature of certain signal processing operations

- **Infinite length signals.** The most general type of discrete-time signal is represented by a generic infinite complex sequence.

- Even if we will always process finite-length signals, studying the asymptotic behavior of algorithms and transformations for infinite sequences is extremely valuable

- **Finite-support signals.** An infinite discrete-time sequence $\bar{x}[n]$ has finite support if its values are zero for all indices outside of an interval; that is, there exist N and $M \in \mathbb{Z}$ such that

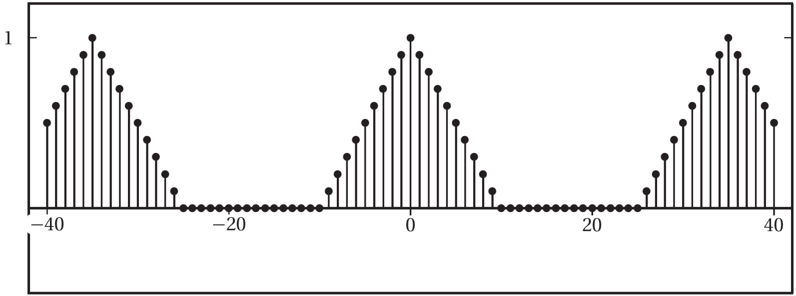
$$\bar{x}[n] = 0 \quad \text{for } n < M \text{ and } N > M + N - 1$$



- **Periodic signals.** A periodic sequence with period N is an infinite length signal for which

$$\tilde{x}[n] = \tilde{x}[n + kN] \qquad k \in \mathbb{Z}$$

An N-periodic sequence is completely defined by its N values over a period



Classes of discrete-time signals

- **Periodization of finite-support signals.** Given a finite-support signal $\bar{x}[n]$ and an integer $N > 0$, we can always formally write

$$\tilde{x}[n] = \sum_{k=-\infty}^{\infty} \bar{x}[n - kN]$$

  The periodic signal $\tilde{x}[n]$ is obtained by superimposing infinite copies of the original signal $\bar{x}[n]$ spaced N samples apart

  - If N is bigger than the size of the support, then the copies in the sum do not overlap
  - if N is smaller than the size of the support then the copies in the sum do overlap
  - The finite-support signal $\bar{x}[n]$ represents one period of the periodic signal $\tilde{x}[n]$

Signals and vector spaces

- Most of the signal processing theory can be usefully cast in terms of vector notation
- Let us consider two finite length discrete-time signals $x[n]$ and $y[n]$, $n = 0, \dots, N-1$, along with their equivalent vector notation $\mathbf{x}, \mathbf{y} \in \mathbb{C}^N$
- **Inner product.** The inner product is defined as

$$(\mathbf{x}, \mathbf{y}) = \sum_{n=0}^{N-1} x^*[n]y[n]$$

- Properties:
  - **Energy.** $(\mathbf{x}, \mathbf{x}) = \|\mathbf{x}\|^2 = \sum_{n=0}^{N-1} |x[n]|^2 = E_x$
  - **Distance.** $\|\mathbf{x} - \mathbf{y}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\mathcal{R}\{(\mathbf{x}, \mathbf{y})\}$
  - **Orthogonality.** $(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} \perp \mathbf{y}$
  - **CBS inequality.** $(\mathbf{x}, \mathbf{y}) \leq \|\mathbf{x}\| \|\mathbf{y}\| = \sqrt{E_x}\sqrt{E_y}$

- Discrete-time signals $\mathbf{x} \in \mathbb{C}^N$ can be expanded over a basis:

$$\mathbf{x} = \sum_{k=0}^{N-1} X_k \mathbf{u}_k$$

  where $\{\mathbf{u}_k\}_{k=0}^{N-1}$ are linearly independent vectors that constitute a basis for $\mathbb{C}^N$

- If the vectors $\{\mathbf{u}_k\}_{k=0}^{N-1}$ are orthogonal, i.e., $(\mathbf{u}_k, \mathbf{u}_l) = \delta[k-l]$, for $k, l = 0, \ldots, N-1$, then the expansion coefficients $\{X_k\}_{k=0}^{N-1}$ can be computed by the simple formula:

$$X_k = \frac{(\mathbf{u}_k, \mathbf{x})}{\|\mathbf{x}\|^2} \qquad k = 0, \ldots, N-1,$$

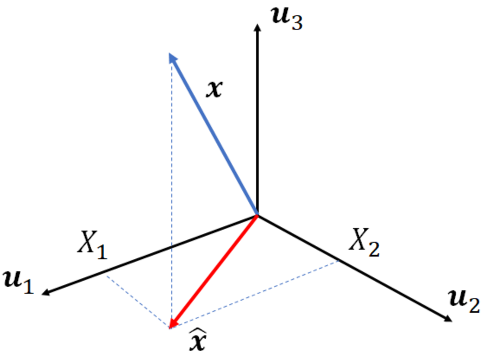
  which represents the (normalized) orthogonal projection of $\mathbf{x}$ over the vectors $\{\mathbf{u}_k\}_{k=0}^{N-1}$

- **Best approximations.** Let us assume that we want to approximate $\mathbf{x} \in \mathbb{C}^N$ by a linear combination of basis vectors of a K-dimensional subspace $\mathcal{P}$, with $K < N$
- Let $\mathcal{P} = \left\{ \mathbf{z} \,|\, \mathbf{z} = \sum_{k=0}^{K-1} Z_k \mathbf{u}_k \right\}$ be the subspace spanned by the basis vectors $\{\mathbf{u}_k\}_{k=0}^{K-1}$
- Let us define the "best approximation" vector $\tilde{\mathbf{x}} = \displaystyle\sum_{k=0}^{K-1} \frac{(\mathbf{u}_k, \mathbf{x})}{\|\mathbf{x}\|^2} \mathbf{u}_k$

Projection Theorem

Letting $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$ be the approximation error, it is easy to show that:

- The error is orthogonal to the approximation, i.e., $\mathbf{e} \perp \tilde{\mathbf{x}}$
- $\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{x} - \mathbf{z}\|$ for all $\mathbf{z} \in \mathcal{P}$, i.e., $\tilde{\mathbf{x}}$ is the best approximation of $\mathbf{x}$ among all the possible vectors belonging to $\mathcal{P}$



- **Example.** The best approximation $\tilde{\mathbf{x}}$ is the orthogonal projection of vector $\mathbf{x}$ over the subspace $\mathcal{P}$ spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$

- In the signal processing jargon, the following equation is called the **analysis formula:**

$$X_k = \frac{(\mathbf{u}_k, \mathbf{x})}{\|\mathbf{x}\|^2} \qquad k = 0, \dots, N-1.$$

  The vector $[X_0, \dots, X_{N-1}]^T \in \mathbb{C}^N$ is an equivalent representation of the signal, which can be seen as a **set of features** extracted by $\mathbf{x}$ using the basis functions $\{\mathbf{u}_k\}_{k=0}^{N-1}$

- On the opposite side, the following equation is called the **synthesis formula:**

$$\mathbf{x} = \sum_{k=0}^{N-1} X_k \mathbf{u}_k,$$

  which builds (i.e., synthesize) the signal $\mathbf{x}$ as a linear combination of the basis functions $\{\mathbf{u}_k\}_{k=0}^{N-1}$, weighted by the coefficients $\{X_k\}_{k=0}^{N-1}$

- **Examples of basis**
  - **Canonical.** $u_k[n] = \delta[n-k] \qquad k, n = 0, \ldots, N-1$
  - **Fourier.** $u_k[n] = e^{j\frac{2\pi}{N}nk} = \cos(\frac{2\pi}{N}nk) + j\sin(\frac{2\pi}{N}nk) \qquad k, n = 0, \ldots, N-1$
  - **Walsh-Hadamard.**

$$u_0[n] = 1, \quad 0 \leq n \leq N-1$$

$$u_1[n] = 1, \quad 0 \leq n < \frac{N-1}{2}; \quad u_1[n] = -1, \quad \frac{N-1}{2} \leq n \leq N-1$$

$$u_2[n] = 1, \quad 0 \leq n < \frac{N-1}{4}\ , \ \frac{3(N-1)}{4} \leq n < N-1\ ; \ u_2[n] = -1, \quad \frac{(N-1)}{4} \leq n < \frac{3(N-1)}{4}$$

and so on...

- Depending on the selected basis functions, different signal representations (i.e., extracted features) can be obtained

- **Question.** How to choose the best basis functions for the signal at hand?

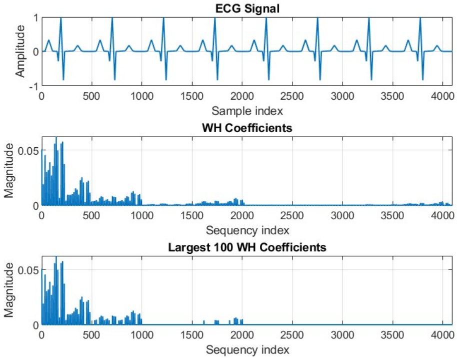
  Amongst competing representations, one should choose the **sparsest one**, i.e. the one requiring the smallest number of components

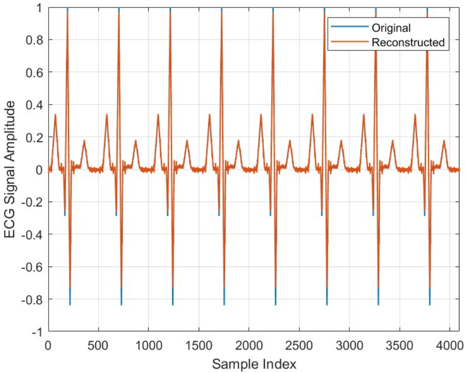
  "Non sunt multiplicanda entia sine necessitate" —*Ockham's razor* (1287–1347)

- Sparse representations are fundamental in SP to improve processing, learning, storing, and transmission of information

Exercise 1: Compression of ECG signals

2 Signals





- **Matlab code:** Exercise1.m

Continuous-time signals: inner product

- Given two signals $x(t)$ and $y(t)$ having finite energy, their inner product is defined as:

$$(\mathbf{x}, \mathbf{y}) = \int_{-\infty}^{\infty} x^*(t) y(t) dt$$

- **Signal energy:** $E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt = \|\mathbf{x}\|^2$

- Given two signals $x(t)$ and $y(t)$ having finite power, their inner product is defined as:

$$(\mathbf{x}, \mathbf{y}) = \frac{1}{\Delta t} \lim_{\Delta t \to \infty} \int_{-\Delta t/2}^{\Delta t/2} x^*(t) y(t) dt$$

- **Signal power:** $P_x = \lim_{\Delta t \to \infty} \int_{-\Delta t/2}^{\Delta t/2} |x(t)|^2 dt = \|\mathbf{x}\|^2$

- Let us consider a continuous-time signal $x(t)$ with $t \in [-T/2; T/2]$
- The inner product is defined as:

$$(\mathbf{u}, \mathbf{x}) = \int_{-T/2}^{T/2} u^*(t)x(t)dt$$

- Considering a set of basis functions $\{u_k(t)\}_{k=0}^{\infty}$ that are linearly independent, orthogonal, and complete, we have the following analysis formula:

$$X_k = \frac{(\mathbf{u}_k, \mathbf{x})}{\|\mathbf{u}_k\|^2} = \frac{\int_{-T/2}^{T/2} u_k^*(t)x(t)dt}{\int_{-T/2}^{T/2} |u_k(t)|^2 dt} \qquad k = 0, 1, 2, \dots$$

- The synthesis formula is given by $\quad x(t) = \sum_{k=0}^{\infty} X_k u_k(t)$

- Let $\tilde{x}_K(t) = \sum_{k=0}^{K-1} X_k u_k(t)$ be the best approximation signal with K coefficients, the completeness property states that:

$$\lim_{K\to\infty} \int_{-T/2}^{T/2} |x(t) - \tilde{x}_K(t)|^2 = 0,$$

i.e., the energy of the approximation error goes to zero as K increases

Dirichlet conditions

Consider a continuous-time signal $x(t)$, $t \in [-T/2, T/2]$, which satisfies:

1. $\int_{-T/2}^{T/2} |x(t)| dt < \infty$
2. the number of discontinuities of $x(t)$ in the definition interval is finite
3. the number of maximum and minimum points of $x(t)$ in the definition interval is finite

- Examples of complete basis for signals satisfying the Dirichlet conditions
  - **Fourier.** $u_k(t) = e^{j2\pi kt/T} \qquad t \in [-T/2; T/2], \quad k \in \mathbb{Z}$
  - **Walsh functions.**

    $u_0(t) = 1, \quad 0 \leq t \leq 1$

    $u_1(t) = 1, \quad 0 \leq t \leq 0.5; \quad u_1(t) = -1, \quad 0.5 \leq t \leq 1$

    $u_2(t) = 1, \quad 0 \leq t \leq 1/4\ ,\ 1/2 \leq t \leq 3/4\ ;\ u_2(t) = -1, \quad 1/4 \leq t < 1/2,\ 3/4 \leq t < 1$

    $\ldots$
  - **Legendre polynomials.** $u_n(t) = \sqrt{\frac{2n+1}{2}}p_n(t) \quad t \in [-1; 1]$

    where $p_0(t) = 1$, $p_n(t) = \dfrac{1}{2^n n!}\dfrac{d^n}{dt^n}(t^2 - 1)^n \quad n = 1, 2, \ldots$

Table of Contents

Fourier transform

3 Spectral analysis

- The Fourier transform of a continuous-time signal defined for $t \in (-\infty, \infty)$ is

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

It is an inner product between $x(t)$ and the complex sinusoid $e^{j2\pi ft}$ with frequency f

- The inverse Fourier transform is defined as

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}dt$$

It synthesizes the signals as a linear combination of infinite sinusoids weighted by the values of $X(f)$

Properties of Fourier Transform

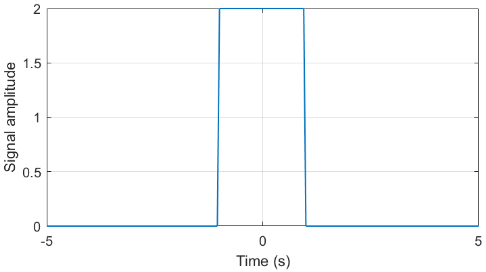
- The function $X(f)$ is referred to as the *spectrum* of the signal
- $X(f)$ shows how much oscillatory behavior at frequency f is contained in the signal $x(t)$; indeed, an inner product is a measure of similarity
- The square magnitude $|X(f)|^2$ is a measure of the signal's energy at the frequency f
- **Parseval relation.** For signals with finite energy, it holds

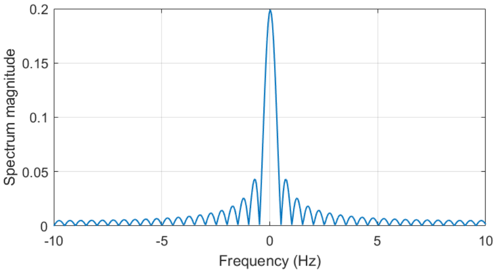
$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df$$

  Thus, the spectrum shows how the global energy of the original signal is distributed in the frequency domain

- A simple example:

$$x(t) = A\mathrm{rect}_T(t) \quad \overset{\mathcal{F}}{\Longrightarrow} \quad X(f) = AT\mathrm{sinc}(\pi Tf) = AT\frac{\sin(\pi Tf)}{\pi Tf}$$





- **Area:**

$$X(0) = \int_{-\infty}^{\infty} x(t)dt \qquad x(0) = \int_{-\infty}^{\infty} x(f)df$$

- **Conjugate symmetry:** If $x(t)$ is real, then

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

- **Delay:** $\mathcal{F}\{x(t-\tau)\} = X(f)e^{-j2\pi f\tau}$
- **Modulation:** $\mathcal{F}\{x(t)e^{j2\pi f_0 t}\} = X(f-f_0)$
- **Duality:** If $X(f) = \mathcal{F}\{x(t)\}$ then $\mathcal{F}\{X(f)\} = x(-t)$

- **Covolution:**

$$z(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau \quad \rightarrow \quad Z(f) = X(f)Y(f)$$

$$z(t) = x(t)y(t) \quad \rightarrow \quad Z(f) = \int_{-\infty}^{\infty} X(\nu)Y(f-\nu)d\nu$$

- **Correlation:**

$$z(t) = \int_{-\infty}^{\infty} x^*(\tau)y(t+\tau)d\tau \quad \rightarrow \quad Z(f) = X^*(f)Y(f)$$

- **Change of scale:**

$$\mathcal{F}\{x(\alpha t)\} = \frac{1}{|\alpha|}X\left(\frac{f}{\alpha}\right)$$

Properties of Fourier transform

- **Periodization:** Let $g(t)$ be a finite-length signal of duration T. The periodization of $g(t)$ is given by (Every periodic signal can be written in this form):

$$x(t) = \sum_{k=-\infty}^{\infty} g(t - kT)$$

- We have that

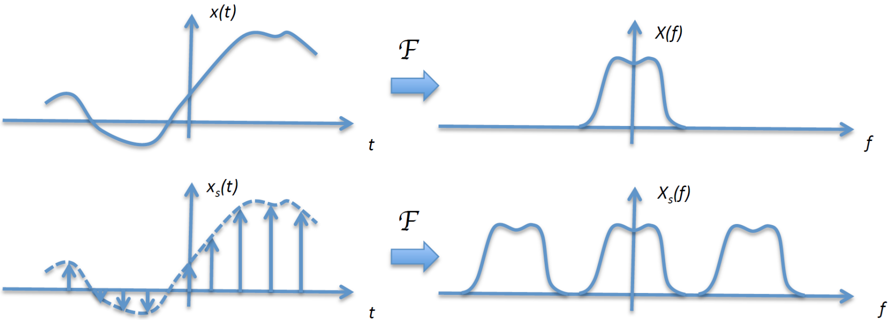
$$X(f) = \mathcal{F}\left\{ \sum_{k=-\infty}^{\infty} g(t - kT) \right\} = \frac{1}{T} \sum_{k=-\infty}^{\infty} G\left(\frac{k}{T}\right) \delta\left(f - \frac{k}{T}\right)$$

  with $G(f) = \mathcal{F}\{g(t)\}$, and $\delta(t)$ being the Dirac delta function

- The spectrum of a periodic signal is composed of pulses centered in the harmonic frequencies (i.e., integer multiples of the fundamental frequency $1/T$) of the signal

- **Sampling:**

$$x_s(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t-kT) = \sum_{k=-\infty}^{\infty} x(kT)\delta(t-kT) \quad \Rightarrow \quad X_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{T}\right)$$



From continuous-time SP to discrete-time SP

- Classical signal processing theory is developed in continuous-time, but real processing is typically performed by computers on discrete-time signals

- Data can be discrete by nature, or achieved through the discretization (e.g., sampling) a continuous-time signal $x(t)$, i.e., observing $x(t)$ at given time instants nT; we say that $x[n] = x(nT)$.

- **Question:** How can we extend Fourier analysis to discrete-time signals?

Discrete Fourier basis

- The discrete Fourier basis functions are given by the complex exponentials:

$$w_k[n] = e^{j\frac{2\pi}{N}kn} = \cos\left(\frac{2\pi}{N}kn\right) + j\sin\left(\frac{2\pi}{N}kn\right) \qquad n,k = 0,\dots,N-1$$

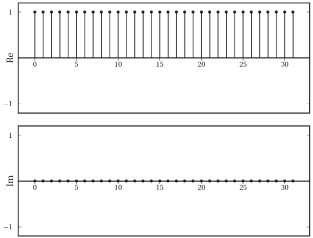
- **Vector representation.** The signals $w_k[n]$ can be equivalently cast as the vectors

$$\mathbf{w}_k = \left[1 \; e^{j\frac{2\pi}{N}k} \; e^{j\frac{2\pi}{N}2k} \; \dots \; e^{j\frac{2\pi}{N}(N-1)k}\right]^T \in \mathbb{C}^N \quad k = 0,\dots,N-1$$

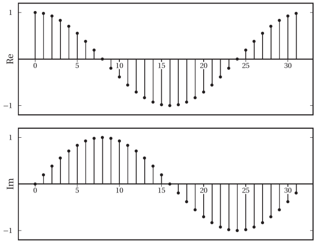
- **Orthogonality.** Taking the inner product between two basis vectors, we get

$$(\mathbf{w}_k, \mathbf{w}_l) = \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(l-k)n} = \begin{cases} N & \text{for } k = l \\ \frac{1-e^{j\frac{2\pi}{N}(l-k)N}}{1-e^{j\frac{2\pi}{N}(l-k)}} = 0 & \text{for } k \neq l \end{cases}$$

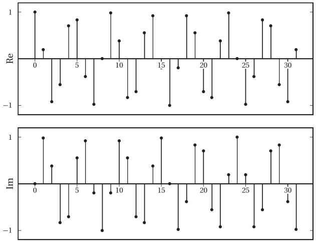
- **Examples of Fourier basis functions.**



Basis vector $\mathbf{w}_0 \in \mathbb{C}^{32}$



Basis vector $\mathbf{w}_1 \in \mathbb{C}^{32}$



Basis vector $\mathbf{w}_7 \in \mathbb{C}^{32}$

- The Discrete Fourier transform (DFT) of a (discrete-time) signal is an alternative representation of the data, which lives in the *(discrete) frequency domain*

- **Discrete Fourier Transform.** (Analysis formula)

$$X[k] = (\mathbf{w}_k, \mathbf{x}) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \qquad k = 0, \dots, N-1$$

- **Inverse Discrete Fourier Transform.** (Synthesis formula)

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn} \qquad k = 0, \dots, N-1$$

Discrete Fourier Transform in matrix form

- Let $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_{N-1}] \in \mathbb{C}^{N \times N}$ be the matrix collecting the Fourier basis vectors in its columns; this matrix is often call the DFT matrix

- **Matrix Form.** The DFT can be conveniently recast in compact matrix form as

$$\hat{\mathbf{x}} = \mathbf{W}^H \mathbf{x}$$

  where $\hat{\mathbf{x}} = [X[1], \ldots, X[N-1]]^T$, and the superscript ^{H} denotes the hermitian operator, i.e., complex conjiugate transposition

- The IDFT matrix form readily follows as

$$\mathbf{x} = \frac{1}{N} \mathbf{W} \hat{\mathbf{x}}$$

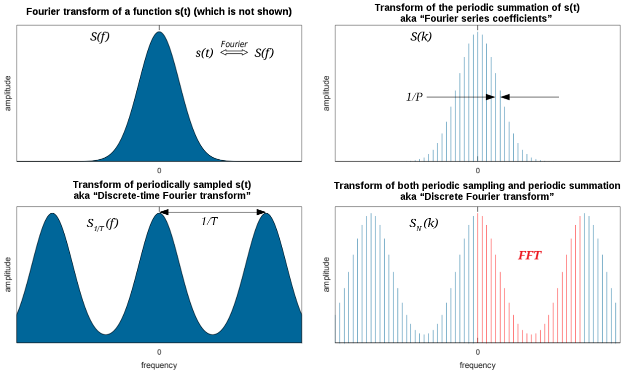
The Fast Fourier Transform

- The Fast Fourier transform, or FFT, is not another type of transform but simply the name of an efficient algorithm to compute the DFT
- Considering the matrix DFT form

$$\hat{\mathbf{x}} = \mathbf{W}^H \mathbf{x},$$

  the computation of the DFT requires a number of operations on the order of N^2
- The FFT algorithm exploits the highly structured nature of $\mathbf{W}$ to reduce the number of operations to $N \log(N)$
- The FFT algorithm is particularly efficient for data lengths which are a power of 2

Relation between FT and DFT



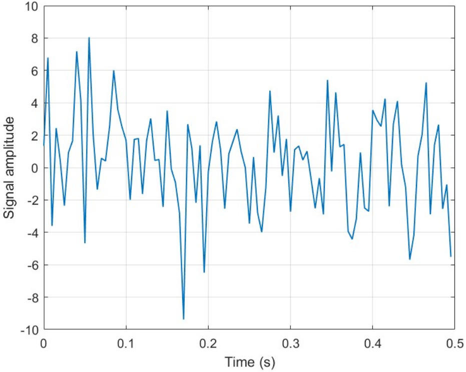
- The DFT is a discrete version (i.e. samples) of the periodization of the continuous-time signal spectrum, where samples are collected over a single period

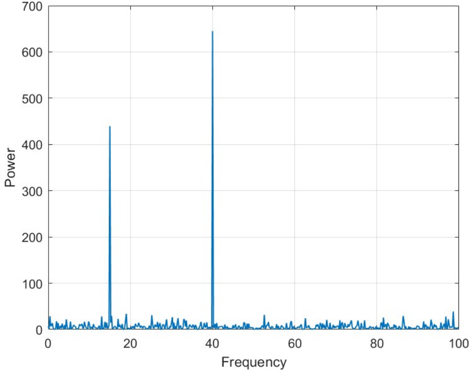
Properties

- The coefficients $X[k]$ are referred to as the *spectrum* of the signal
- Each $X[k]$ shows how much oscillatory behavior at frequency $(2\pi/N)k$, is contained in the signal; indeed, an inner product is a measure of similarity
- The square magnitude $|X[k]|^2$ is a measure (up to a scale factor N) of the signal's energy at the frequency $(2\pi/N)k$
- **Parseval relation.** For the DFT, it holds $\|\mathbf{x}\|^2 = \frac{1}{N}\|\hat{\mathbf{x}}\|^2$. Thus, the spectrum shows how the global energy of the original signal is distributed in the frequency domain

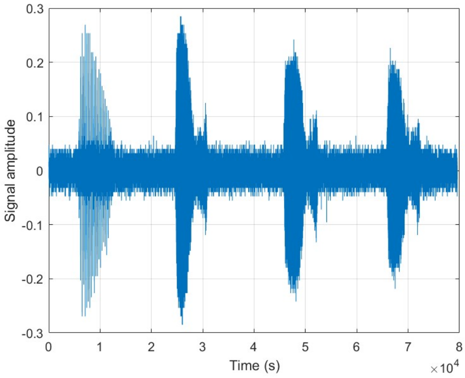
Exercise 2: Detecting periodic trends embedded in noise

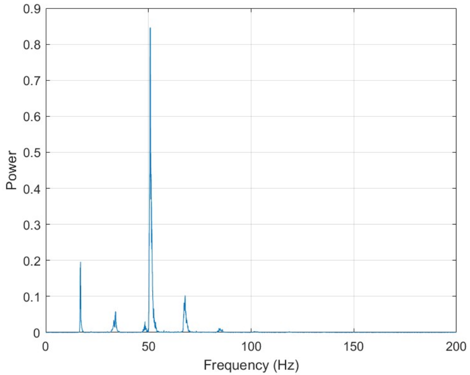
3 Spectral analysis





- **Matlab code:** Exercise2.m

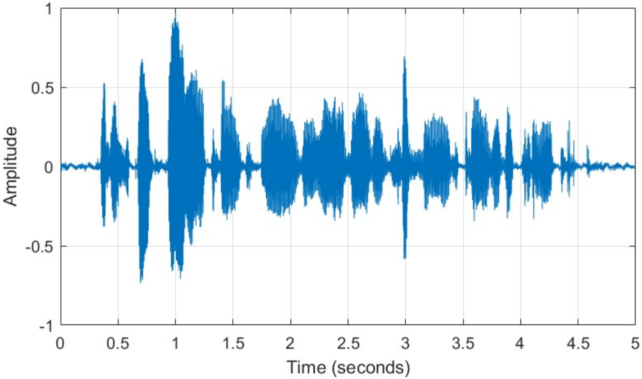




- **Matlab code:** Exercise3.m

Exercise 4: Spectrum of speech signals

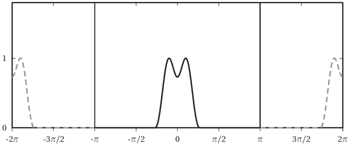
3 Spectral analysis



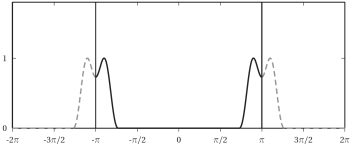
- Vowels and consonants have a very different spectrum!
- **Matlab code:** Exercise4.m

Spectrum classification

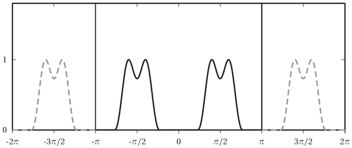
- Analyzing the spectrum, one can obtain, at a glance, the fundamental information, required to characterize and classify a signal in the frequency domain
- **Magnitude.** The magnitude of a signal's spectrum, obtained by the Fourier transform, represents the energy distribution in frequency for the signal
- It is customary to broadly classify discrete-time signals into three classes



Lowpass spectrum



Highpass spectrum



Bandpass spectrum

Phase spectrum

- **Phase.** The phase of each spectrum coefficient, indicated by $\angle X[k]$, represents the relative alignment of each complex exponential with the oscillation at frequency $(2\pi/N)k$ contained in the signal

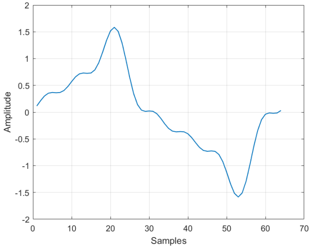
- This alignment determines the shape of the signal in the discrete-time domain

- **Example.** Consider the 64-periodic discrete-time signal

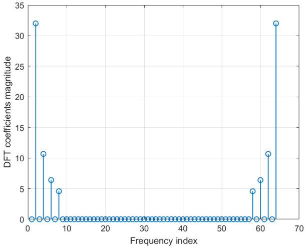
$$x[n] = \sum_{i=0}^{3} \frac{1}{2i+1} \sin\left(\frac{2\pi}{64}(2i+1)n + \phi_i\right)$$

- The magnitude of the spectrum is independent of the phases ϕ_i, $i = 0, 1, 2, 3$, but the signal has very different shape in the time domain

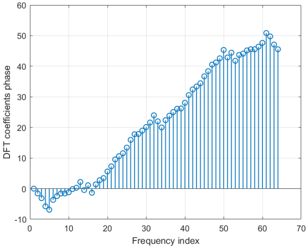
- **Case 1.** $\phi_i = 0$ for $i = 0, 1, 2, 3$.



Signal in time

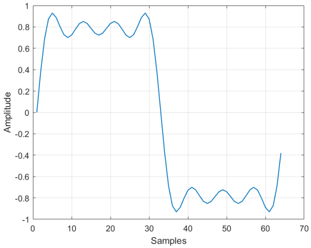


Magnitude spectrum

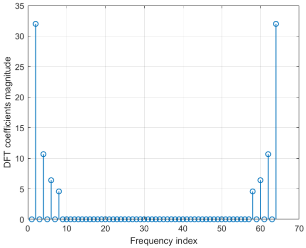


Phase spectrum

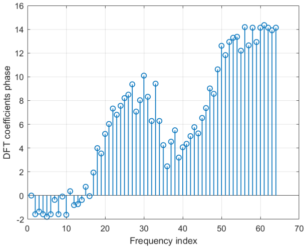
- **Case 2.** $\phi_i = 2\pi i/3$ for $i = 0, 1, 2, 3$.



Signal in time



Magnitude spectrum



Phase spectrum

- Let us consider a piece of music. It is obvious that the melodic information is determined not only by the pitch values but also by their duration and order
- If we take a global Fourier Transform of the entire musical piece, we have a comprehensive representation of the frequency content of the piece: in the resulting spectrum there is information about the frequency of each played note
- The time information, however, that is the information pertaining to the order in which the notes are played, is completely hidden by the spectral representation
- **Question.** There exists a time-frequency representation of a signal, in which both time and frequency information are readily apparent?

The spectrogram

- The simplest time-frequency transformation is called the **spectrogram**
- The basic idea involves splitting the signal into small consecutive (and possibly overlapping) length-N pieces and computing the DFT of each

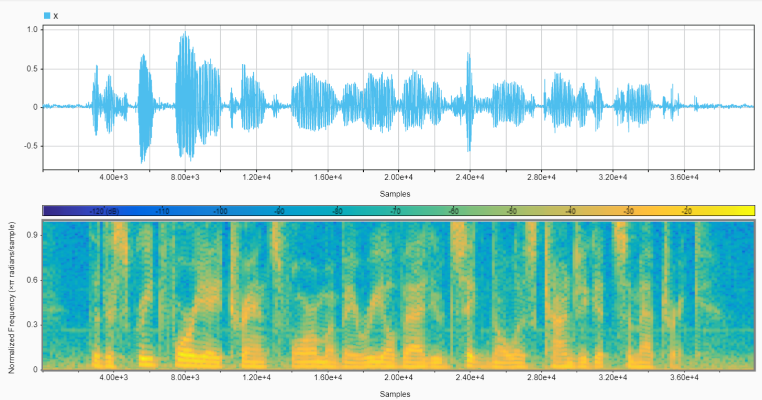
  Short-time Fourier Transform $$S[k,m] = \sum_{n=0}^{N-1} x[mM+n] e^{-j\frac{2\pi}{N}nk}$$

  where M, $1 \leq M \leq N$ controls the overlap between segments
- In matrix notation we have $\mathbf{S} = \mathbf{W}_N \begin{bmatrix} x[0] & x[M] & x[2M] & \dots \\ x[1] & x[M+1] & x[2M+1] & \dots \\ \vdots & \vdots & \vdots & \dots \\ x[N-1] & x[M+N-1] & x[L] & \dots \end{bmatrix}$

  The spectrogram is an $N \times \lfloor L/M \rfloor$ matrix, where L is the total length of the signal $x[n]$

Exercise 5: Spectrogram of speech signals

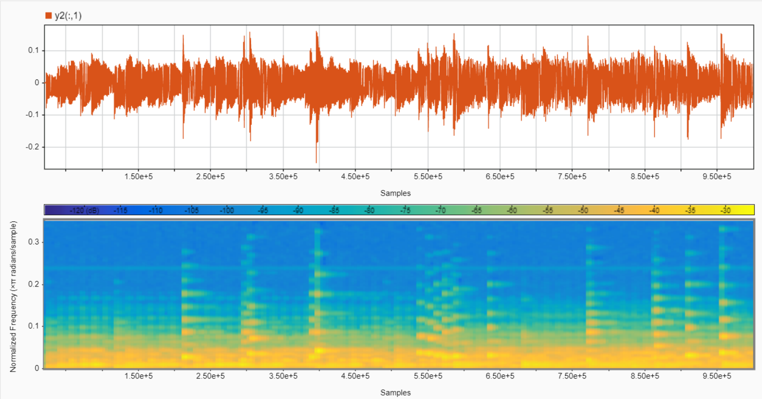
3 Spectral analysis



- **Matlab code:** Exercise5.m

Exercise 6: Spectrogram of piano music

3 Spectral analysis



- **Matlab code:** Exercise6.m

Time-frequency localization of signals

- Each of the columns of $\boldsymbol{S}$ represents the local spectrum for a time interval of length N
- We can therefore say that the time resolution of the spectrogram is N samples
- The frequency resolution of the spectrogram is $2\pi/N$
- If we want to increase the frequency resolution we need to take longer windows but in so doing, we lose the time localization of the spectrogram
- Likewise, if we want to achieve a fine resolution in time, the corresponding spectral information for each "time slice" will be very coarse
- The above problem is actually a particular instance of a general **uncertainty principle for time-frequency analysis**

Wavelets

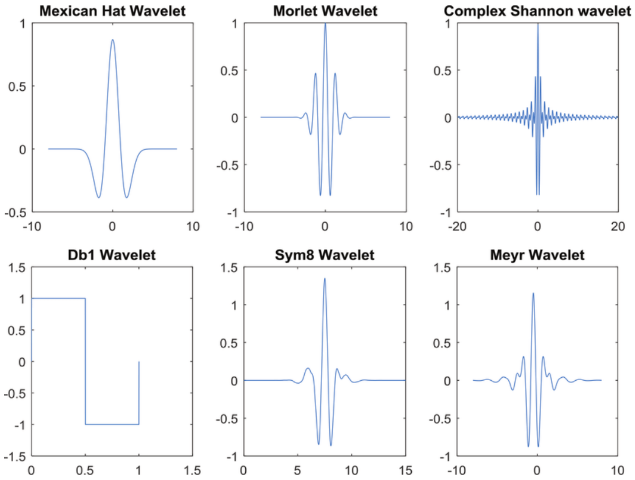
- The STFT enables signal analysis using a fixed time and frequency resolution
- If the signal is composed of small bursts associated with long quasi-stationary components, the components can be analyzed with good time resolution or frequency resolution, but not both
- To overcome this limitation, we should perform a multi-resolution analysis that uses short windows at high frequencies and long windows at low frequencies
- To this aim, we define the **wavelets**:

$$h_{a,\tau}(t) = \frac{1}{\sqrt{a}} h\left(\frac{t-\tau}{a}\right)$$

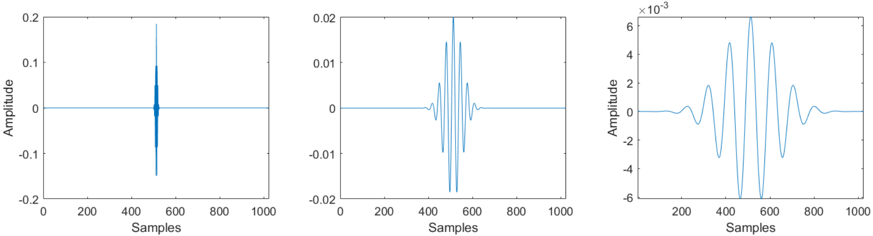
where h is a prototype function (mother wavelet), a is a scale factor, and τ is a delay

Wavelets

3 Spectral analysis



- **Scaling.**



- Small scale $0 < a < 1 \rightarrow$ compressed wavelet $\rightarrow$ fast-varying details $\rightarrow$ High frequency
- Large scale $a > 1 \rightarrow$ stretched wavelet $\rightarrow$ slow-varying details $\rightarrow$ Low frequency

Continuous wavelet transform

- Wavelets represent a family of basis functions
- The **continuous wavelet transform (CWT)** is given by

$$\mathrm{CWT}_x(\tau, a) = \int_{-\infty}^{\infty} x(t) h^*_{a,\tau}(t) dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) h^* \left(\frac{t - \tau}{a} \right) dt$$

  It represents an inner product between the signal and the wavelet basis function characterized by a and τ

- The synthesis formula reads as:

$$x(t) = c \int_{a>0} \int_{-\infty}^{\infty} \mathrm{CWT}_x(\tau, a) h^*_{a,\tau}(t) \frac{da d\tau}{a^2}$$

The scalogram

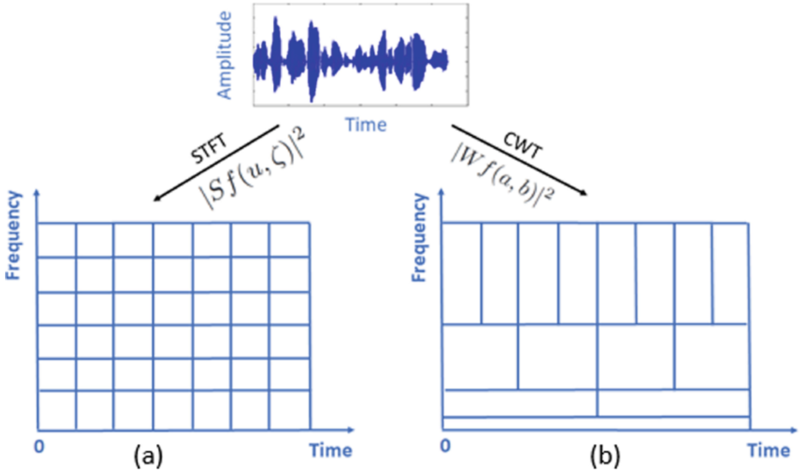
- The squared modulus of the CWT, i.e., $|\mathrm{CWT}_x(\tau, a)|^2$, is called **scalogram**
- Since the CWT behaves like an orthonormal basis decomposition, it preserves energy

$$\int_{a>0} \int_{-\infty}^{\infty} |\mathrm{CWT}_x(\tau, a)|^2 \frac{da d\tau}{a^2} = E_x$$

- Similarly to the spectrogram, the scalogram describes how the signal energy is distributed over the time-scale(frequency) domain
- However, in contrast to the spectrogram, the energy of the signal is here distributed with different resolutions

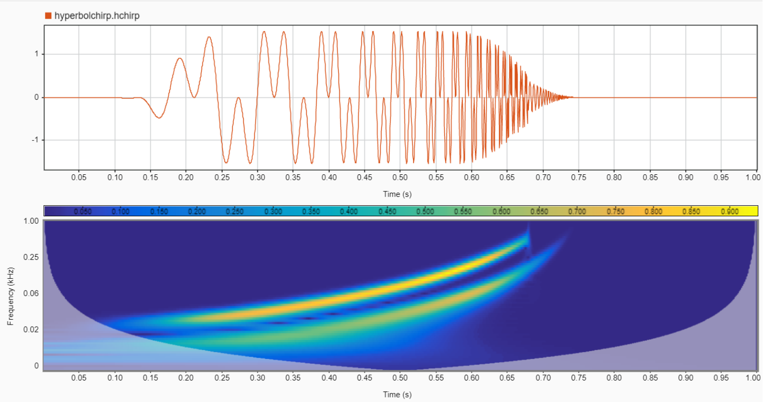
The scalogram

3 Spectral analysis



Exercise 7: Scalogram of chirp signals

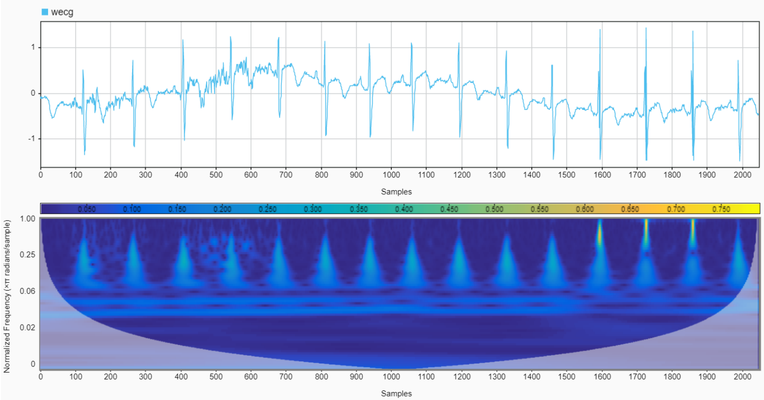
3 Spectral analysis



- **Matlab code:** Exercise7.m

Exercise 8: Scalogram of ECG signals

3 Spectral analysis



- **Matlab code:** Exercise8.m

2D Discrete Fourier transform

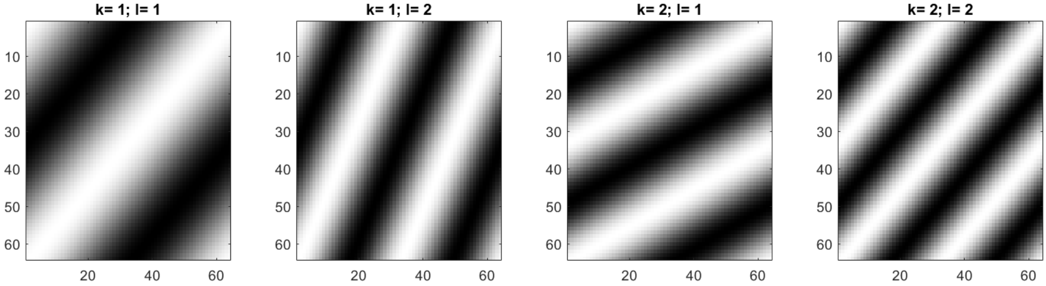
- Fourier analysis can be extended to 2D signals (e.g., images) $X \in \mathbb{C}^{M\times N}$
- The 2D discrete Fourier transform, i.e., the analysis formula, reads as:

$$\hat{X}[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X[m,n] e^{-j2\pi\left(\frac{k}{M}m+\frac{l}{N}n\right)}$$

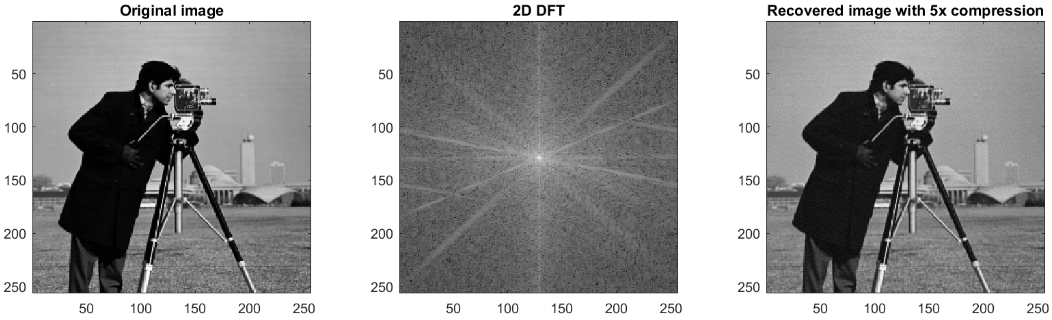
- The inverse 2D discrete fourier transform, i.e., the synthesis formula is given by

$$X[m,n] = \frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \hat{X}[k,l] e^{j2\pi\left(\frac{k}{M}m+\frac{l}{N}n\right)}$$

  It represents the signal expansion over the basis composed by 2D complex sinusoids



- Some examples of (real) 2D sinusoids involved in the 2D DFT



- **Matlab code:** Exercise9.m

Table of Contents

- Signal compression is the art and science to reduce the amount of data required for effective signal representation

- It is one of the most useful and commercially successful technologies in the field of digital signal processing and communications

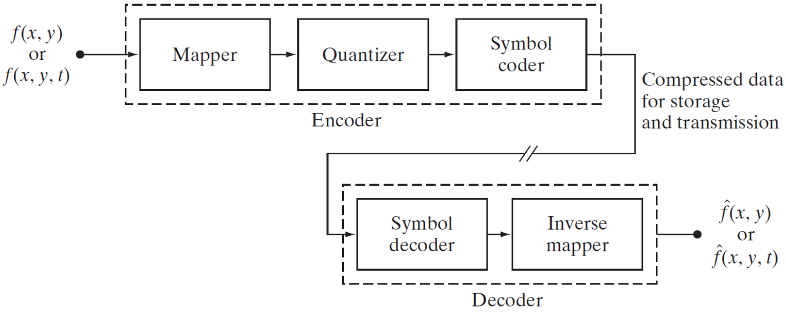
- **Example:** Two-hour standard definition (SD) television movie using bit pixel arrays

$$30\,\frac{\text{frame}}{\text{sec}} \times (720 \times 480)\,\frac{\text{pixels}}{\text{frame}} \times 3\,\frac{\text{bytes}}{\text{pixel}} \times (2 \times 3600)\,\text{sec} = 224\,GB$$

A 26.3 compression factor is needed to store a 2 hours SD video into an 8.5 GB DVD support!

- Various amounts of data can be used to represent the same amount of information. Representations that contain irrelevant or repeated information are said redundant
- Examples of data redundancies in signals:
  - **Spatial and temporal redundancy.** In images, neighbor pixels are correlated spatially, and thus information is unnecessarily replicated in the representations of the correlated pixels. In a video sequence, temporally correlated pixels over frames also duplicate information.
  - **Irrelevant information.** Often, data contain information that is ignored by the human senses (e.g., psycho-acoustics, visual system, etc.)
- **Compression ratio.** $C = \dfrac{b}{b'}$ where b and b' are the number of bits used in two representations of the same information

- We consider image compression, since it provides a generalization for 1D signals (e.g., audio) and a very good starting point for time-varying 2D signals (i.e., videos)



Block scheme of an image compression system

Image compression

- The encoder maps the input image to a compressed representation, removing the redundancies through a series of three independent operations
  - The **mapper** transforms $f(x, y)$ into a (usually non-visual) format designed to reduce spatial and temporal redundancy
  - The **quantizer** reduces the accuracy of the mapper's output in accordance with a pre-established fidelity criterion. The goal is to keep irrelevant information out of the compressed representation
  - The **symbol coder** generates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code
- The decoder contains only two components: a symbol decoder and an inverse mapper. They perform the inverse operations of the encoder's symbol encoder and mapper

Image compression standards

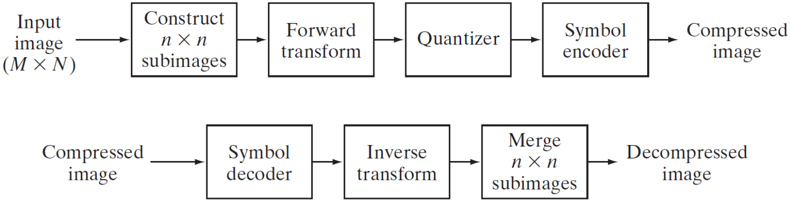
4 Data Compression

Image Compression Standards, Formats, and Containers

- **Still Image**
  - **Binary**
    - CCITT Group 3
    - CCITT Group 4
    - JBIG (or JBIG1)
    - JBIG2
    - TIFF
  - **Continuous Tone**
    - JPEG
    - JPEG-LS
    - JPEG-2000
    - BMP
    - GIF
    - PDF
    - PNG
    - TIFF
- **Video**
  - DV
  - H.261
  - H.262
  - H.263
  - H.264
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - MPEG-4 AVC
  - AVS
  - HDV
  - M-JPEG
  - QuickTime
  - VC-1 (or WMV9)

Block transform coding

4 Data Compression



- An $M \times N$ input image is divided into subimages of size $n \times n$ which are then transformed to generate MN/n^2 subimage transform arrays, each of size $n \times n$
- The quantization stage then selectively eliminates the coefficients that carry the least amount of information in a predefined sense
- The encoding process terminates by coding (normally using a variable-length code) the quantized coefficients

- Consider an $n \times n$ sub-image $g(x, y)$, its forward transform can be generally cast as:

$$T(u, v) = \sum_{x=0}^{n-1} \sum_{x=0}^{n-1} g(x, y) r(x, y, u, v) \qquad u, v = 0, \ldots, n-1$$

- Given $T(u, v)$, $g(x, y)$ can be obtained through the generalized inverse transform:

$$g(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) s(x, y, u, v) \qquad x, y = 0, \ldots, n-1$$

- $r(x, y, u, v)$ and $s(x, y, u, v)$ are called forward and inverse transformation kernels
- $T(u, v)$ are the representation coefficients of $g(x, y)$ w.r.t. the basis functions $s(x, y, u, v)$

Forward and inverse transforms

- The forward and inverse transformation kernels determine the type of transform that is computed and the overall computational complexity and reconstruction error
- **Examples of kernels:**
  - *Fourier transform:* $r(x,y,u,v) = e^{-j2\pi(ux+vy)/n}$ $s(x,y,u,v) = \frac{1}{n^2} e^{j2\pi(ux+vy)/n}$
  - *Discrete cosine transform*

$$r(x,y,u,v) = s(x,y,u,v) = \alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

where $\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u = 0 \\ \sqrt{\frac{2}{n}} & \text{for } u = 1,\dots,n-1 \end{cases}$

  - *Walsh-Hadamard transform*

- Let us consider the compact matrix notation

$$\mathbf{G} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u,v)\mathbf{S}_{u,v}$$

where $\mathbf{S}_{u,v} = \{s(x,y,u,v)\}_{x,y=0}^{n-1}$ are the $n \times n$ inverse kernels (i.e., basis images)

- If we now define a transform coefficient masking function

$$\chi(u,v) = \begin{cases} 0 & \text{if } T(u,v) \text{ satisfies a specified truncation criterion} \\ 1 & \text{otherwise} \end{cases}$$

an approximation of G can be obtained from the truncated expansion

$$\hat{\mathbf{G}} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} \chi(u,v) T(u,v)\mathbf{S}_{u,v}$$

Quantization of transform coefficients

- The reconstruction error associated with the truncated series expansion, i.e., $\mathbb{E}\|\mathbf{G} - \hat{\mathbf{G}}\|^2$, is a function of the number and relative importance of the discarded transform coefficients

- **Question:** How do we select the transform coefficient masking function $\chi(u, v)$?

  There are two popular approaches:

  - *Zonal coding*: It is based on the information theory concept that coefficients of maximum variance carry the most image information and should be retained in the coding process

  - *Threshold coding*: The underlying concept is that, for any subimage, the transform coefficients of largest magnitude make the most significant contribution to reconstructed subimage quality

Zonal coding

4 Data Compression

- A zonal mask is constructed by placing a 1 in the locations of maximum variance and a 0 in all other locations
- Coefficients of maximum variance are located around the origin of an image transform

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Zonal mask (left) and bit allocation (right)

Threshold coding

4 Data Compression

- The transform coefficients of largest magnitude make the most significant contribution to reconstructed subimage quality
- The elements of $\chi(u, v)T(u, v)$ normally are reordered (in a predefined manner) to form a 1-D, run-length coded sequence

1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Threshold mask (left) and coefficients ordering sequence (right)

Threshold coding

- There are three basic ways to threshold a transformed subimage or, stated differently, to create a subimage threshold masking function
  - A single global threshold can be applied to all subimages
  - a different threshold can be used for each subimage
  - the threshold can be varied as a function of the location of each coefficient
- In the first approach, the level of compression differs from image to image, depending on the number of coefficients that exceed the global threshold
- In the second approach, called N-largest coding, the same number of coefficients is discarded for each subimage

- The third technique results in a variable code rate and offers the advantage that thresholding and quantization can be combined by replacing $\chi(u,v)T(u,v)$ with

$$\hat{T}(u,v) = \text{round}\left[\frac{T(u,v)}{Z(u,v)}\right]$$

where $Z(u,v)$ is an element of the transform normalization array

- $\hat{T}(u,v)$ is then multiplied by $Z(u,v)$ as:

$$\dot{T}(u,v) = \hat{T}(u,v)Z(u,v)$$

- The inverse transform of $\dot{T}(u,v)$ yields the decompressed subimage approximation

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

A typical notmalization matrix Z

JPEG: Joint Photographic Experts Group

- JPEG is one of the most popular and comprehensive image compression standards
- The compression itself is performed in three sequential steps: DCT computation, quantization, and variable-length code assignment
- The image is first subdivided into pixel blocks of size which are processed left to right, top to bottom
- The 64 pixels are level-shifted by subtracting the number of intensity levels
- The 2-D discrete cosine transform of the block is then computed, quantized, and reordered, using the zigzag pattern to form a 1-D sequence of quantized coefficients
- The nonzero coefficients are coded using a variable-length code

Exercise 15: JPEG compression

original image

Z = Q50 ; Compression ratio = 9.2

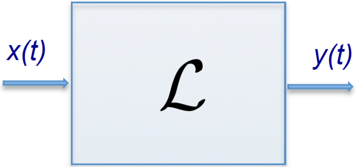
Z = 2*Q50 ; Compression ratio = 14.2

- **Matlab code:** Exercise15.m

Table of Contents

Linear and time-invariant systems

5 Linear processing



- **System:** mapping from an input signal to an output signal $y(t) = \mathcal{L}\{x(t)\}$
- The system is **linear** if

$$\mathcal{L}\{a_1 x_1(t) + a_2 x_2(t)\} = a y_1(t) + a_2 y_2(t) \quad \text{for all } x_1(t), x_2(t), a_1, a_2$$

- The system is **time-invariant** if it satisfies the property

$$\text{if } \; y(t) = \mathcal{L}\{x(t)\} \;\; \Rightarrow \;\; \mathcal{L}\{x(t-t_0)\} = y(t-t_0) \quad \text{for all } x(t), t_0$$

Impulse response

- If a system is linear and time-invariant (LTI), the output can always be written as a function of the input through a convolution operation

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

  where $h(t)$ is the **impulse response**, i.e., the output of the system when the input is a Dirac pulse centered at $t = 0$

$$h(t) = \mathcal{L}\{\delta(t)\}$$

- Once we know the impulse response of an LTI system, we can compute its output with respect to any input, even if we do not know how the system is actually done

Frequency response

- LTI systems can be analyzed in the frequency domain
- **Frequency response:** It is the Fourier transform of the impulse response

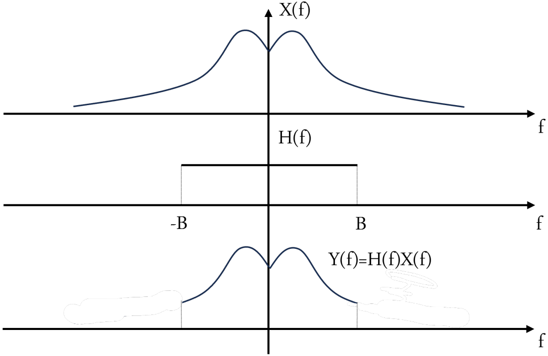
$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt$$

- If the input signal admits a Fourier Transform, the spectrum of the output is

$$Y(f) = X(f)H(f)$$

- The frequency behavior of LTI systems is completely described by the frequency response, which act in a very simple way on the spectrum of the input signal

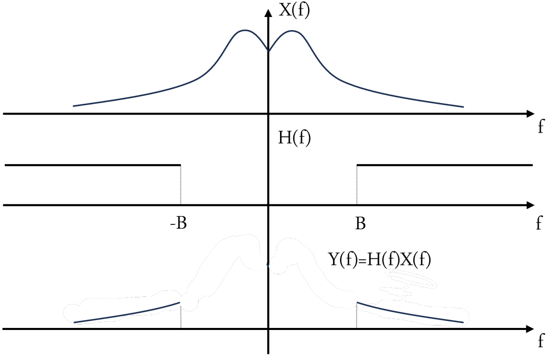
- **Low-pass filter.** It leaves low frequency components unaltered (up to B Hz), while setting to zero high frequencies



$$H(f) = rect_{2B}(f)$$

$$h(t) = 2B\,\text{sinc}(2\pi Bt)$$

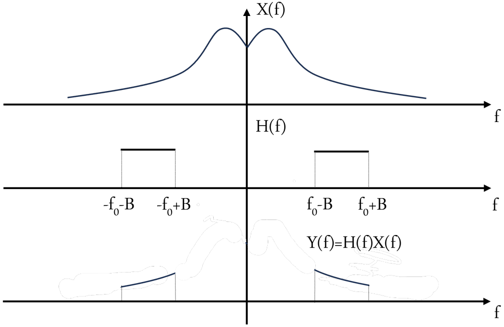
- **High-pass filter.** It leaves high frequency components unaltered (upper than B Hz), while setting to zero low frequencies



$$H(f) = 1 - rect_{2B}(f)$$

$$h(t) = \delta(t) - 2B\,\mathrm{sinc}(2\pi Bt)$$

- **Band-pass filter.** It leaves frequency components unaltered in the band $[f_0 - B, f_0 + B]$, while setting to zero the others



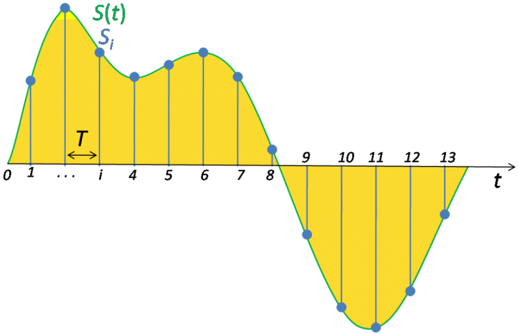
$$H(f) = rect_{2B}(f - f_0) + rect_{2B}(f + f_0)$$

$$h(t) = 4B\,\mathrm{sinc}(2\pi Bt)\cos(2\pi Bt)$$

From continuous to discrete-time

- Real signals often describe continuous-time variation of a physical quantity; in such a case, they are called analog
  - Audio signals are continuous-time analog pressure waves
  - Images are continuous-space analog scenes of light
  - Radio signals are analog electromagnetic waves that propagate over time and space
  - And so on...
- To process data in a computer, analog signals must be necessarily discretized
- The discretization of an analog signal involves both the domain (i.e., sampling) and the codomain (i.e., quantization)

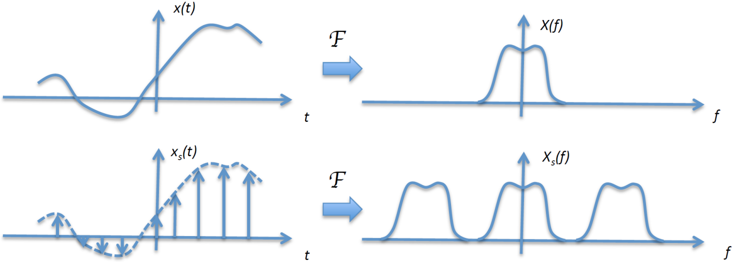
- The process of domain discretization is called **sampling**
- Sampling a continuous-time signal $x(t)$ means observing it at given time instants nT, where T is called sampling period. This generates a discrete-time signal $x[n] = x(nT)$



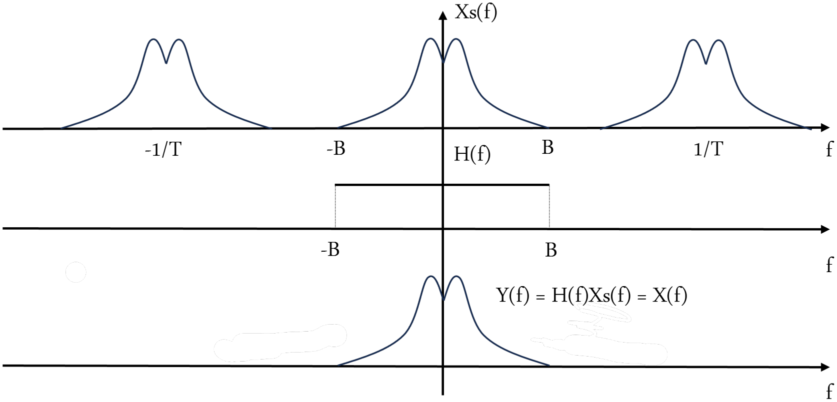
- **Question:** Is it possible to recover the analog signal $x(t)$ from its samples $x(nT)$?

- By the properties of the FT, sampling determines a periodization of the signal spectrum

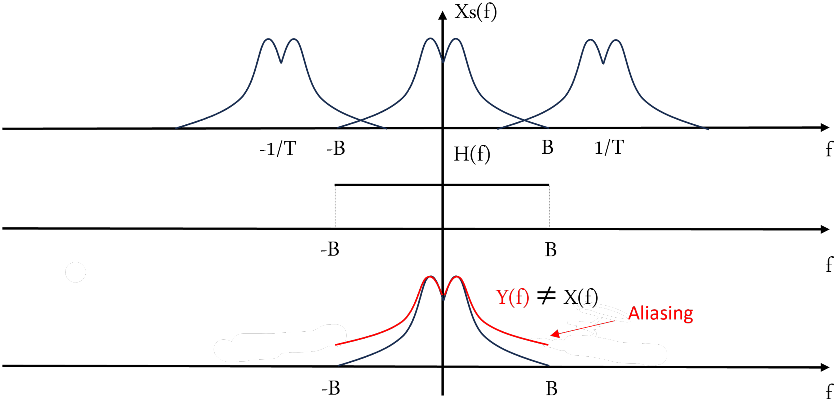
$$x_s(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT) = \sum_{k=-\infty}^{\infty} x(kT)\delta(t - kT) \quad \Rightarrow \quad X_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{T}\right)$$



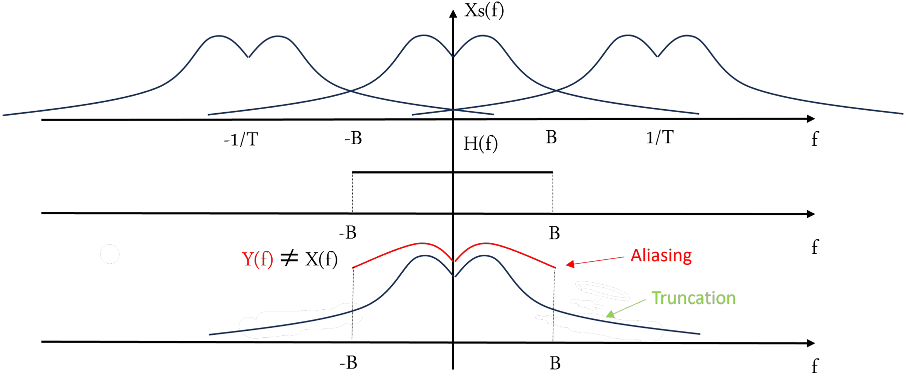
- **Case 1:** The signal $x(t)$ is B-band-limited and $T \leq \dfrac{1}{2B}$



- **Case 2:** The signal $x(t)$ is B-band-limited and $T > \dfrac{1}{2B}$



- **Case 3:** The signal $x(t)$ is not band-limited



Sampling theorem

If a signal $x(t)$ contains no frequencies higher than B Hz, it is completely determined by its samples collected $T = 1/2B$ seconds apart. The interpolation formula is given by:

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \mathrm{sinc}\left(2\pi B\left(t - \frac{n}{2B}\right)\right)$$

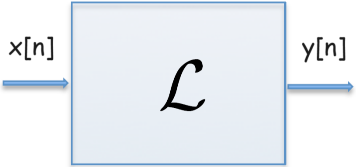
Proof. From case 1, we have

$$x(t) = x_s(t) * h_{LP}(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \delta\left(t - \frac{n}{2B}\right) * \mathrm{sinc}(2\pi Bt)$$

$$= \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \mathrm{sinc}\left(2\pi B\left(t - \frac{n}{2B}\right)\right)$$

Discrete-time LTI systems

5 Linear processing



- **System:** mapping from an input signal to an output signal $y[n] = \mathcal{L}\{x[n]\}$
- The system is **linear** if

$$\mathcal{L}\{a_1x_1[n] + a_2x_2[n]\} = ay_1[n] + a_2y_2[n] \quad \text{for all } x_1[n], x_2[n], a_1, a_2$$

- The system is **time-invariant** if it satisfies the property

$$\text{if } \; y[n] = \mathcal{L}\{x[n]\} \quad \Rightarrow \quad \mathcal{L}\{x[n-k]\} = y[n-k] \quad \text{for all } x[n], k$$

- The output of a discrete-time LTI system can always be written as a function of the input through a linear convolution operation

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

wnere $h[n] = \mathcal{L}\{\delta[n]\}$ is the discrete impulse response of the system

- If the input signal and the impulse response are defined over a finite number of points N, their DFTs are

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad k = 0, \dots, N-1$$

$$H[k] = \sum_{n=0}^{N-1} h[n]e^{-j2\pi kn/N} \qquad k = 0, \dots, N-1$$

- Frequency behavior of the discrete-time LTI system:

$$Y[k] = H[k]X[k] \qquad k = 0, \ldots, N-1$$

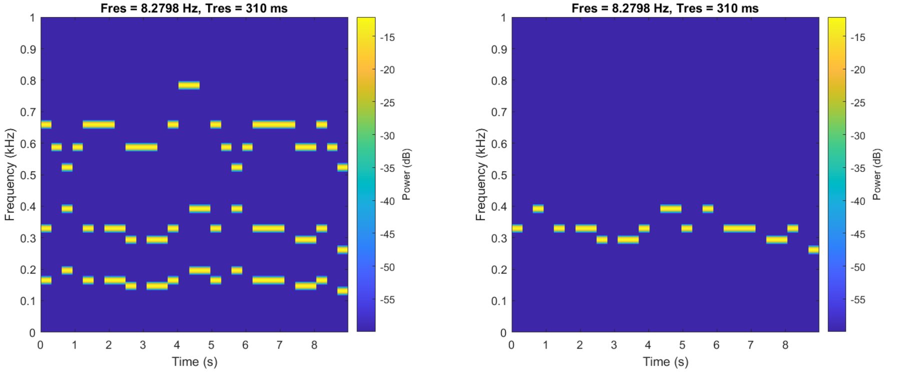
- **Question:** If $Y[k] = H[k]X[k]$, what is $y[n]$?

Circular convolution: $$y[n] = \sum_{k=0}^{N-1} h[k]x[[n-k]]_N$$

where $x[[n]]_N = x[n \bmod N] = \sum_{l=-\infty}^{\infty} x[n-lN]$ is the N-periodic extension of $x[n]$

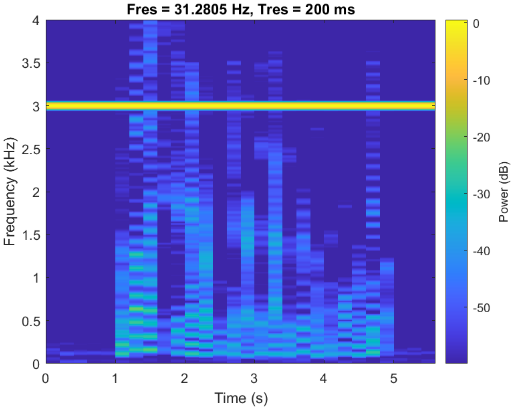
Exercise 10: Band-pass Filtering

5 Linear processing



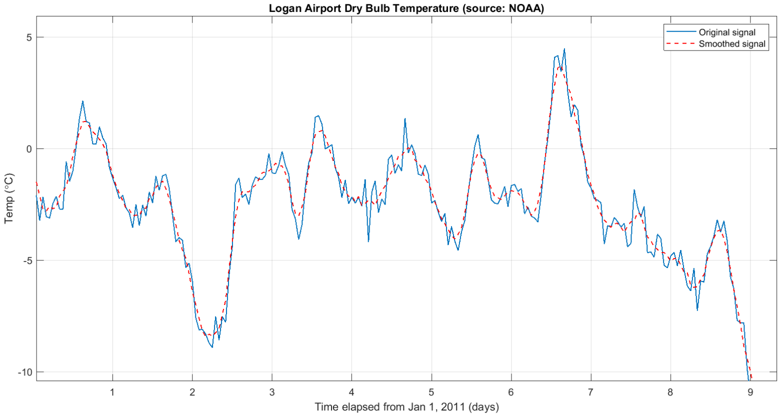
- **Matlab code:** Exercise10.m

Exercise 11: Filtering a narrow-band interference



- Can you filter a narrow-band interference from speech signals? **Matlab code:** Exercise11.m

Exercise 12: Signal smoothing with moving average



- **Matlab code:** Exercise12.m

- In the 2D case, an LTI system is always characterized by the input-output relation

$$y(t_1,t_2)=\int_{-\infty}^{\infty} h(\tau_1,\tau_2)x(t_1-\tau_1,t_2-\tau_2)d\tau_1 d\tau_2$$

where $h(t_1,t_2)$ is the 2D impulse response of the system

- The spectrum of the output is then

$$Y(f_1,f_2)=H(f_1,f_2)X(f_1,f_2)$$

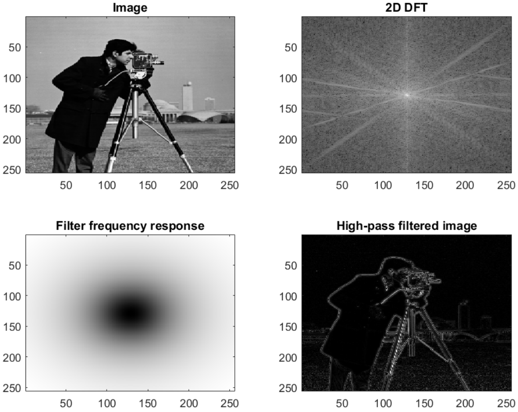
where $X(f_1,f_2)$ and $H(f_1,f_2)$ are the 2D Fourier transforms of the input signal and the impulse response, respectively

Exercise 13: Image deblurring

5 Linear processing

- **Matlab code:** Exercise13.m

Exercise 14: High-pass filtering of images



- **Matlab code:** Exercise14.m