Signals and systems

Signals, systems and tools: 3BM - 3BS
Signals, systems and telecommunications: 3BE

Franky De Bruyne

F.DeBruyne@ecam.be

ECAM Brussels Engineering School

2024

1. Introduction
   - Practicalities
   - References
   - Mathematical foundations
   - The usual suspects
   - Protagonists of the course
   - Women in science
   - Setting the stage

2. Continuous-time signals
- Basic signal properties and operations
- Basic signals
- Energy and power
- Complex Exponentials

3. Continuous-time systems
- Lumped versus distributed
- Linear time invariant systems
- Representations
- Convolution integral
- First order system
- Stability

4. Laplace transform

Definitions
Laplace transform computations
Laplace transform and non causal signals
Properties of the Laplace transform
Transfer function
One-sided Laplace transforms
Stability in the Laplace domain
Dynamic behaviour
Initial and final value theorems
Inverse of one-sided Laplace transforms
Analysis of LTI systems
Frequency response
First order system with delay
Bode diagram animation
Filtering
Matlab and Octave
One-sided Laplace transforms
Basic properties of one-sided Laplace transforms

5. Fourier frequency analysis
Some linear algebra
Projection on a function space
Fourier series
Alternative fourier series
Basic properties of Fourier series
Fourier transform
Matlab
Basic properties of the Fourier transform

6. Sampling theory

Ideal impulse sampling
Frequency folding
Analog-to-Digital Conversion
Digital-to-Analog Conversion
ADC followed by DAC

7. Discrete-time signals and systems

Discrete-time signals
Discrete-time systems

8. Z-transform

Introduction
Definitions
Z-transform and non causal signals
Mapping the s-plane into the z-plan
Computation of z-transforms
Properties of the z-transform
Transfer function
Stability
Dynamic behaviour
Initial and final value theorems
Inverse z-transform
LTI system analysis
Laplace, Fourier and z-transforms
Sampled cosine
Matlab and Octave
One-sided z-transforms
Basic properties of one-sided z-transforms

9. Discrete-time Fourier frequency analysis
- Discrete-time Fourier transform
- Discrete Fourier transform
  - Time aliasing
  - Zero padding
  - Spectral leakage
  - Practical aspects
  - Illustrating example
- Fourier series and transforms
- Fast Fourier transform
- Fast Fourier transform algorithm
- Matlab and Octave

10. Getting started with ...

- Introduction
- Variables
- Script files
- Strings
- Graphics
- Function files
- Programming
- Polynomials
- Numerical methods
- Symbolic computations

11. Dictionary

1. Introduction

Practicalities
References
Mathematical foundations
The usual suspects
Protagonists of the course
Women in science
Setting the stage

Practicalities

- Exam: slides + (video-)course + exercises[1]
- Typos
- Matlab

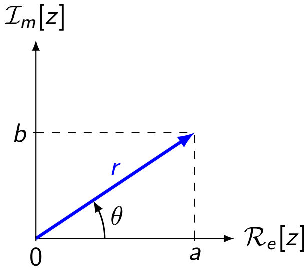
[1] Questions are in English

References

Luis F. Chaparro, "Signals and Systems using Matlab", Academic Press, Elsevier, 2011.

The Scientist and Engineer's Guide to Digital Signal Processing by Steven W. Smith, Ph.D.
http://www.dspguide.com/pdfbook.htm

Complex numbers



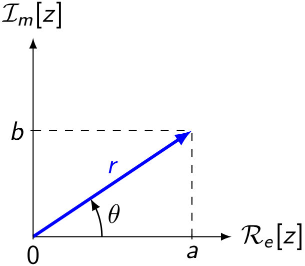
$$r = |z| = \sqrt{a^2 + b^2}$$

$$\theta = \arctan\left(\frac{b}{a}\right)$$

- **Cartesian form**: $z = a + j\,b$, $j = \sqrt{-1}$
  $a = \mathcal{R}_e[z]$, a is the **real** part of z
  $b = \mathcal{I}_m[z]$, b is the **imaginary** part of z
- **Polar form**: $z = r\,e^{j\theta}$
  $r = |z|$, r is the **modulus** of z
  $\theta = \arg[z]$, θ is the **argument** of z
- **Complex conjugate**: $z^\star = a - jb$

$$zz^\star = |z|^2 = a^2 + b^2$$

Complex numbers



$$r = |z| = \sqrt{a^2 + b^2}$$

$$\theta = \arctan\left(\frac{b}{a}\right)$$

- **Euler identity**:

$$e^{j\theta} = \cos(\theta) + j\, sin(\theta)$$

- Note that

$$\cos(\theta) = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right)$$

$$\sin(\theta) = \frac{1}{2j}\left(e^{j\theta} - e^{-j\theta}\right)$$

- Note that $e^{j\pi} = -1$, $e^{j\frac{\pi}{2}} = j$.

Exponentials

F.Y.I.

- **Natural** exponential function of x is denoted $\exp(x)$ or e^x with properties
  - $e^{(x+y)} = e^x e^y$,
  - $e^{(x-y)} = e^x e^{-y} = \frac{e^x}{e^y}$,
  - $[e^x]' = e^x$,
  - $a^x = e^{x \ln(a)}, a > 0$.

Logarithms

F.Y.I.

- **Natural** logarithm of $x > 0$ is denoted $\ln(x)$ with properties
  - $\ln(e^x) = x$,
  - $\ln(1) = 0$,
  - $\ln(e) = 1$,
  - $\ln(ab) = \ln(a) + \ln(b)$,
  - $\ln(\frac{a}{b}) = \ln(a) - \ln(b)$,
  - $\ln(x^r) = r\ln(x)$.
- **Logarithms** in any **base** $b > 1$ can be determined using
  - $\log_b(x) = \frac{\ln(x)}{\ln(b)}$,
  - $\log_b(ac) = \log_b(a) + \log_b(c)$,
  - $\ln(x) = \log_e(x)$,
  - $\log_b(x) = y \Leftrightarrow b^y = x$.

Subadditivity or triangle inequality

Subadditivity or triangle inequality

With a, b **real numbers**, the following property holds

$$|a+b| \leq |a| + |b|$$

Proof:

$$(a+b)^2 = a^2 + b^2 + 2\,ab \leq a^2 + b^2 + 2\,|a||b| = (|a| + |b|)^2$$

Subadditivity or triangle inequality

The property extends to a **finite collection** of **real numbers** a_k

$$\left|\sum_k a_k\right| \leq \sum_k |a_k|,$$

to **complex numbers** z_k, i.e.

$$\left|\sum_k z_k\right| \leq \sum_k |z_k|,$$

and to **integration** of **complex valued functions**,

$$\left|\int f\,dx\right| \leq \int |f|\,dx.$$

The usual suspects

F.Y.I.

These are problems that **often** occur in the written component, i.e. exercises on the Laplace and z-transforms, of the exam:

- **Factorising** an algebraic expression: **multiply** out the factors to check that your answer is **correct** !
- **Partial fraction** expansion: **check** the partial fraction decomposition by **combining** the partial fractions to obtain the **original** fraction.
- The solution of your **differential equations** or **difference equations** should **verify** the **initial conditions** !

The usual suspects

F.Y.I.

These are problems that **often** occur in the written component, i.e. exercises on the Laplace and z-transforms, of the exam:

- **Understand** how you should take into account **initial conditions** in your **differential equations** or **difference equations**.
- In particular:

$$\begin{aligned}\mathcal{L}[f''(t)u(t)] &= s\,\mathcal{L}[f'(t)u(t)] - f'(0)\\ &= s\,(s\,F(s) - f(0)) - f'(0)\\ &= s^2F(s) - s\,f(0) - f'(0)\end{aligned}$$

The usual suspects

F.Y.I.

These are problems that **often** occur in the written component, i.e. exercises on the Laplace and z-transforms, of the exam:

- We have that

$$\mathcal{L}[e^{2t}\,u(t)] = \frac{1}{s-2}$$

  This is **consistent** as $e^{2t}\,u(t)$ is an **unstable** function and its **Laplace transform** should have a pole, i.e. $p = 2$, on the **right-hand side of the imaginary axis** !

Hendrik Bode

F.Y.I.

Hendrik Wade Bode (December 24, 1905 – June 21, 1982) was an American engineer, researcher, inventor, author and scientist, of Dutch ancestry. As a pioneer of modern control theory and electronic telecommunications he revolutionized both the content and methodology of his chosen fields of research.

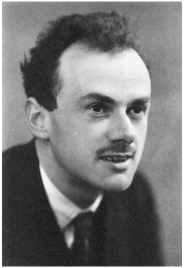
He made important contributions to the design, guidance and control of anti-aircraft systems during World War II and, continuing post-World War II during the Cold War, to the design and control of missiles and anti-ballistic missiles.

He also made important contributions to control system theory and mathematical tools for the analysis of stability of linear systems, inventing Bode plots, gain margin and phase margin.

Sources: Wikipedia
https://en.wikipedia.org//wiki/Hendrik_Wade_Bode

Paul Dirac

F.Y.I.



Paul Adrien Maurice Dirac (8 August 1902 – 20 October 1984) was an English theoretical physicist who is regarded as one of the most significant physicists of the 20th century.

Dirac made fundamental contributions to the early development of both quantum mechanics and quantum electrodynamics. Among other discoveries, he formulated the Dirac equation which describes the behaviour of fermions and predicted the existence of antimatter. Dirac shared the 1933 Nobel Prize in Physics with Erwin Schrödinger "for the discovery of new productive forms of atomic theory". He also made significant contributions to the reconciliation of general relativity with quantum mechanics.

Dirac was regarded by his friends and colleagues as unusual in character. In a 1926 letter to Paul Ehrenfest, Albert Einstein wrote of Dirac, "This balancing on the dizzying path between genius and madness is awful".

Sources: Wikipedia
https://en.wikipedia.org//wiki/Paul_Dirac

Joseph Fourier

F.Y.I.

Jean-Baptiste Joseph Fourier (21 March 1768 – 16 May 1830) was a French mathematician and physicist born in Auxerre and best known for initiating the investigation of Fourier series and their applications to problems of heat transfer and vibrations. The Fourier transform and Fourier's law are also named in his honour. Fourier is also generally credited with the discovery of the greenhouse effect.

Sources: Wikipedia
`https://en.wikipedia.org//wiki/Joseph_Fourier`

Pierre-Simon Laplace

F.Y.I.

Pierre-Simon, marquis de Laplace (23 March 1749 – 5 March 1827) was a French scholar whose work was important to the development of engineering, mathematics, statistics, physics, astronomy, and philosophy. He summarized and extended the work of his predecessors in his five-volume Mécanique Céleste (Celestial Mechanics) (1799–1825). This work translated the geometric study of classical mechanics to one based on calculus, opening up a broader range of problems. In statistics, the Bayesian interpretation of probability was developed mainly by Laplace. Laplace formulated Laplace's equation, and pioneered the Laplace transform which appears in many branches of mathematical physics, a field that he took a leading role in forming.

Sources: Wikipedia
https://en.wikipedia.org//wiki/Pierre-Simon_Laplace

Harry Nyquist

F.Y.I.

Harry Nyquist (February 7, 1889 – April 4, 1976) was a Swedish-born American electronic engineer who made important contributions to communication theory. As an engineer at Bell Laboratories, Nyquist did important work on thermal noise ("Johnson–Nyquist noise"), the stability of feedback amplifiers, telegraphy, facsimile, television, and other important communications problems.

Sources: Wikipedia
https://en.wikipedia.org//wiki/Harry_Nyquist

Ingrid Daubechies[2]

- Renowned **Belgian mathematician** and **physicist**
- Known for her work on **wavelets** and image compression
- She lent her name to the **Daubechies wavelets**, used in the JPEG 2000 standard.
- She has been the recipient of **numerous awards**.

[2]Picture drawn from https://wolffund.org.il/ingrid-daubechies/

Fourier transform and wavelet transform

UN WOMEN F.Y.I.

Fourier transform (FT):

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

The **FT** represents a signal in the **frequency domain**, decomposing it into its **constituent frequencies**. It gives a global view of **frequency content** but does **not** provide information about **when** these frequencies occur.
Assumption: signal $f(t)$ is **stationary** !

Continuous wavelet transform (CWT):

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t)\; \frac{1}{\sqrt{|a|}}\, \psi^{*}\left(\frac{t-b}{a}\right)\, dt$$

The **CWT** represents a signal in **both time** and **frequency domains** simultaneously. Unlike the Fourier Transform, which uses fixed sinusoids, wavelet transform employs functions that are **localized in both time and frequency**. This provides a time-frequency representation of the signal, capturing both high and low-**frequency components** at **different time intervals**.

Fourier transform and wavelet transform

UN WOMEN F.Y.I.

Inverse Fourier transform:

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

Essentially, this means that $f(t)$ is **reconstructed** as a **sum of sinusoids** (sines and cosines) with **different frequencies**, where **each sinusoid contributes** according to its **magnitude and phase** in the **frequency domain**.

Inverse continuous wavelet transform:

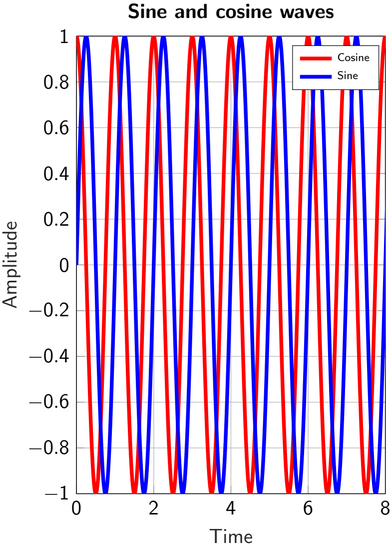
$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b)\ \frac{1}{\sqrt{|a|}}\, \psi\left(\frac{t-b}{a}\right)\, da\, db$$

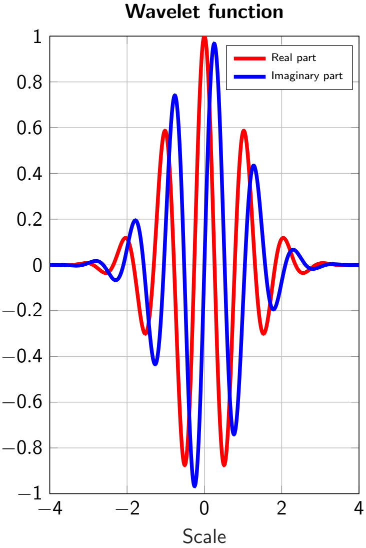
In essence, the **inverse CWT reconstructs** $f(t)$ by **summing up localized wavelet contributions** across **different scales and positions**.

Fourier basis and wavelet functions

```
% Define parameters
t = linspace(0, 8, 1000); % Time vector
f = 1; % Frequency of the sine wave
A = 1; % Amplitude of the sine wave

% Generate sine and cosine waves
sine_wave = A*sin(2*pi*f*t);
cosine_wave = A*cos(2*pi*f*t);

% Define wavelet function (example: Morlet wavelet)
s = linspace(-4, 4, 1000); % Scale vector
wavelet_function = exp(1i*2*pi*f*s).*exp(-s.^2/2);

% Plot sine and cosine waves, and wavelet function
figure;
subplot(1,2,1);
plot(t, cosine_wave, 'r', 'LineWidth', 2); hold on;
plot(t, sine_wave, 'b', 'LineWidth', 2);
xlabel('Time'); ylabel('Amplitude');
title('Sine and cosine waves'); legend('Cosine','Sine'); grid on;

subplot(1,2,2);
plot(s, real(wavelet_function), 'r', 'LineWidth', 2); hold on;
plot(s, imag(wavelet_function), 'b', 'LineWidth', 2);
xlabel('Scale'); title('Wavelet function');
legend('Real part', 'Imaginary part'); grid on;
```

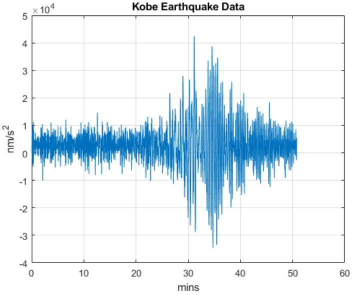
Fourier basis and wavelet functions

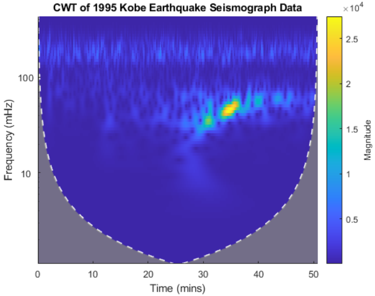




Kobe earthquake 1995[3]

```
load kobe;
Ts = 1;
plot((1:numel(kobe))./60,kobe);
xlabel('mins'); ylabel('nm/s^2');
grid on;
title('Kobe Earthquake Data');
figure;
cwt(kobe,1/Ts)
title('CWT of 1995 Kobe Earthquake Seismograph Data');
```





[3] Requires Matlab Wavelet Toolbox. Source:
https://nl.mathworks.com/help/wavelet/ug/continuous-wavelet-analysis-of-modulated-signals.html

Signal

Signal

A **signal** is a formal description of a phenomenon evolving over **time** or space. It is the **information carrier** emitted by a **source** and intended for a **receiver**; it is the **vehicle of intelligence** in systems.

Classification according to:

- **origin**: audio, video, speech, image, communication, geophysical, sonar, radar, medical and musical signals, etc.
- **dimension**: real (or complex) valued function of one or more real variables.
- **description type**: signals that can be modeled exactly by a mathematical formula are known as **deterministic** signals. Deterministic signals are not always adequate to model real-world situations. **Stochastic** (random) signals, on the other hand, cannot be described by a mathematical equation; they are modeled in probabilistic terms.

Analog and digital signals

- Most natural signals are **analog** in nature:
  - these signals are continuous functions of time and/or space.
  - Historically, before the advent of widespread digital technology, analog signal processing was the only method by which to manipulate a signal, e.g. telephone, loud speaker.
- Since that time, as computers and software became more advanced, digital signal processing has become the method of choice. **Analog to Digital Converters** (ADC) perform periodic **sampling** and **quantisation** of the input signal. The result is a sequence of **digital**[4] values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal, i.e.
  - the signals are defined **only** at given **discrete** sampling instants
  - and only take a **finite number of discrete values**.

[4] The adjective "digital" derives from *digitus*, Latin for finger: it captures the idea that the signal is represented by a sequence of integer numbers.

Digital Signal Processing (DSP)

Objects: digital signals, i.e.

- sequences of integer numbers,
- digitized discrete-time signals sampled from a time continuous analog signal,
- variations of a physical quantity that provides information on the status of a physical system.

Operations:

- analysis: understand the information content of the signal,
- processing: modify this information.

Wikipedia

Digital Signal Processing (DSP) is the **mathematical manipulation** of an information signal to **modify** or **improve** it in some way.

It is characterized by the representation of **discrete-time**, **discrete frequency**, or other discrete domain signals by a **sequence of numbers** or symbols and the **processing** of these signals.

Historical background

F.Y.I.

Roots of DSP: 1960s and 1970s

- Digital computers/processors first becomes available.
- Fast Fourier Transform (FFT) algorithm and description how to implement it conveniently on a computer: "An algorithm for the machine calculation of complex Fourier series", Cooley & Tukey, 1965.
- Before this time, signal processing is analog, numerical algorithms are available but not recognised as being useful because of the lack of computational power.
- Computers are expensive during this era, and DSP is limited to only a few critical applications. Pioneering efforts are made in four key areas:
  - **radar and sonar**: national security of United States
  - **oil exploration**: financial motivation
  - **space exploration**: irreplaceable data
  - **medical imaging**: saving lives

Historical background

F.Y.I.

Personal Computer revolution: 1980s and 1990s

- Rather than being motivated by military and government needs, DSP is suddenly driven by the commercial marketplace.
- First commercial Digital Signal Processors (DSPs).
- DSP reaches the public in such products as: mobile telephones, compact disc players, and electronic voice mail.

Historical background

F.Y.I.

Towards embedded intelligence:

- The technologies underlying digital signal processing (ADC - DAC - computing and communications devices) have ever increasing capabilities and declining costs. A multitude of commercial objects have been equipped with them.
- The paradigm **central intelligence unit** connected to peripheral units is gradually abandoned in favour of **distributed embedded systems** with distributed intelligence resulting in networked systems of embedded computers whose functional components are nearly invisible to end users.
- Systems have the potential to alter radically the way in which people interact with their environment by linking a range of devices and sensors that will allow information to be collected, shared, and processed in unprecedented ways.
- Today, DSP is a basic skill needed by scientists and engineers in many fields.

Advantages of DSP

Digital systems have a number of key advantages over their analog counterparts:

- **simplicity**: difference or recursive equations
- **flexibility**: programmable and re-programmable
- **power**: more complex computations, e.g. nonlinear operations
- **cost and congestion**: ever decreasing cost and size
- **robustness to noise**: digital circuits are less affected by noise
- **precision and stability**: quantisation noise has to be taken into consideration, insensibility to temperature and system age
- **memory**: information storage is easy and less memory space is required

Advantages of DSP

DSP has produced revolutionary changes in a number of fields:

- **classical signal processing**: filtering, fast transforms (FFT, wavelets), signal analysers and generators
- **telecommunication**: modulation and demodulation, adaptive equalising, echo cancellation
- **image and speech processing**: compression, analysis, voice recognition and synthesis, pattern recognition
- **radar and sonar**: echo location, target identification
- **medical applications**: Digital Signal Processing for EEG[5] and Image processing for MRI[6]
- **control**: filtering, estimators, control algorithms, optimisation
- **asset management**: performance monitoring, condition monitoring

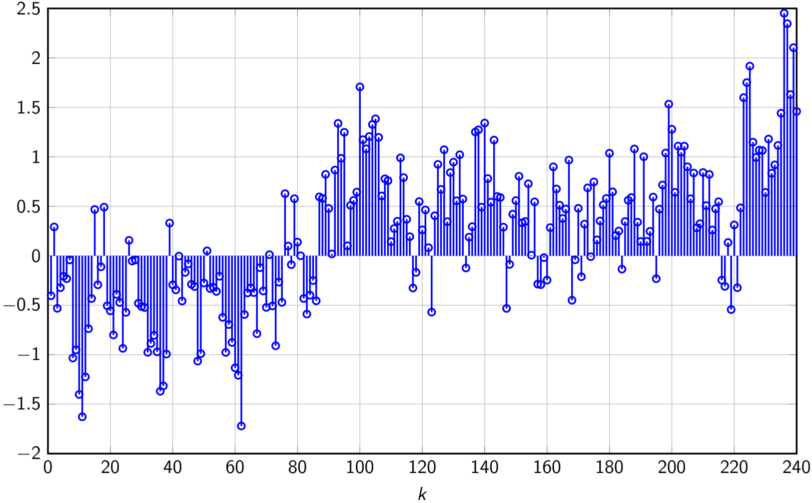
[5] Electro-EncephaloGraphy
[6] Magnetic Resonance Imaging

Spectral analysis example[7]

- Suppose we would like to analyse sounds recorded in the ocean.
- A hydrophone is used to amplify and record the signal.
- An **analog low pass filter** is used to remove spectral components above 80Hz. The signal is sampled at 160Hz.
- The **spectrum** of the signal is subsequently **analysed**.

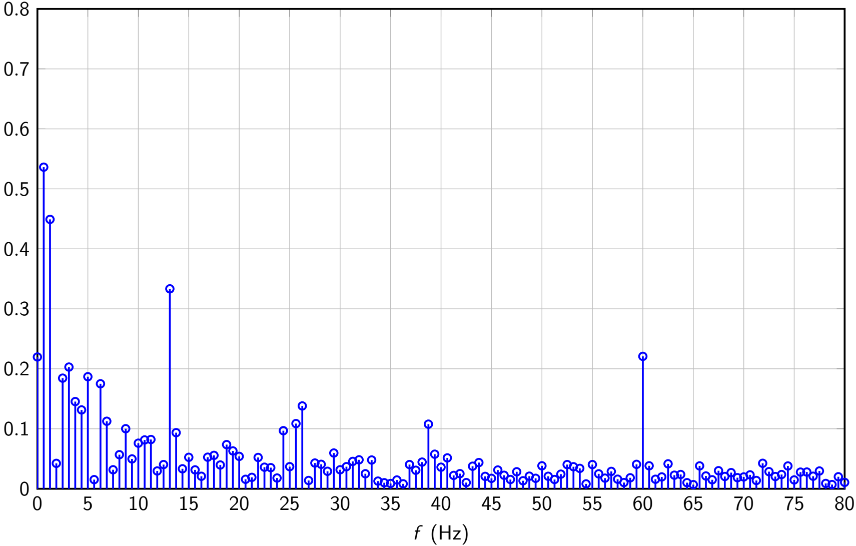
[7] GELE2511, Chapitre 7: Transformée de Fourier discrète, Gabriel Cormier, Université de Moncton.

Spectral analysis example: preprocessed signal



240 data points, i.e. 1.5 seconds of recording

Spectral analysis example: signal spectrum



Spectral analysis example

Let us analyse the **spectral content**:

- If the peaks are ignored, the spectrum is relatively constant between 20Hz and 70Hz; this is **white Gaussian noise**.
- White Gaussian noise draws its name from the flat spectrum over frequency.
- White Gaussian noise can be caused from many sources, e.g. microphone, ocean.

Spectral analysis example

Let us analyse the **spectral content** (continued):

- At low frequencies, the noise level increases rapidly in what appears to be a $\frac{1}{f^\alpha}$ relation: this is **flicker noise** or pink noise[8].
- Flicker noise seems ever present in physical systems (mechanical and electronic devices) and life science.
- Although many sources of noise are well understood, the origin of flicker noises remains, in general, a mystery in spite of their remarkably widespread occurrence in nature.

[8] bruit de scintillement

Spectral analysis example

Let us analyse the **spectral content** (continued): peaks

- 60Hz[9] peak: as most appliances and devices draw their power from the power grid, power line related noises are most prominent in our living environment.

En general, we can also observe:

- **harmonics** (150Hz, 250Hz, etc.) of the main power frequency,
- higher frequency pollution emitted by **switching power adapters** (generally from 10kHz to 1MHz),
- frequencies related to **mechanical devices**, e.g. a faulty bearing will causes vibrations[10] that depend on the rotational speed and the bearing characteristics.

[9] 50Hz in Europe
[10] Vibration based condition monitoring

Spectral analysis example

Let us analyse the **spectral content** (continued): peaks

- An important peak can be distinguished around 13Hz with smaller peaks around 26Hz and 39Hz, i.e. second and third order harmonics of the first peak.
- The peaks could have been caused by a three blade propeller rotating at approximately 4.33 turns/second.
- This technique lies on the basis of **passive sonar**.

2. Continuous-time signals

Basic signal properties and operations
Basic signals
Energy and power
Complex Exponentials

Definition

Continuous signal

A **continuous signal** $f(.)$ can be thought of as a real or complex valued function of an independent variable.

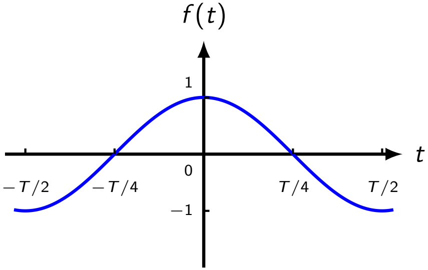
The independent variable is often **time** t, i.e. this results in a continuous-time signal $f(t)$.

The terms **continuous-time** and **analog** are used interchangeably for these signals.

Properties: parity

Even function

A function is **even** if $\forall t \in \mathbb{R}$: $f(-t) = f(t)$.



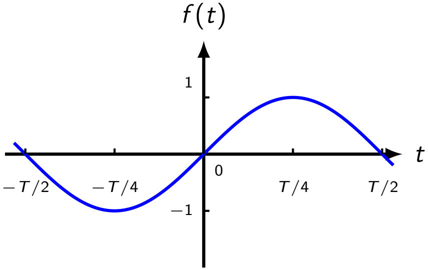
Graphical properties

The graph of an even function is **symmetric around the vertical axis**.
A graph is symmetric around the vertical axis if the point (-t, $f(t)$) lies on the graph whenever $(t, f(t))$ does.

Properties: parity

Odd function

A function is **odd** if $\forall t \in \mathbb{R}$: $f(-t) = -f(t)$.



Graphical properties

The graph of an odd function is **symmetric around the origin**.
A graph is symmetric around the origin if the point $(t, f(t))$ lies on the graph whenever $(-t, -f(t))$ does.

Properties: even and odd decomposition

Even and odd decomposition

A continuous-time signal $f(t)$ is representable as a **sum** of an **even component** and an **odd component**:

$$\begin{aligned} f(t) &= \underbrace{\frac{1}{2}(f(t)+f(-t))}_{f_e(t)} + \underbrace{\frac{1}{2}(f(t)-f(-t))}_{f_o(t)} \\ &= f_e(t)+f_o(t) \end{aligned}$$

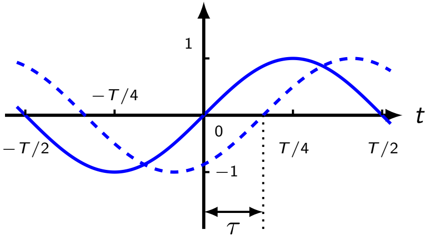
By construction, $f_e(t)$ and $f_o(t)$ are, respectively, even and odd.

Operations: time shifting

Delayed or shifted function (time shifting)

A function $g(t)$ is the original function $f(t)$ **delayed** of **shifted** by τ if $\forall t \in \mathbb{R}$, $g(t) = D_\tau f(t) = f(t-\tau)$. D_τ is the **delay operator**. The output of D_τ is the input signal delayed by τ seconds.

$f(t), g(t) = f(t-\tau)$ with $\tau > 0$



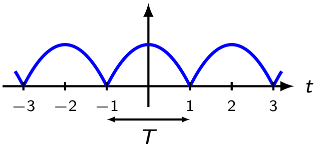
Properties

- If $\tau > 0$, $g(t)$ **lags** $f(t)$.
- If $\tau < 0$, $g(t)$ **leads** $f(t)$.

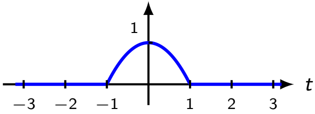
Properties: periodicity

Periodic function

A **periodic** function $f(t)$ with period T is the **infinite repetition** of the same **motif** $f_T(t)$ defined on an interval T. The **motif** $f_T(t)$ is often the **restriction** of function $g(t)$ on the **interval** T.



$$f_T(t) = \begin{cases} g(t) = 1 - t^2, t \in [-1, 1] \\ 0, \text{ elsewhere} \end{cases}$$



Properties

Periodic signals are defined by the relation

$$f(t) = \sum_{k=-\infty}^{\infty} f_T(t - kT), \quad k \in \mathbb{Z}$$

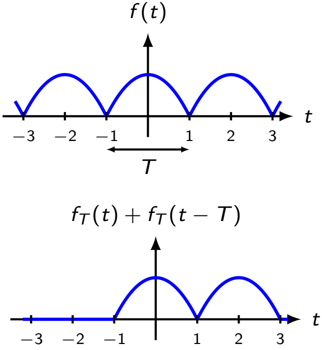
Periodicity:

$$\boxed{\forall t \in \mathbb{R}, k \in \mathbb{Z}, f(t) = f(t - kT)}$$

Properties: periodicity

Periodic function

A **periodic** function $f(t)$ with period T is the **infinite repetition** of the same **motif** $f_T(t)$ defined on an interval T. The **motif** $f_T(t)$ is often the **restriction** of function $g(t)$ on the **interval** T.



Properties

Periodic signals are defined by the relation

$$f(t) = \sum_{k=-\infty}^{\infty} f_T(t - kT), \quad k \in \mathbb{Z}$$

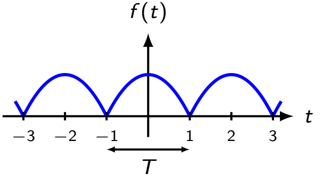
Periodicity:

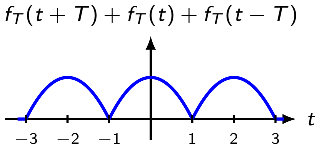
$$\boxed{\forall t \in \mathbb{R}, k \in \mathbb{Z}, f(t) = f(t - kT)}$$

Properties: periodicity

Periodic function

A **periodic** function $f(t)$ with period T is the **infinite repetition** of the same **motif** $f_T(t)$ defined on an interval T. The **motif** $f_T(t)$ is often the **restriction** of function $g(t)$ on the **interval** T.





Properties

Periodic signals are defined by the relation

$$f(t) = \sum_{k=-\infty}^{\infty} f_T(t - kT), \quad k \in \mathbb{Z}$$

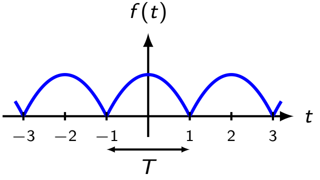
Periodicity:

$$\boxed{\forall t \in \mathbb{R}, k \in \mathbb{Z}, f(t) = f(t - kT)}$$

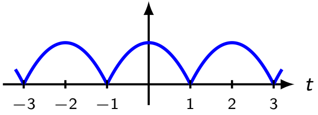
Properties: periodicity

Periodic function

A **periodic** function $f(t)$ with period T is the **infinite repetition** of the same **motif** $f_T(t)$ defined on an interval T. The **motif** $f_T(t)$ is often the **restriction** of function $g(t)$ on the **interval** T.



$\sum_{k=-\infty}^{\infty} f_T(t-kT), \quad k \in \mathbb{Z}$



Properties

Periodic signals are defined by the relation

$$f(t) = \sum_{k=-\infty}^{\infty} f_T(t-kT), \quad k \in \mathbb{Z}$$

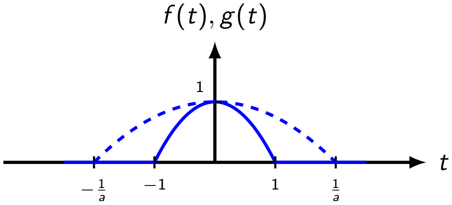
Periodicity:

$$\boxed{\forall t \in \mathbb{R}, k \in \mathbb{Z}, f(t) = f(t-kT)}$$

Operations: time scaling

Time scaling (expansion or contraction)

The **time scaling** operation transforms the function $f(t)$ into the function $g(t) = f(at)$ with $a \in \mathbb{R}^{0+}$. Equivalently, $g(t/a) = f(t)$.



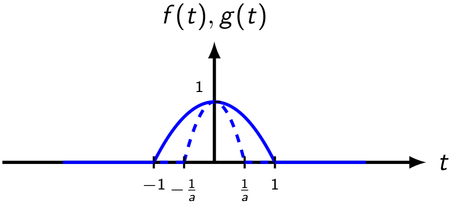
Expansion

$g(t) = f(at), 0 < a < 1$

Operations: time scaling

Time scaling (expansion or contraction)

The **time scaling** operation transforms the function $f(t)$ into the function $g(t) = f(at)$ with $a \in \mathbb{R}^{0+}$. Equivalently, $g(t/a) = f(t)$.



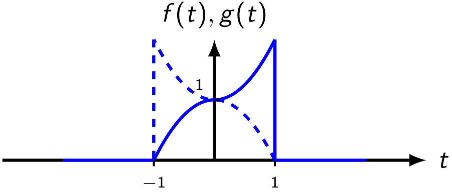
Compression

$$g(t) = f(at),\ a > 1$$

Operations: time reversal

Time reversal

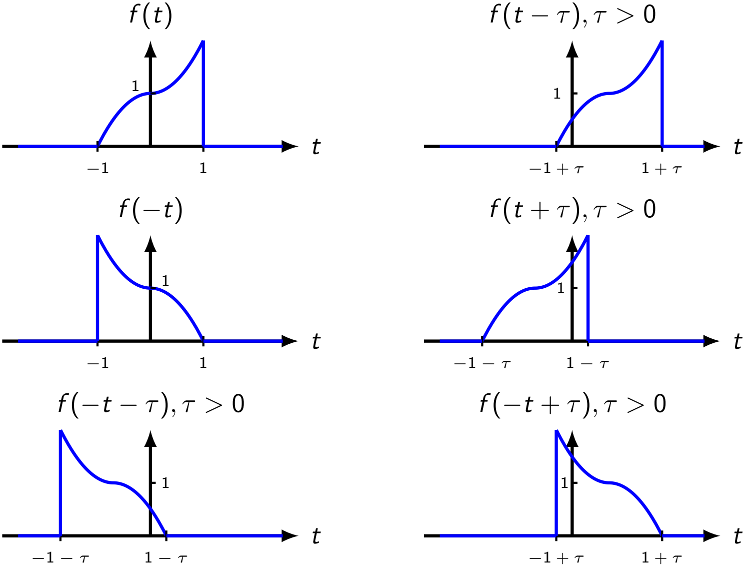
The **time reversal** or **reflection** operation transforms the function $f(t)$ into the function $g(t) = f(-t)$.



Use

The **time reversal** operation will be used when computing the **convolution integral**.

Operations: combinations



Operations: remarks

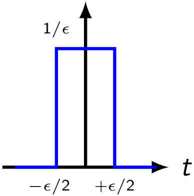
- Note that $f(-t+1) = f(-(t-1))$ is $f(t)$ **reflected** and **delayed**. The value $f(0)$ of the original signal is found in $f(-t+1)$ at $t = 1$ or shifted to the **right**, by 1.
- Likewise, $f(-t-1) = f(-(t+1))$ is $f(t)$ **reflected** and **advanced**. The value $f(0)$ of the original signal is found in $f(-t-1)$ at $t = -1$ or shifted to the **left** by 1.

- It is important to understand that **advancing** or **reflecting** **cannot be implemented in real-time**, i.e. that is as the signal is being processed.
- **Delays** can be implemented in **real-time**.

Dirac impulse

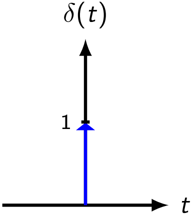
Consider a **rectangular impulse** of duration ϵ and **unit area**:

$$\delta_\epsilon(t) = \frac{1}{\epsilon}\,\Pi\left(\frac{t}{\epsilon}\right)$$



$$\delta_\epsilon(t) = \begin{cases} \frac{1}{\epsilon} & -\epsilon/2 \le t \le \epsilon/2 \\ 0 & t < -\epsilon/2 \text{ and } t > \epsilon/2 \end{cases}$$

Extremely narrow unit area impulse $\delta_\epsilon(t)$ when $\epsilon \to 0$.



Dirac impulse or distribution:

$$\delta(t) = \lim_{\epsilon \to 0} \delta_\epsilon(t)$$

Dirac impulse properties

Properties

- $\delta(t) = 0$ for $t \neq 0$,
- $\delta(t)$ is **not** defined for $t = 0$,
- $\int_{-a}^{a} \delta(t) dt = 1$ for $a > 0$.

Dirac impulse: sifting property[11]

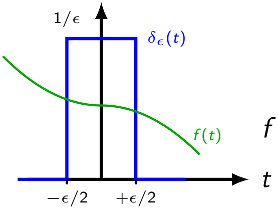
If $f(t)$ is a **continuous signal** in $t = 0$, then

$$\boxed{f(t)\,\delta(t) = f(0)\,\delta(t)}$$

In particular

$$\int_{-a}^{a} f(t)\,\delta(t)dt = f(0) \text{ for } 0 < a \leq \infty.$$

To see this, approximate $\delta(t)$ by $\delta_\epsilon(t)$



$f(t) \approx f(0)$ for $-\epsilon/2 \leq t \leq \epsilon/2$ and ϵ small

[11] **propriété de localisation**; to sift: passer au crible

Dirac impulse: sifting property[11]

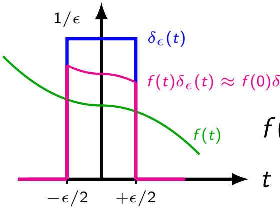
If $f(t)$ is a **continuous signal** in $t = 0$, then

$$\boxed{f(t)\,\delta(t) = f(0)\,\delta(t)}$$

In particular

$$\int_{-a}^{a} f(t)\,\delta(t)dt = f(0) \text{ for } 0 < a \leq \infty.$$

To see this, approximate $\delta(t)$ by $\delta_\epsilon(t)$



$f(t) \approx f(0)$ for $-\epsilon/2 \leq t \leq \epsilon/2$ and ϵ small

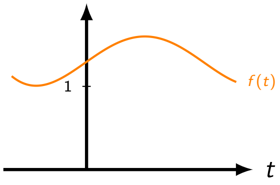
[11] **propriété de localisation**; to sift: passer au crible

Dirac impulse: generalisation of the sifting property

Generalisation of the sifting property

$$\boxed{f(t)\,\delta(t-t_0) = f(t_0)\,\delta(t-t_0)}$$

i.e. a **Dirac impulse** of weight $f(t_0)$ in $t = t_0$.



Schematic representation

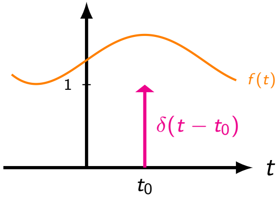
The **Dirac impulse** is represented by a **line surmounted by an arrow**. The **height** of the arrow is usually used to specify the value of the **weight**, which gives the **area** under the curve.

Dirac impulse: generalisation of the sifting property

Generalisation of the sifting property

$$\boxed{f(t)\,\delta(t-t_0) = f(t_0)\,\delta(t-t_0)}$$

i.e. a **Dirac impulse** of weight $f(t_0)$ in $t = t_0$.



Schematic representation

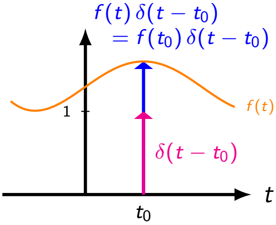
The **Dirac impulse** is represented by a **line surmounted by an arrow**. The **height** of the arrow is usually used to specify the value of the **weight**, which gives the **area** under the curve.

Dirac impulse: generalisation of the sifting property

Generalisation of the sifting property

$$\boxed{f(t)\,\delta(t-t_0) = f(t_0)\,\delta(t-t_0)}$$

i.e. a **Dirac impulse** of weight $f(t_0)$ in $t = t_0$.



Schematic representation

The **Dirac impulse** is represented by a **line surmounted by an arrow**. The **height** of the arrow is usually used to specify the value of the **weight**, which gives the **area** under the curve.

Generic representation of a signal: mathematics

Generic representation of a continuous signal

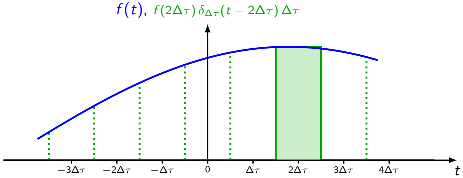
$$\boxed{f(t) = \int_{-\infty}^{\infty} f(\tau)\,\delta(t-\tau)\,d\tau}$$

Proof:

$$\begin{aligned} \int_{-\infty}^{\infty} f(\tau)\,\delta(t-\tau)d\tau &= \int_{-\infty}^{\infty} f(t)\,\delta(t-\tau)d\tau \\ &= f(t)\int_{-\infty}^{\infty} \delta(t-\tau)d\tau = f(t) \end{aligned}$$

Generic representation of a signal: intuition

F.Y.I.



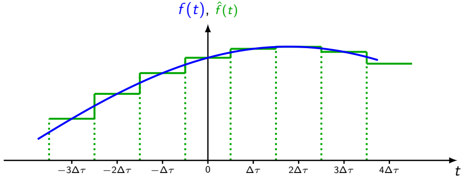
We approximate $f(t)$ by a **succession of pulses**[12] of width $\Delta\tau$, i.e.

$$\hat{f}(t) = \sum_{k=-\infty}^{\infty} f(k\Delta\tau)\,\delta_{\Delta\tau}(t-k\Delta\tau)\,\Delta\tau$$

[12] We have used the Dirac impulse approximation introduced previously.

Generic representation of a signal: intuition

F.Y.I.



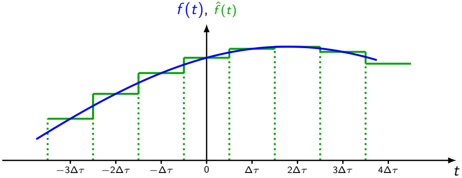
We approximate $f(t)$ by a **succession of pulses**[12] of width $\Delta\tau$, i.e.

$$\hat{f}(t) = \sum_{k=-\infty}^{\infty} f(k\Delta\tau)\,\delta_{\Delta\tau}(t - k\Delta\tau)\,\Delta\tau$$

[12] We have used the Dirac impulse approximation introduced previously.

Generic representation of a signal: intuition

F.Y.I.



We approximate $f(t)$ by a **succession of pulses**[12] of width $\Delta\tau$, i.e.

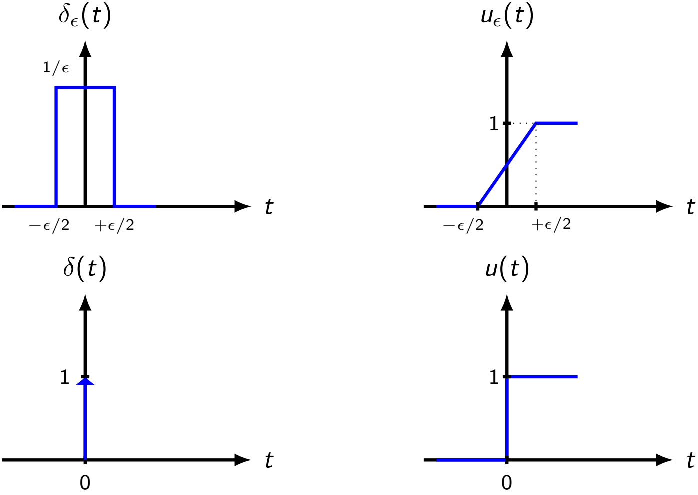
$$\hat{f}(t) = \sum_{k=-\infty}^{\infty} f(k\Delta\tau)\,\delta_{\Delta\tau}(t - k\Delta\tau)\,\Delta\tau$$

When $\Delta\tau \to 0$, $\hat{f}(t) \to f(t)$, $k\Delta\tau \to \tau$, $\Delta\tau \to d\tau$, $\delta_{\Delta\tau}(t) \to \delta(t)$ and the **sum becomes an integral**, i.e.

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\,\delta(t - \tau)\,d\tau$$

[12] We have used the Dirac impulse approximation introduced previously.

Unit step or Heaviside function



Unit step or Heaviside function

Definition

The **unit step** or **Heaviside function** is the **integral** of the **Dirac impulse**, i.e.

$$\begin{aligned} u(t) &= \int_{-\infty}^{t} \delta(\tau) d\tau \\ &= \lim_{\epsilon \to 0} u_\epsilon(t) = \lim_{\epsilon \to 0} \left(\int_{-\infty}^{t} \delta_\epsilon(\tau) d\tau \right) \end{aligned}$$

$$u_\epsilon(t) = \begin{cases} 0 & t < -\epsilon/2 \\ \frac{1}{\epsilon}\left(t + \frac{\epsilon}{2}\right) & -\epsilon/2 \le t \le \epsilon/2 \\ 1 & t > \epsilon/2 \end{cases}$$

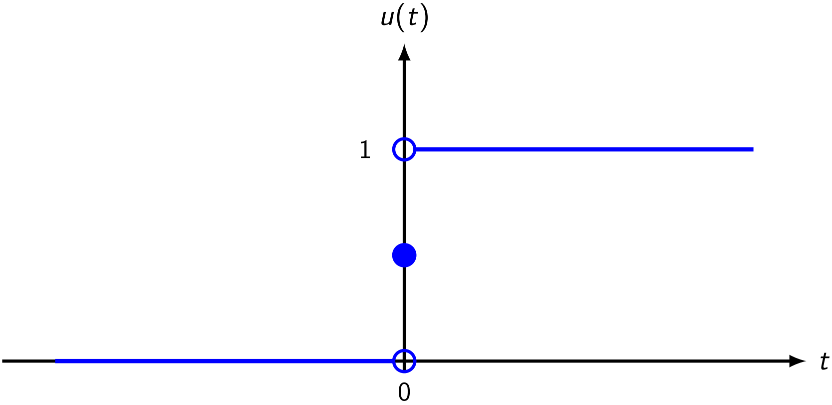
Unit step or Heaviside function

Properties

- $u(t) = \lim_{\epsilon \to 0} u_\epsilon(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2} & t = 0 \\ 1 & t > 0 \end{cases}$
- The **derivative** of $u(t)$ is **zero** over $\mathbb{R}^0$.
- The derivative of $u(t)$ is a **Dirac impulse** of weight 1 centered in $t = 0$, i.e.
$$\frac{du(t)}{dt} = \delta(t).$$
- **Useful property**: multiplying an ordinary function $f(t)$ by the step function $u(t)$ changes it into a **causal**[13] function; e.g. if $f(t) = \sin(t)$ then $f(t)u(t) = \sin(t)\,u(t)$ is **causal**.

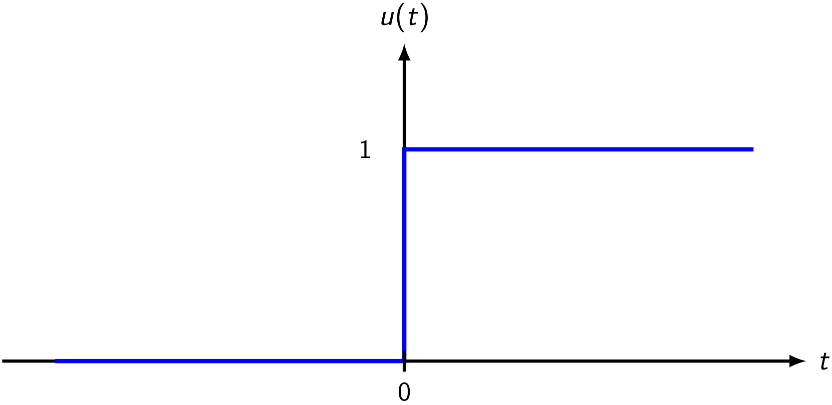
[13] causal functions take the value 0 when $t < 0$.

Unit step: "mathematically correct" representation



The Heaviside function is a **piecewise constant** function.

Unit step: "oscilloscope" representation

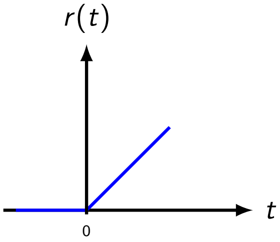


The Heaviside function as it would appear on an **oscilloscope**.

Unit step: remarks

- Since $u(t)$ is not a continuous function, it jumps from 0 to 1 instantaneously around $t = 0$, from the calculus point of view it should not have a derivative. That $\delta(t)$ is its derivative must be taken with suspicion, which makes the $\delta(t)$ signal also suspicious. Such signals can, however, be formally defined using the **theory of distributions**.
- Signals with **jump discontinuities** can be represented as the **sum** of a continuous signal and unit-step signals at the discontinuities. This is useful in computing the **derivative** of these signals.

Unit ramp



Definition

The **ramp** signal $r(t)$ is defined as

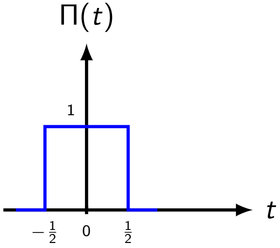
$$r(t) = \int_{-\infty}^{t} u(\tau) d\tau = t\, u(t).$$

Properties

Its **relation** to the **unit-step** and the **unit-impulse** signals is

$$\frac{dr(t)}{dt} = u(t), \quad \frac{d^2 r(t)}{dt^2} = \delta(t).$$

Rectangular impulse window function



Definition

The **rectangular impulse** window function $\Pi(t)$ is defined as

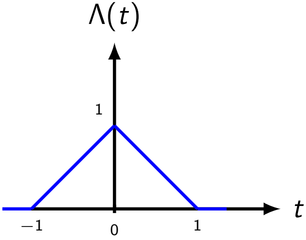
$$\begin{aligned} \Pi(t) &= u(t + 1/2) - u(t - 1/2) \\ &= \begin{cases} 1 & |t| < 1/2 \\ 1/2 & |t| = 1/2 \\ 0 & |t| > 1/2 \end{cases} \end{aligned}$$

Properties

- The **area** is $\int_{-\infty}^{\infty} \Pi(t) dt = 1$.
- Window of **length** T, **amplitude** A and **centered** in $t = \tau$:

$$\boxed{A\,\Pi\left(\frac{t-\tau}{T}\right)}$$

Triangular impulse window function



Definition

The **triangular impulse** window function $\Lambda(t)$ is defined as

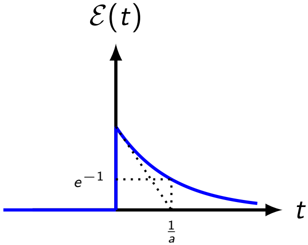
$$\begin{aligned}\Lambda(t) &= (t+1)\,u(t+1) - 2\,t\,u(t) \\ &\quad + (t-1)\,u(t-1) \\ &= \begin{cases} 1-|t| & |t| \le 1 \\ 0 & |t| > 1 \end{cases}\end{aligned}$$

Properties

- The **area** is $\int_{-\infty}^{\infty} \Lambda(t)dt = 1$.
- Window of **length** $2T$, **amplitude** A and **centered** in $t = \tau$:

$$\boxed{A\,\Lambda\left(\frac{t-\tau}{T}\right)}$$

Exponential impulse window function



Definition

The **exponential impulse** window function $\mathcal{E}(t)$ is defined as

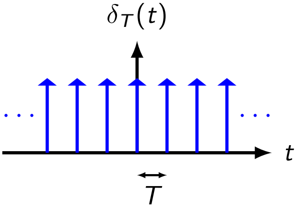
$$\mathcal{E}(t) = e^{-a\,t}u(t) \text{ for } a > 0.$$

Properties

- The **area** is $\int_{-\infty}^{\infty} \mathcal{E}(t)dt = \frac{1}{a}$.
- The exponential impulse window function ($a > 0$) can be used to **dampen**[14] a causal signal.

[14] amortir

Dirac comb[15]



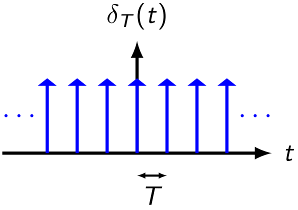
Definition

The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$

[15] peigne de Dirac

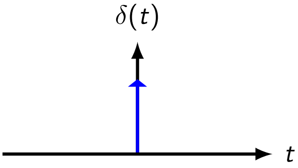
Dirac comb[15]



Definition

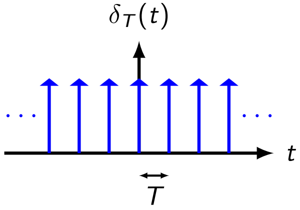
The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$



[15]peigne de Dirac

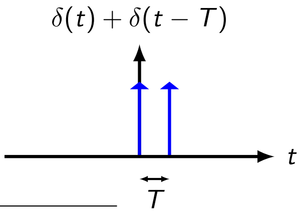
Dirac comb[15]



Definition

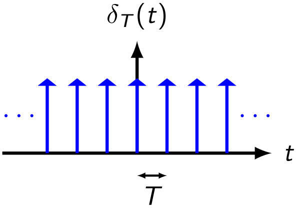
The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$



[15] peigne de Dirac

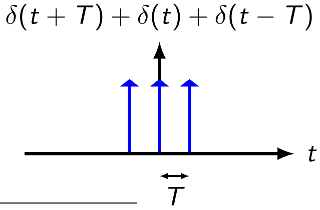
Dirac comb[15]



Definition

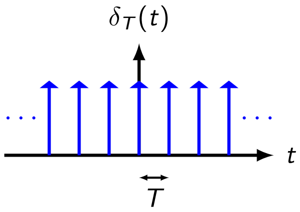
The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$



[15] peigne de Dirac

Dirac comb[15]

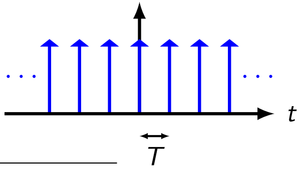


Definition

The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

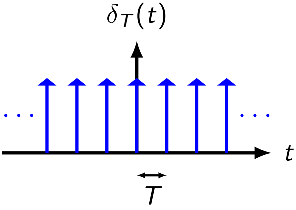
$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$

$\sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}$



[15] peigne de Dirac

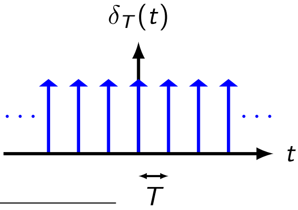
Dirac comb[15]



Definition

The **Dirac comb**, denoted $\delta_T(t)$, is an **infinite series** of Dirac delta functions **spaced** at **intervals** of T, i.e.

$$\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT), \quad k \in \mathbb{Z}.$$



[15] peigne de Dirac

Total energy

F.Y.I.

Total energy

In signal processing[16], the **energy** of a continuous-time signal $f(t)$ is defined as

$$E = \int_{-\infty}^{\infty} |f(t)|^2 \, dt.$$

In the formulas for energy (and power) we are considering the possibility that the signals might be **complex** and so we are squaring its magnitude. If the signal being considered is real, this simply is equivalent to squaring the signal.

[16]Signal energy is not the same as the conventional energy in physics !

Average power

F.Y.I.

Average power

In signal processing[17], the **average total power** of a continuous-time signal $f(t)$ is defined as

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(t)|^2 dt.$$

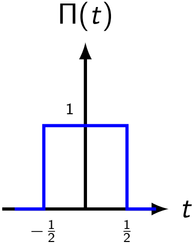
For a **periodic** signal, P is computed over a **period** T

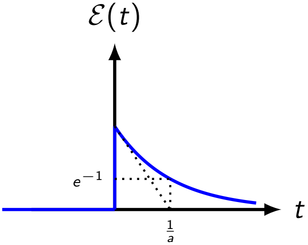
$$P = \frac{1}{T} \int_{T} |f(t)|^2 \, dt.$$

[17]Signal power is not the same as the conventional power in physics !

Finite energy, or square integrable signals

F.Y.I.





Finite energy signals

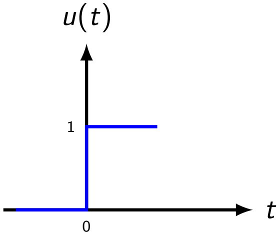
$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty$$

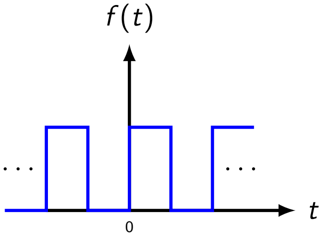
Remarks

- A **finite-energy signal** has **zero power**. Indeed, if the energy of the signal is some constant $E < \infty$, then
$$P = \lim_{T\to\infty} \frac{E}{T} = 0.$$
- These signals are often called **square integrable**.

Finite power signals

F.Y.I.





Finite (average total) power signals

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(t)|^2 dt < \infty$$

Remark

Finite power signals ($P > 0$) **always** have an **infinite total energy**, e.g. **periodic signals**.

Average power: sinusoidal case

F.Y.I.

The **average total power** of a **sinusoidal signal** is

$$f(t) = V_{max} \sin\left(\frac{2\pi t}{T}\right)$$

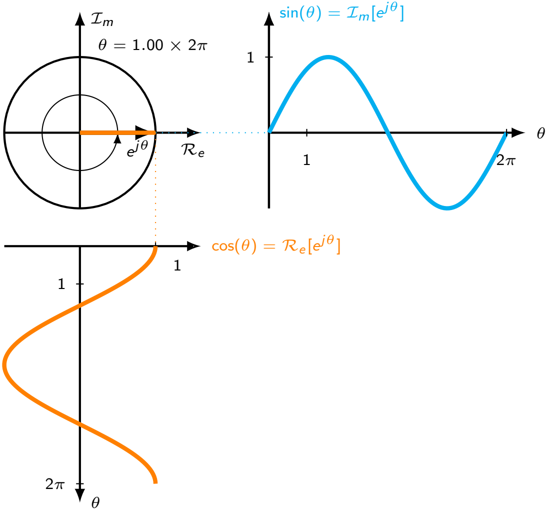
$$\begin{aligned} P = \frac{1}{T}\int_T |f(t)|^2 dt &= \frac{1}{T}\int_0^T V_{max}^2 \sin^2\left(\frac{2\pi t}{T}\right) dt \\ &= \frac{V_{max}^2}{2T}\int_0^T \left(1 - \cos\left(\frac{4\pi t}{T}\right)\right) dt \\ &= \frac{V_{max}^2}{2T}\left[t - \frac{\sin\left(\frac{4\pi t}{T}\right)}{\frac{4\pi}{T}}\right]_0^T = \frac{V_{max}^2}{2} \end{aligned}$$

This yields the **root mean square value** or "**rms**" value[18], i.e.

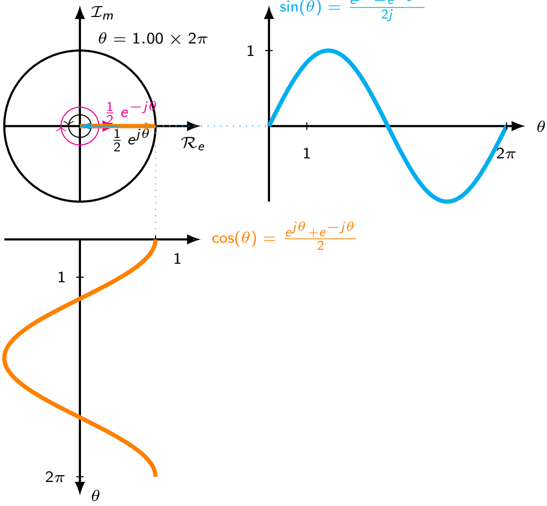
$$V_{rms} = \sqrt{P} = \frac{V_{max}}{\sqrt{2}}.$$

[18] **valeur efficace**

Sine and cosine: projection on imaginary and real axis



Sine and cosine: vector difference and sum



Complex exponentials

Complex exponentials

A **complex exponential** is a complex signal of the form

$$f(t) = A\, e^{at}, \quad -\infty < t < \infty$$

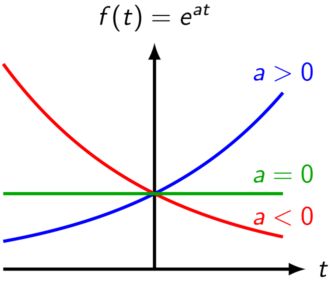
where $A = K\, e^{j\phi}$ and $a = \sigma_0 + j\omega_0$ are **complex numbers**, respectively in **polar** and **Cartesian form**.

Note that K and ϕ are respectively the **modulus** and **argument** of A. Similarly, σ_0 and ω_0 are, respectively, the **real** and **imaginary parts** of a.

It now **follows** that

$$\begin{aligned} f(t) &= A\, e^{at} = K\, e^{j\phi} e^{(\sigma_0 + j\omega_0)t} = K\, e^{j\phi}\, e^{\sigma_0 t}\, e^{j\omega_0 t} = K\, e^{\sigma_0 t} e^{j(\omega_0 t + \phi)} \\ &= K\, e^{\sigma_0 t}\, [\cos(\omega_0 t + \phi) + j \sin(\omega_0 t + \phi)]. \end{aligned}$$

Real exponentials[19]



Suppose that A and a are **real** then

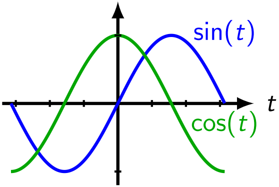
$$f(t) = A\,e^{at} = K\,e^{\sigma_0 t}, \;\; -\infty < t < \infty$$

is

- exponentially **decreasing** if $\sigma_0 = a < 0$,
- exponentially **increasing** if $\sigma_0 = a > 0$.

[19] A is **real** $\Rightarrow A = K, \phi = 0$ and a is **real** $\Rightarrow a = \sigma_0, \omega_0 = 0$

Purely imaginary exponentials[20]



Suppose that A is **real** and a is **purely imaginary**, i.e. $a = j\omega_0$, then

$$\begin{aligned} f(t) &= A\,e^{at} = K\,e^{j\omega_0 t} \\ &= K\,[\cos(\omega_0 t) + j\sin(\omega_0 t)], \\ &\qquad -\infty < t < \infty. \end{aligned}$$

Using the **Euler identity**, the following signals can be obtained

- $K\cos(\omega_0 t) = \frac{K}{2}[e^{j\omega_0 t} + e^{-j\omega_0 t}] = \mathcal{R}_e[f(t)]$,
- $K\sin(\omega_0 t) = \frac{K}{2j}[e^{j\omega_0 t} - e^{-j\omega_0 t}] = \mathcal{I}_m[f(t)]$.

[20] A is **real** $\Rightarrow A = K, \phi = 0$ and a is **purely imaginary** $\Rightarrow a = j\omega_0, \sigma_0 = 0$

Complex exponentials

The **complex exponential** $f(t) = A\,e^{at}$ with $A = K\,e^{j\phi}$ and $a = \sigma_0 + j\omega_0$ can be written as

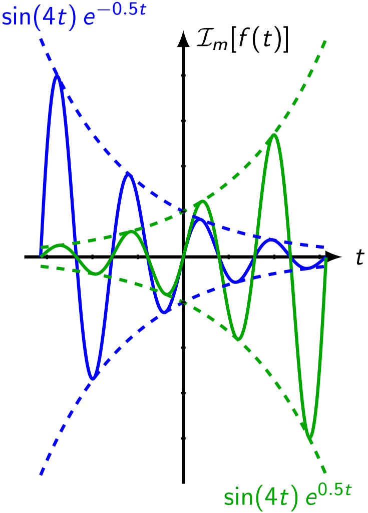
$$\boxed{f(t) = A\,e^{at} = K\,e^{\sigma_0 t}\left[\cos(\omega_0 t + \phi) + j\sin(\omega_0 t + \phi)\right]}$$

Note that, in order to construct a **real exponential modulated by a sine or cosine**, we need **<u>2 complex exponentials</u>**:

$$\mathcal{R}_e[f(t)] = K\,e^{\sigma_0 t}\,\cos(\omega_0 t + \phi) = \frac{1}{2}\left(f(t) + f^\star(t)\right) = \frac{1}{2}\left(Ae^{at} + A^\star e^{a^\star t}\right)$$

$$\mathcal{I}_m[f(t)] = K\,e^{\sigma_0 t}\,\sin(\omega_0 t + \phi) = \frac{1}{2j}\left(f(t) - f^\star(t)\right) = \frac{1}{2j}\left(Ae^{at} - A^\star e^{a^\star t}\right)$$

Complex exponentials



Consider the **complex exponential**

$$f(t) = A\, e^{at}$$

with A and a are **complex**, i.e.

$$A = K\, e^{j\phi} \quad \text{and} \quad a = \sigma_0 + j\omega_0.$$

Let us, for example, **study the signal**

$$\mathcal{I}_m[f(t)] = K\, e^{\sigma_0 t} \sin(\omega_0 t + \phi)$$

for $-\infty < t < \infty$.

This signal is

- an exponentially **decreasing** sinusoid if $\sigma_0 = \mathcal{R}_e[a] < 0$,
- an exponentially **increasing** sinusoid if $\sigma_0 = \mathcal{R}_e[a] > 0$.

3. Continuous-time systems

Lumped versus distributed
Linear time invariant systems
Representations
Convolution integral
First order system
Stability

Continuous-time system

Definition

A **continuous-time system** is a system in which the signals at its input and output are **continuous-time signals**.

Mathematically we represent it as an **operator** (transformation) H that converts an **input** signal $x(t)$ into an **output** signal $y(t)$, i.e.

$$x(t) \stackrel{H}{\Longrightarrow} y(t) = H[x(t)].$$

Input-output description

Definition

The **input-output description** of a system[21] gives a mathematical relation between the input and the output of a system.

Knowledge of the internal structure of the system is unavailable: access to the system is by means of input and output terminals.

In order to leave aside the initial conditions, we assume that system is **relaxed**[22] at time t_0, i.e. **no energy is stored** in that system at that instant.

$$x(t) \longrightarrow \boxed{H} \longrightarrow y(t)$$

[21]The system is described by the operator H. The symbol H is often used to make the link with the impulse response $h(t)$ of the system introduced later in the course. Often the symbol P is used instead to refer to a "process".

[22]le système est au repos.

Lumped versus distributed

F.Y.I.

- A **lumped system**[23] is a system in which the only dependent variable is time. In general, this means solving a set of **ordinary differential equations** (ODEs).
- A **distributed system**[24] is a system in which the dependent variables are functions of **time** and **one** or **more spatial** variables. In this case, this means solving **partial differential equations** (PDEs).

In this course, we will focus on **lumped** systems.

[23] Systèmes à constantes localisées
[24] Systèmes à constantes réparties

Lumped versus distributed: example

F.Y.I.

In general, the **temperature** in a body evolves with **time** as well as **space**.

- If we assume that the temperature of a body is **uniform** at all times then the temperature of the body is described by $T(t)$. An analysis of the heat transfer based on such an **idealisation** is called a **lumped system analysis**. The lumped system **approximation** can be used for **small bodies** of **highly conductive** materials.
- If we are interested in describing the heat transfer in **large** objects, i.e. walls, more **advanced mathematical tools** are required to solved the underlying **partial differential equations**.

Linearity

Definition

A **relaxed** system is **linear** if and only if

$$\mathbf{y} = H\left(\alpha_1 \mathbf{x_1} + \alpha_2 \mathbf{x_2}\right) = \alpha_1\, H(\mathbf{x_1}) + \alpha_2\, H(\mathbf{x_2})$$

for **any** inputs $\mathbf{x_1}$ and $\mathbf{x_2}$ and **any** real numbers α_1 and α_2.
Otherwise the system is **nonlinear**.

A linear system satisfies the **principle of superposition**.

Sinusoidal fidelity

If the input to a linear system is a **sinusoidal wave**, the output will **also** be a **sinusoidal wave**, and at exactly the **same frequency** as the input: only the **amplitude** and the **phase** will be **modified**, i.e.

$$\sin(\omega_0 t) \longrightarrow \boxed{H} \longrightarrow K(\omega_0)\, \sin\left(\omega_0 t + \Phi(\omega_0)\right)$$

Linearity and relaxedness

The **step response** of an **electrical RC system**[25] is

$$V_C(t) = \underbrace{\left(1 - e^{-\frac{t}{RC}}\right) E(t)\,u(t)}_{\text{zero-state response}} + \underbrace{V_C(0)\,e^{-\frac{t}{RC}}\,u(t)}_{\text{zero-input response}}$$

Note that the **linearity** presupposes **zero initial conditions**, i.e. a system **at rest** or a **relaxed** system.

Without relaxedness, the **superposition principle** does **not** hold !

If the response to input signal $E_1(t)$ is $V_{C_1}(t)$, it is easily checked that the response to the input signal $2\,E_1(t)$ is **not** $2\,V_{C_1}(t)$.

[25]This will be reviewed subsequently.

Nonlinear systems

- Nonlinear systems do **not** satisfy the **superposition principle** and do **not** follow **sinusoidal fidelity**.
- **Examples**:
  - **saturation** effects
  - systems described by **nonlinear differential equations**, etc.
- The dynamics of many processes that occur in the **real world** are sometimes dominated by **nonlinear effects** and are increasingly **exploited** in applications.
- LTI theory is still applicable after **linearisation** of the system around a given **working point**[26] or **trajectory**.

[26] See Fundamentals of control theory (RCH)

Causality

Definition

A system is **causal** or **non-anticipatory** if the output of the system at time t does not depend on the input applied after time t.

- The output at time t only depends on the input applied **before** and **at** time t.
- The past affects the future, but **not** conversely.
- If a system is causal, its input output relation can be written as

$$\mathbf{y}(t) = H\,\mathbf{u}_{(-\infty,\mathbf{t}]}$$

  for all t in $(-\infty, \infty)$.

Time invariance

Definition

A system is **time invariant** or **stationary** if the characteristics of the system do **not change with time**.

An experiment conducted at time t will yield the **same** result an hour later, the day after or a year later.

A **relaxed** system is **time invariant** if and only if

$$\begin{aligned} H\,D_\tau\,x(t) &= D_\tau\,H\,x(t) \\ H\,x(t-\tau) &= D_\tau\,y(t) = y(t-\tau) \end{aligned}$$

where D_τ is the **delay** or **shift operator**. Otherwise the system is **time-varying**.

Static system

- A system is said to be **static** if its output $\mathbf{y}(t)$ **depends only** on the **input** $\mathbf{x}(t)$ at the **present** time t, mathematically described as
$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t)).$$
- A static system is also referred to as a **memoryless system** since its output response $\mathbf{y}(t)$ is **not influenced** by the **past** of input $\mathbf{x}(\tau)$ where $\tau < t$.
- When subject to a **step-like input**, a **static** system will **settle out** <u>**instantaneously**</u>.
- **Examples**:
  - A purely resistive electrical circuit
  - A gain (linear)
  - A saturation (nonlinear)

Dynamic system

- Unlike a static system, the output $\mathbf{y}(t)$ of a **dynamic system** will be **affected by the input $\mathbf{x}(\tau)$** for $\tau \leq t$.
- When subject to a **step-like input**, a **dynamic system** will go through a **transient phase** and **take some time to settle-out**.
- Much can be **inferred** about the system from the **transient of a step response**, i.e. the step response is the "**signature**" of the system.
- **Examples**:
  - An electrical RC system
  - Any system that can be described by a **differential equation**

Differential equations

Continuous-time dynamic systems with lumped parameters are often representable by **linear** ordinary **differential equations** with **constant coefficients**.

Definition (1)

Given a **dynamic system** represented by a linear **differential** equation with **constant coefficients**,

$$a_0\, y(t) + a_1 \frac{dy(t)}{dt} + \ldots + a_n \frac{d^n y(t)}{dt^n} = b_0\, x(t) + b_1 \frac{dx(t)}{dt} + \ldots + b_m \frac{d^m x(t)}{dt^m}$$

with n **initial conditions** $y(0)$ and ${}^{d^k y(t)}/_{dt^k}\big|_{t=0}$ for $k = 1, \ldots, n-1$ and a **causal input** $x(t) = 0$ for $t < 0$.

Differential equations

Definition (2)

The complete response of the system for $t \geq 0$ has **two components**:

- The **zero-state** response[27], $y_{zs}(t)$, due exclusively to the input $x(t)$ as the initial conditions are zero.
- The **zero-input** response[28], $y_{zi}(t)$, due exclusively to the initial conditions as the input $x(t)$ is zero.

So that

$$y(t) = y_{zs}(t) + y_{zi}(t).$$

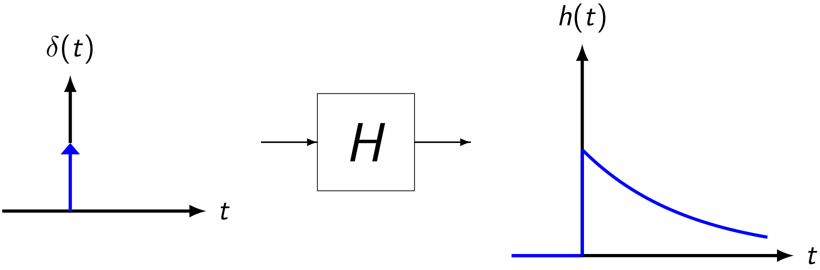
Thus, when the initial conditions are zero, i.e. the system is **relaxed**, then $y(t)$ depends exclusively on the input: $y(t) = y_{zs}(t)$. In such a case, the system is **Linear** and **Time Invariant** or **LTI**.

[27] la réponse **forcée**
[28] la réponse **libre**

Impulse response

Impulse response

The **impulse response**[29] $h(t)$ of a LTI system is its response to a **Dirac impulse** $\delta(t)$ given that **initial conditions** are **zero**, i.e. the system is **relaxed**.

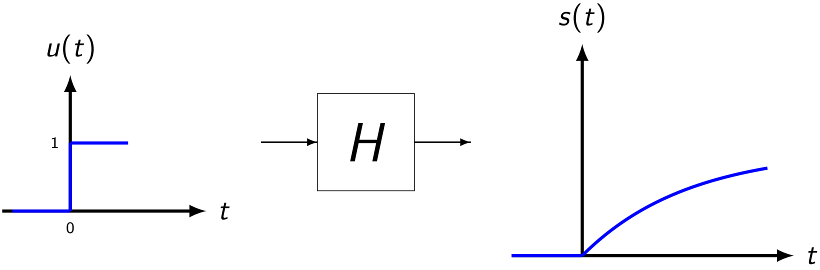


[29] Réponse impulsionnelle

Unit step response

Unit step response

The **step response**[30] $s(t)$ of a LTI system is its response to a **unit step input** $u(t)$ given that **initial conditions** are **zero**, i.e. the system is **relaxed**.



[30] Réponse indicielle ou réponse à un échelon unitaire

Convolution integral[31]

Definition

The **response** of a **relaxed LTI system represented by its impulse response** $h(t)$ to **any signal** $x(t)$ is the **convolution integral**

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,d\tau = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)\,d\tau \\ &= x(t) * h(t) = h(t) * x(t) \end{aligned}$$

where the symbol $*$ is used to denote the **convolution operation**.

We will see subsequently that the convolution integral is **often evaluated** using the **Laplace transform**.

[31] produit de convolution

Convolution integral

Reminder: generic representation of a continuous-time signal

$x(t) = \int_{-\infty}^{\infty} x(\tau)\,\delta(t-\tau)\,d\tau$

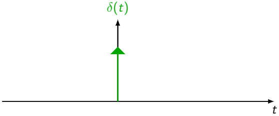
- **Hypothesis**: $h(t)$ is the impulse response of a relaxed system: the response to a $\delta(t)$ input Dirac impulse is $h(t)$.
- **Time invariance**: the response to $\delta(t-\tau)$ is $h(t-\tau)$.
- **Linearity**: the response to $x(\tau)\,\delta(t-\tau)$ is $x(\tau)\,h(t-\tau)$ given that $x(\tau)$ is independent of time t.
- **Superposition**: the response to $x(t) = \int_{-\infty}^{\infty} x(\tau)\,\delta(t-\tau)d\tau$ is:

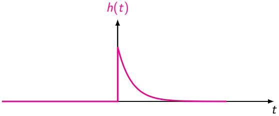
$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} x(t-\sigma)h(\sigma)d\sigma.$$

The last equality is obtained using the **substitution** $\sigma = t - \tau$.

Convolution integral: intuition

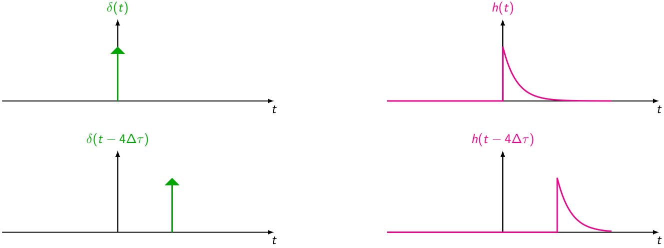
F.Y.I.





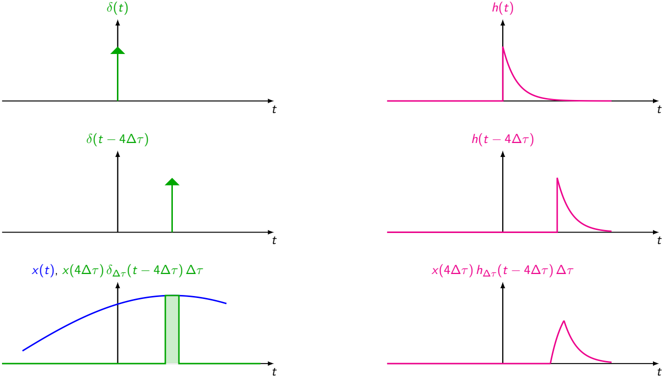
Convolution integral: intuition

F.Y.I.



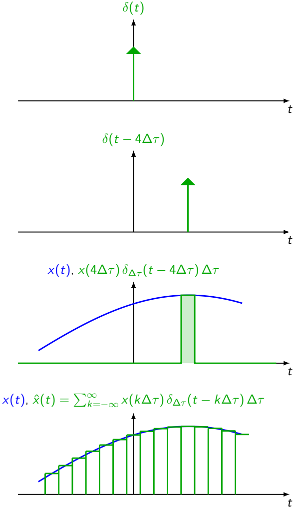
Convolution integral: intuition

F.Y.I.



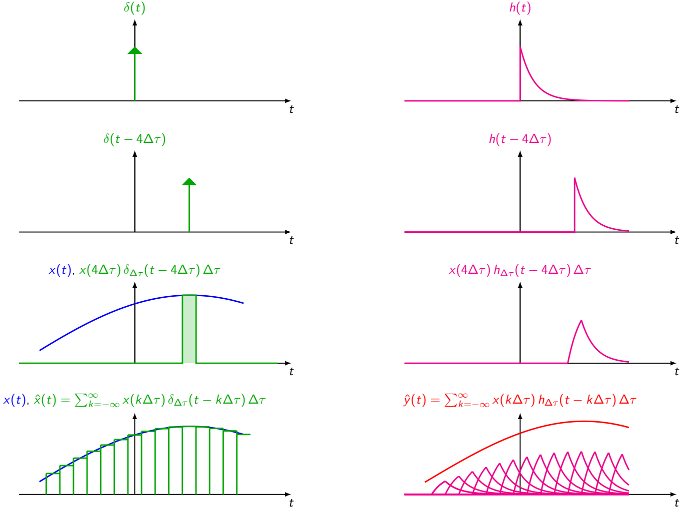
Convolution integral: intuition

F.Y.I.



Convolution integral: intuition

F.Y.I.



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

Convolution integral: step response

F.Y.I.

The **response** of a **relaxed LTI system represented by its impulse response** $h(t)$ to **any signal** $x(t)$ is the **convolution integral**

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

This can be applied to $x(t) = u(t)$ in which case we obtain the **step response**, i.e.

$$\boxed{s(t) = \int_{-\infty}^{t} h(\tau)\,d\tau}$$

showing that the **step response** can be obtained from the **impulse response** by **integrating** it over time.

Convolution integral: remarks

- The **impulse response** and the **step response** are **fundamental** in the **characterization** of **linear time-invariant systems**.
- Any system characterized by the convolution integral is linear and time invariant by construction. The convolution integral is a **general representation of LTI systems**, given that it was obtained from a generic representation of the input signal.
- A **system** represented by a **linear differential equation with constant coefficients** and no initial conditions, or input, before $t = 0$ is LTI. Thus, we should be able to represent that system by a **convolution integral** after finding its impulse response $h(t)$.

Convolution integral: causality

Causality

An LTI system is **causal** if

$$h(t) = 0 \text{ for } t < 0.$$

The **response** of a **causal system** to a **causal input** $x(t)$, i.e. $x(t) = 0$ for $t < 0$, is

$$y(t) = \int_0^t x(\tau)h(t-\tau)d\tau.$$

Convolution integral: causality

Convolution integral: general case

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

- $y(t) = \int_{-\infty}^{t} x(\tau)h(t-\tau)d\tau + \int_{t}^{\infty} x(\tau)h(t-\tau)d\tau$.
- **Causality of the system**: $y(t) = \int_{-\infty}^{t} x(\tau)h(t-\tau)d\tau$ given that $h(t-\tau) = 0$ for $\tau > t$.
- **Causality of the input**: $y(t) = \int_{0}^{t} x(\tau)h(t-\tau)d\tau$ given that $x(\tau) = 0$ for $\tau < 0$.

Convolution integral: properties

Properties

- **Commutativity**:

$$\boxed{f(t) * g(t) = g(t) * f(t)}$$

- **Distributivity**:

$$\boxed{[f(t) + g(t)] * h(t) = f(t) * h(t) + g(t) * h(t)}$$

- **Associativity**:

$$\boxed{[f(t) * g(t)] * h(t) = f(t) * [g(t) * h(t)]}$$

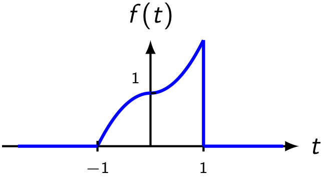
- **Identity element**:

$$\boxed{f(t) * \delta(t) = f(t)}$$

Convolution integral: support

Definition

The **support** of a signal $f(t)$ can be understood as the **smallest time interval of the signal outside of which the signal is always zero**, i.e. $f(t) = 0$.



Support of $f(t)$: $[-1, 1]$

Length of the support of $f(t)$: 2

Property

The **length of the support** of $y(t) = x(t) * h(t)$ is equal to the **sum of the lengths of the supports** of $x(t)$ and $h(t)$.

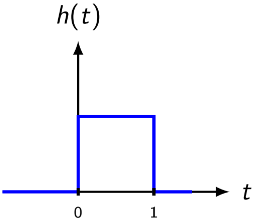
Graphical convolution

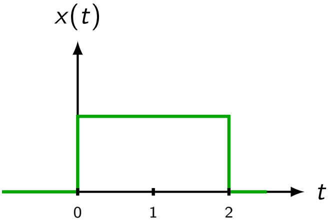
Graphical convolution

- **Express** $x(.)$ and $h(.)$ in terms of a **dummy variable** τ.
- **Reflect** the function $h(\tau) \longrightarrow h(-\tau)$.
- Add a **time-offset** t, which allows $h(t-\tau)$ to **slide** along the τ-axis.
- Start the time-offset t at 0 and slide it all the way to ∞. Wherever the functions $x(\tau)$ and $h(t-\tau)$ **intersect**, find the **integral** from 0 to t of their **product**.

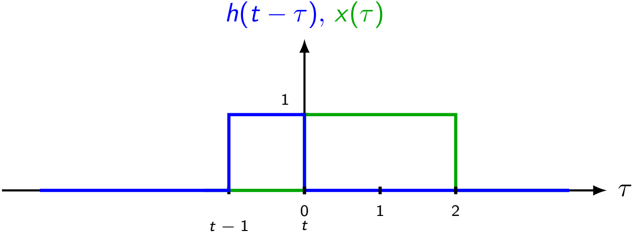
Graphical convolution: an example

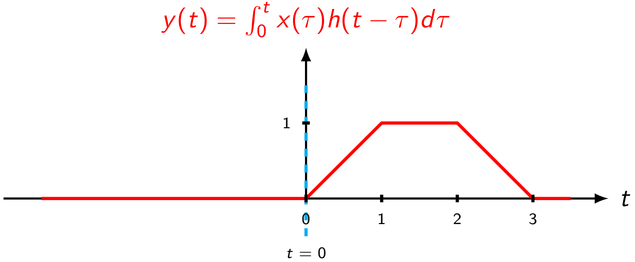
- **Impulse response**: $h(t) = u(t) - u(t-1)$
- **System input**: $x(t) = u(t) - u(t-2)$



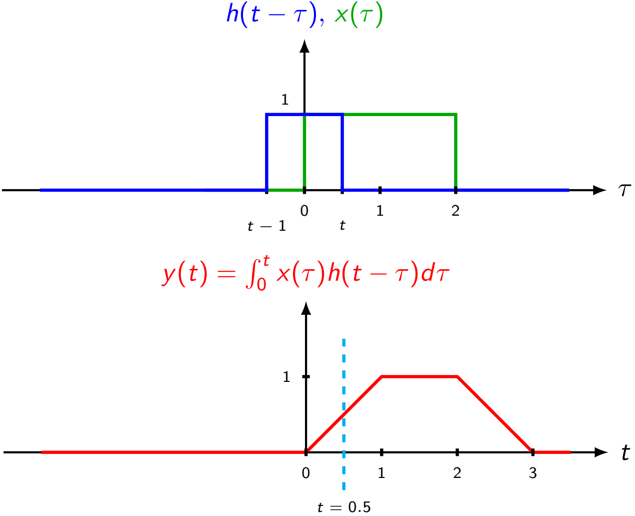


Graphical convolution: an example, $t = 0$



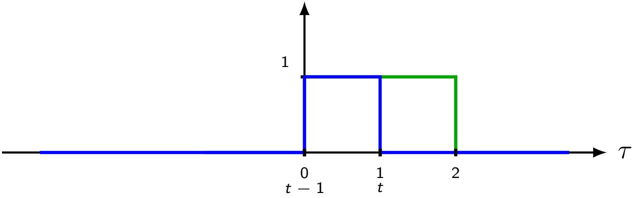


Graphical convolution: an example, $t = 0.5$

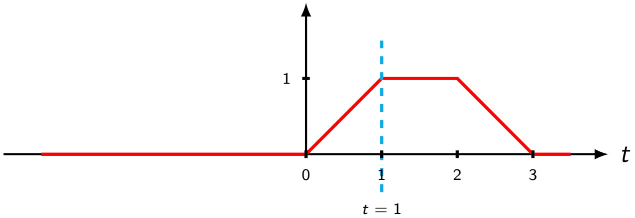


Graphical convolution: an example, $t = 1$

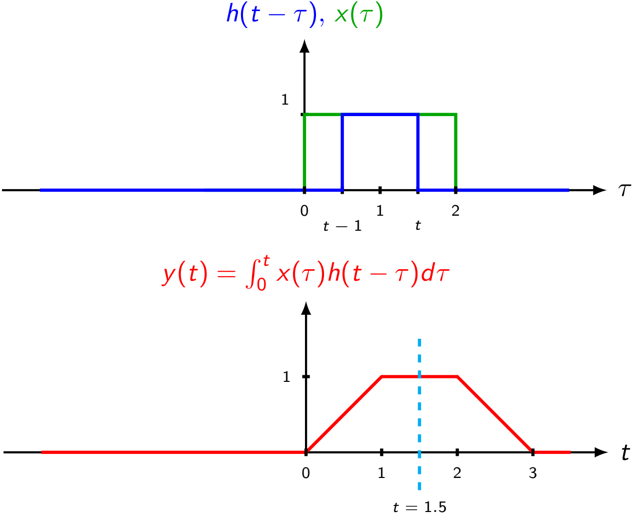
$h(t-\tau)$, $x(\tau)$



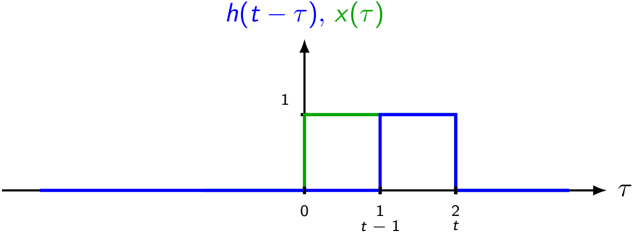
$y(t) = \int_0^t x(\tau)h(t-\tau)d\tau$

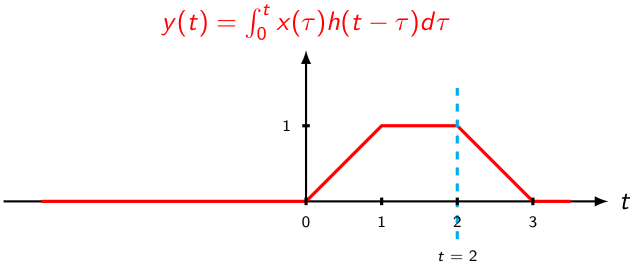


Graphical convolution: an example, $t = 1.5$

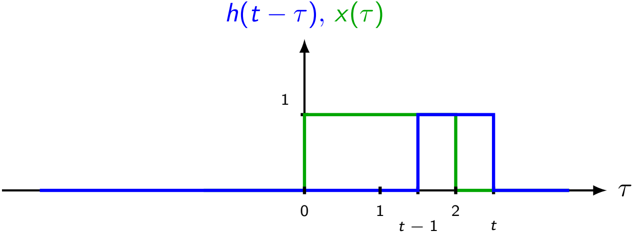


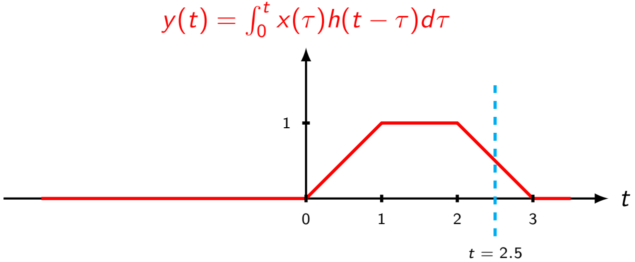
Graphical convolution: an example, $t = 2$



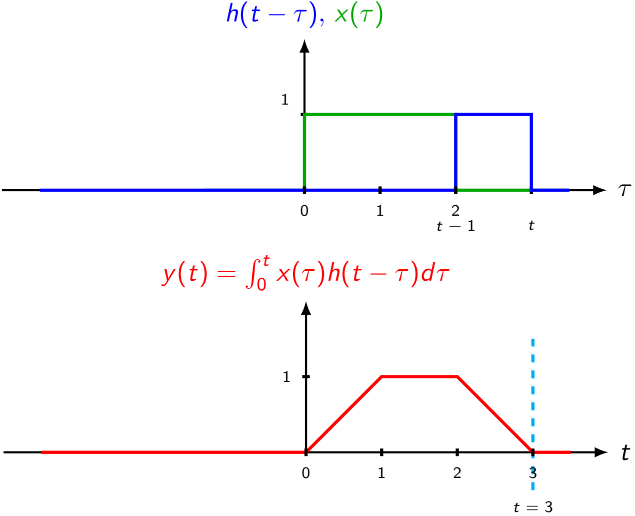


Graphical convolution: an example, $t = 2.5$

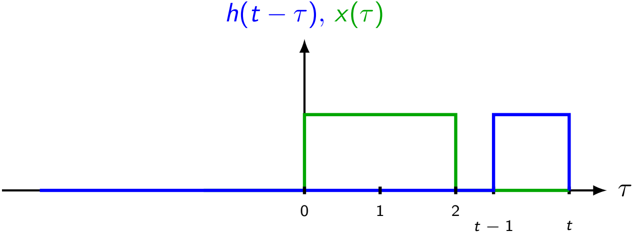


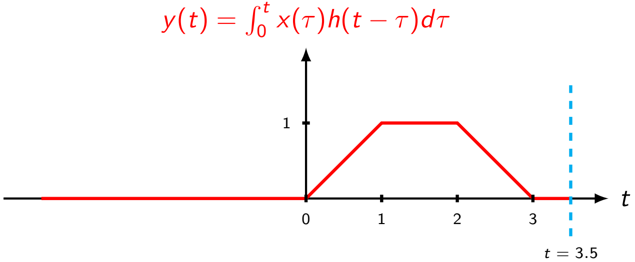


Graphical convolution: an example, $t = 3$

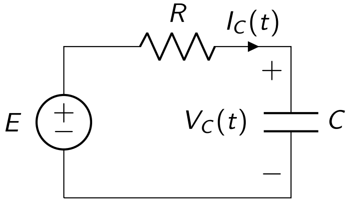


Graphical convolution: an example, $t = 3.5$





First order RC system: differential equation



$$I_C(t) = \frac{E(t) - V_C(t)}{R} = C\,\frac{dV_C(t)}{dt}$$

The **RC system** is described by the **first order differential equation**

$$\boxed{RC\,\frac{dV_C(t)}{dt} + V_C(t) = E(t)}$$

with $T = RC$ the **time constant** of the system.

First order system: differential equation

In general, a **first order system** is described by a **first order differential equation**

$$\boxed{T\,\frac{dy(t)}{dt} + y(t) = K\,x(t)}$$

with T the **time constant** of the system and K is the **static gain** of the system.

It is **obvious** that the **RC system** is a **first order system** with $T = RC$ and $K = 1$.

First order system: solution to differential equation

Let us first solve the **homogeneous** equation by setting the right side to zero.

$$\boxed{T\,\frac{dy(t)}{dt} + y(t) = 0}$$

The **solution** of this differential equation is

$$y(t) = C\,e^{-\frac{t}{T}}$$

which can be easily **verified** by **direct substitution**.

First order system: solution to differential equation

To obtain the **general response**, we let the coefficient C be a function of time

$$y(t) = C(t)\, e^{-\frac{t}{T}}$$

and **substitute** into the original equation. We obtain

$$T\left(C'(t)\, e^{-\frac{t}{T}} - \frac{C(t)}{T}\, e^{-\frac{t}{T}}\right) + C(t)\, e^{-\frac{t}{T}} = K\, x(t).$$

Simplifying

$$C'(t) = \frac{K}{T}\, e^{\frac{t}{T}}\, x(t)$$

which can be integrated to obtain

$$C(t) = C(0) + \int_0^t \frac{K}{T}\, e^{\frac{\tau}{T}}\, x(\tau)\, d\tau$$

with $C(0) = y(0)$.

First order system: solution to differential equation

The **general solution** is

$$y(t) = C(t)\, e^{-\frac{t}{T}}$$

with

$$C(t) = y(0) + \int_0^t \frac{K}{T}\, e^{\frac{\tau}{T}}\, x(\tau)\, d\tau.$$

After substitution, we obtain

$$\boxed{y(t) = \underbrace{\int_0^t \frac{K}{T}\, e^{-\frac{(t-\tau)}{T}}\, x(\tau)\, d\tau}_{\text{zero-state response}} + \underbrace{y(0)\, e^{-\frac{t}{T}}}_{\text{zero-input response}}}$$

First order system: impulse response

The **impulse response** of the **first order** system is the response with input $x(t) = \delta(t)$ with **zero initial conditions**, i.e

$$h(t) = \int_{0^-}^{t} \frac{K}{T}\, e^{-\frac{(t-\tau)}{T}}\, \delta(\tau)\, d\tau$$

Using the **sifting property** of the Dirac impulse, we obtain

$$\boxed{h(t) = \frac{K}{T}\, e^{-\frac{t}{T}}\, u(t)}$$

First order system: convolution integral

The **general response** of the **first order** system is the response with input $x(t)$ with **zero initial conditions** is

$$y(t) = \int_0^t \frac{K}{T}\, e^{-\frac{(t-\tau)}{T}}\, x(\tau)\, d\tau.$$

Notice that this response corresponds to the **convolution** of the **impulse response**

$$h(t) = \frac{K}{T}\, e^{-\frac{t}{T}}\, u(t)$$

with the input $x(t)$, i.e.

$$\boxed{y(t) = x(t) * h(t) = \int_0^t x(\tau)\, h(t-\tau)\, d\tau}$$

First order system: step response

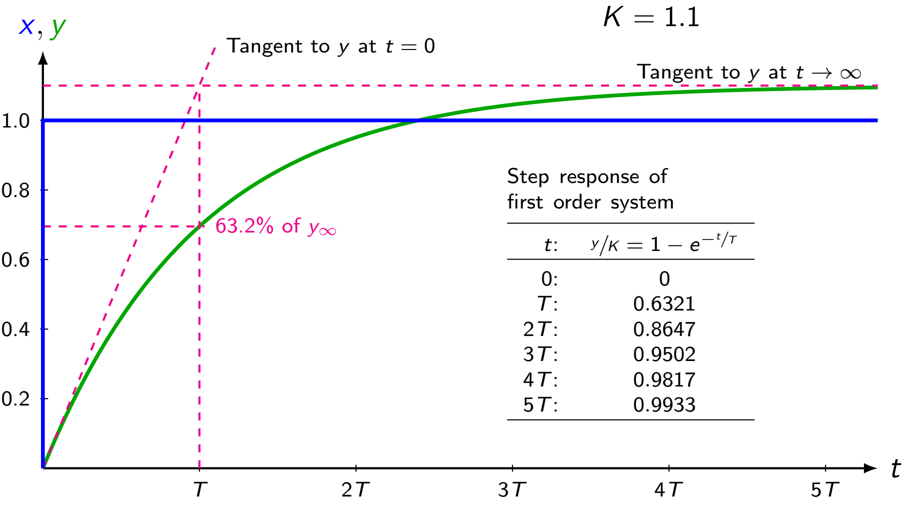
The **step response** of the **first order** system is the response with input $x(t) = u(t)$ with **zero initial conditions**, i.e

$$s(t) = \int_0^t \frac{K}{T}\, e^{-\frac{(t-\tau)}{T}}\, d\tau = \left[K\, e^{-\frac{(t-\tau)}{T}}\right]_0^t$$

We obtain

$$\boxed{s(t) = K\, \left(1 - e^{-\frac{t}{T}}\right)\, u(t)}$$

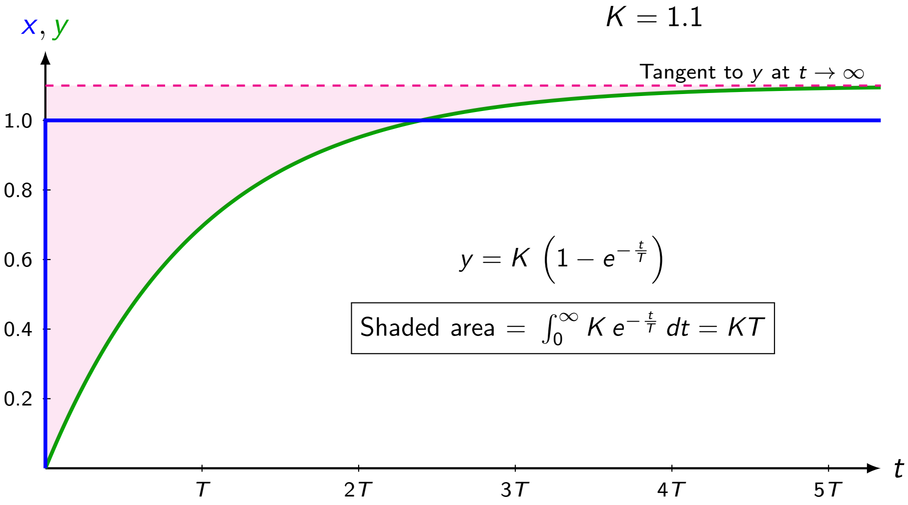
First order system: step response



Step response of first order system

t:	$y/K = 1 - e^{-t/T}$
0:	0
T:	0.6321
$2T$:	0.8647
$3T$:	0.9502
$4T$:	0.9817
$5T$:	0.9933

First order system: step response



BIBO stability

BIBO stability

Bounded-Input-Bounded-Output (BIBO) stability[32] establishes that for a **bounded input** $x(t)$, the **output** of a BIBO stable system $y(t)$ is **also bounded**.

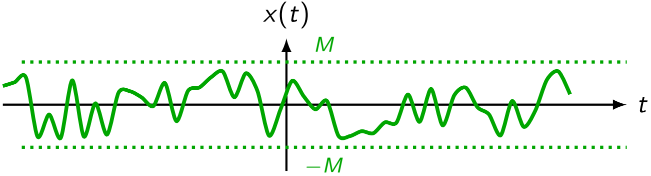
This means that if there exists a **finite bound** $M \leq \infty$ such that $|x(t)| \leq M$ for all t, the **output** is also **bounded**.

Note that $|x(t)| \leq M$ for all t means $x(t) \in [-M, M]$ for all t.

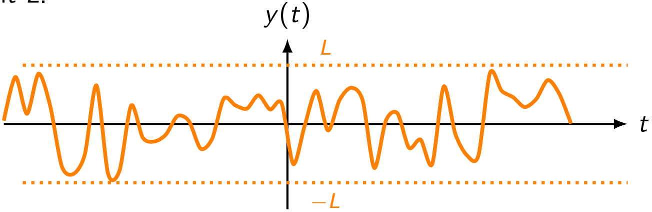
[32] En français, on utilise parfois la notion de stabilité EBSB, c.-à-d. à une Entrée Bornée correspond une Sortie Bornée.

BIBO stability

If the system is **BIBO stable**, any **bounded input** $x(t)$, i.e. $-M \leq x(t) \leq M$, $\forall t$ for some constant M



will produce a **bounded output** $y(t)$, i.e. $-L \leq y(t) \leq L$, $\forall t$ for some constant L.



BIBO stability

BIBO stability condition in the time domain

An **LTI** system is **BIBO stable if and only if**

$$\int_{-\infty}^{\infty} |h(\tau)|\, d\tau < \infty,$$

i.e. its **impulse response is absolutely integrable**.

A **causal LTI** system is **BIBO stable if and only if**

$$\int_{0}^{\infty} |h(\tau)|\, d\tau < \infty.$$

A **simpler way** to **test the BIBO stability** of a system using the **Laplace transform** is given later.

BIBO stability: sufficiency

For an LTI system with impulse response $h(t)$ to be **BIBO stable**, its response to a **bounded** input $x(t)$,

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

needs to be **bounded**. This implies

$$\begin{aligned} |y(t)| &= \left| \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \right| \\ &\leq \int_{-\infty}^{\infty} |h(\tau)||x(t-\tau)|d\tau \\ &\leq \int_{-\infty}^{\infty} |h(\tau)|\, M\, d\tau = M \int_{-\infty}^{\infty} |h(\tau)|d\tau < \infty \end{aligned}$$

i.e. $h(t)$ needs to be **absolutely integrable**[33].

[33]Here we have only proved the **sufficiency** of the stability condition. It is also possible to prove its **necessity** although this is **more complicated**.

4. Laplace transform

Definitions
Laplace transform computations
Laplace transform and non causal signals
Properties of the Laplace transform
Transfer function
One-sided Laplace transforms
Stability in the Laplace domain
Dynamic behaviour
Initial and final value theorems
Inverse of one-sided Laplace transforms
Analysis of LTI systems
Frequency response
First order system with delay
Bode diagram animation
Filtering
Matlab and Octave
One-sided Laplace transforms
Basic properties of one-sided Laplace transforms

Two-sided Laplace transform

Definition

The **two-sided Laplace transform** of a continuous signal $f(t)$ is defined as

$$F(s) = \mathcal{L}[f(t)] = \int_{-\infty}^{\infty} f(t)\, e^{-st} dt, \quad s \in \mathrm{ROC}$$

where $s = \sigma + j\omega$, with ω is the **frequency**[34] expressed in [rad/sec] and σ is a **damping factor**.

ROC stands for the **Region Of Convergence**, i.e. the region where the integral exists.

[34] In **English**, frequency is used both for ω expressed in [rad/s] and f expressed in [Hz]. The difference is clear from the context. Sometimes, angular frequency is used for ω. In **French**, “pulsation” and “fréquence” are respectively used for ω and f.

Inverse Laplace transform

Definition

The **inverse Laplace transform** is given by

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} F(s)\, e^{st}\, ds, \quad s \in \mathrm{ROC}$$

The computation of the **inverse Laplace transform** using this equation would require **complex integration**. Algebraic methods will be used later to find the inverse Laplace transform, thus **avoiding** the complex integration.

Two-sided Laplace transform: remarks

- The Laplace transform $F(s)$ provides a **representation** of $f(t)$ in the s-domain, which in turn can be converted back into the original time-domain function in a **one-to-one** manner using the region of convergence. Thus,

$$F(s) \quad s \in \text{ROC} \iff f(t)$$

- If $f(t) = h(t)$, the **impulse response** of an LTI system, then $H(s)$ is called the system or **transfer function**[35] of the system and it **characterizes the system** in the s-domain just like $h(t)$ does in the time-domain.
- If $f(t)$ is a **signal**, then $F(s)$ is its Laplace transform.

[35] The notion of a transfer function will be explained in more detail later.

Fourier series, Laplace and Fourier transforms[36]

- The Fourier series of a **periodic signal** of period T is

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{jk\omega_0 t}, \;\; \omega_0 = \frac{2\pi}{T}$$

- The **inverse Fourier transform** is given by

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

- The **inverse Laplace transform** is given by

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

[36] The Fourier series and transform will be introduced **later** in the course.

Fourier series, Laplace and Fourier transforms

- The **Fourier series** and the **Fourier transform decompose** the signal $f(t)$ into an **infinite sum**, i.e. a **discrete spectrum** for the Fourier series and a **continuous spectrum** for the Fourier transform, of **sinusoidal type signals** of **varying frequency**, i.e. they give us an idea on how to **weigh** the different sinusoids of varying frequencies that make up the signal $f(t)$.
- The **Laplace transform** has the **same interpretation** but it is **more general** as it decomposes the signal $f(t)$ as a **sum of complex exponentials** which can be seen as **damped sinusoids** of **varying frequency** and **varying damping**.

One-sided Laplace transform

Definition

The **one-sided Laplace transform** of a continuous signal $f(t)$ is defined as

$$F(s) = \mathcal{L}[f(t)u(t)] = \int_{0^-}^{\infty} f(t)e^{-st}dt, \;\; s \in \mathrm{ROC}$$

where $f(t)$ is either **causal** or non-causal and **made into a causal function** by multiplication by $u(t)$.

One-sided Laplace transform: properties

- The **one-sided** Laplace transform is of **significance** given that most of the applications deal with **causal systems and signals**.
- Notice that when $f(t)$ is causal, the two-sided and the one-sided Laplace transforms of $f(t)$ **coincide**.
- The **lower limit** of the integral in the one-sided Laplace transform is set to $0^- = 0 - \epsilon$, which corresponds to a value on the **left side of zero for an infinitesimal value** ϵ.
- The reason for this is to make sure that the Laplace transform of a **Dirac impulse** $\delta(t)$ is defined. For **any other signal** this limit can be taken as **zero** with no effect on the transform.
- As we will see, the **advantage** of the **one-sided** Laplace transform is that it can be used in the solution of differential equations with **initial conditions** at $t = 0$.

Region of convergence

Region of convergence

The **region of convergence** of a **causal** function $f(t)$ are the values of σ for which the **integral converges**. The frequency ω does **not** affect the ROC.

$$\boxed{\text{ROC } = \left\{ s = \sigma + j\omega \text{ such that } \int_0^{\infty} |f(t)e^{-\sigma t}| \, dt < \infty \right\}}$$

Region of convergence

In order for the Laplace transform of $f(t)$ to **exist**, we **need**

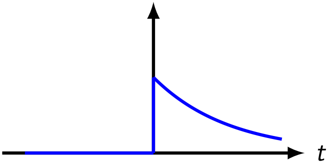
$$\begin{aligned}\left|\int_0^\infty f(t)e^{-st}\,dt\right| &= \left|\int_0^\infty \left(f(t)e^{-\sigma t}\right)e^{-j\omega t}\,dt\right|,\\ &\leq \int_0^\infty |f(t)e^{-\sigma t}||e^{-j\omega t}|\,dt,\\ &= \int_0^\infty |f(t)e^{-\sigma t}|\,dt < \infty,\end{aligned}$$

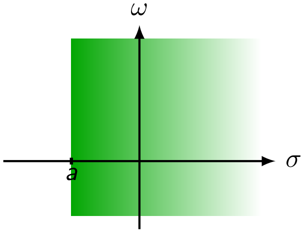
i.e. $f(t)e^{-\sigma t}$ has to be **absolutely integrable**[37]. This is often possible by an **appropriate** choice of σ even if $f(t)$ is **not** itself absolutely integrable.

[37] Here we have only proved the **sufficiency** of this condition. It is also possible to prove its **necessity** although this is **more complicated**.

Region of convergence of a causal function

$f(t) = e^{at}\, u(t), a < 0$





Remember that

$$\text{ROC } = \left\{ s = \sigma + j\omega \text{ such that } \int_0^{\infty} |f(t)e^{-\sigma t}|\, dt < \infty \right\}$$

The integral

$$\int_0^{\infty} |e^{at}\, u(t)\, e^{-\sigma t}|\, dt = \int_0^{\infty} e^{-(\sigma - a)t}\, dt$$

converges for $\sigma > a$.

Dirac impulse

The Laplace transform of a **Dirac impulse** $\delta(t)$ is

$$\mathcal{L}[\delta(t)] = \int_{0^-}^{\infty} \delta(t)e^{-st}dt = \int_{0^-}^{\infty} \delta(t)e^{-s0}dt = \int_{0^-}^{\infty} \delta(t)dt = 1$$

There are **no restrictions** on the region of convergence, i.e. the ROC is the whole s-plane.

Unit step

The Laplace transform of a **unit step** $u(t)$ is

$$\begin{aligned}\mathcal{L}[u(t)] &= \int_0^\infty u(t)e^{-st}dt = \int_0^\infty e^{-st}dt \\ &= \left.\frac{e^{-st}}{-s}\right|_0^\infty = \frac{1}{s}\end{aligned}$$

The integral converges if $\mathcal{R}_e[s] = \sigma > 0$. The **ROC** is the **open right half** s**-plane**.

Unit ramp

The Laplace transform of a **unit ramp** $r(t) = t\,u(t)$ is

$$\begin{aligned}\mathcal{L}[r(t)] &= \int_0^\infty t\,u(t)\,e^{-st}dt = \int_0^\infty t\,e^{-st}dt \\ &= \left.\frac{-t\,e^{-st}}{s}\right|_0^\infty + \int_0^\infty \frac{e^{-st}}{s}\,dt = 0 - \left.\frac{e^{-st}}{s^2}\right|_0^\infty = \frac{1}{s^2}\end{aligned}$$

The integral[38] converges if $\mathcal{R}_e[s] = \sigma > 0$. The **ROC** is the **open right half** s**-plane**.

[38] integration by parts !

Real exponential (1)

The Laplace transform of a **real exponential** $e^{at}u(t)$:

$$\begin{aligned} \mathcal{L}[e^{at}u(t)] &= \int_0^\infty e^{at}u(t)e^{-st}dt = \int_0^\infty e^{-(s-a)t}dt \\ &= \frac{-1}{s-a}\, e^{-(s-a)t}\Big|_0^\infty = \frac{1}{s-a}. \end{aligned}$$

The integral converges if $\mathcal{R}_e[s] = \sigma > a$. The **ROC** is the open half plane to the right of the axis $\mathcal{R}_e[s] = a$.

The result is **<u>identical</u>** for $a \in \mathbb{C}$ with **ROC** $\mathcal{R}_e[s] = \sigma > \mathcal{R}_e[a]$. More in particular,

$$\mathcal{L}[e^{j\omega_0 t}u(t)] = \frac{1}{s - j\omega_0}$$

with **ROC** $\mathcal{R}_e[s] = \sigma > 0$.

Real exponential (2)

The Laplace transform of a **real exponential** $\frac{K}{T}\, e^{-\frac{t}{T}} u(t)$ is

$$\begin{aligned}\mathcal{L}[\tfrac{K}{T}\, e^{-\frac{t}{T}} u(t)] &= \frac{K}{T}\int_0^\infty e^{-\frac{t}{T}} u(t) e^{-st} dt = \frac{K}{T}\int_0^\infty e^{-(Ts+1)\frac{t}{T}} dt \\ &= \frac{-K}{Ts+1}\, e^{-(Ts+1)\frac{t}{T}}\Big|_0^\infty = \frac{K}{Ts+1}.\end{aligned}$$

The integral converges if $\mathcal{R}_e[s] = \sigma > -\frac{1}{T}$. The **ROC** is the open half plane to the right of the axis $\mathcal{R}_e[s] = -\frac{1}{T}$.

One-sided Laplace transforms[39]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$		
$\mathcal{L}[\delta(t)] = 1$ $\mathcal{L}[u(t)] = \frac{1}{s}$ $\mathcal{L}[t\,u(t)] = \frac{1}{s^2}$ $\mathcal{L}[e^{at}\,u(t)] = \frac{1}{s-a}$ $\mathcal{L}[e^{j\omega_0 t}\,u(t)] = \frac{1}{s-j\omega_0}$		

[39] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

Poles and zeros

Poles and zeros

Suppose $F(s) = \mathcal{L}[f(t)] = \frac{n(s)}{d(s)}$ is a **rational** function with $n(s)$ and $d(s)$ **polynomials** in s,

- A **zero** of $F(s)$ is a value of s for which $F(s) = 0$, i.e. a value of s for which $n(s) = 0$.
- A **pole** of $F(s)$ is a value of s for which $F(s) \to \infty$., i.e. value of s for which $d(s) = 0$.

The poles and zeros of $F(s)$ can be **complex**[40].

Remark: By definition, **no poles are included in the ROC**.

[40] They come in **complex conjugate pairs** as we will consider $n(s)$ and $d(s)$ with **real** coefficients.

Region of convergence and poles

If $\{\sigma_i\}$ are the **real parts** of the **poles** of $F(s) = \mathcal{L}[f(t)]$ then:

- If $f(t)$ has **finite support**, i.e. $f(t) = 0$ for $t < t_1$ and $t > t_2$ and $t_1 < t_2$, then
  ROC = **whole s-plane**.
- If $f(t)$ is **causal**, i.e. $f(t) = 0$ for $t < 0$, then
  ROC $= \{(\sigma, \omega) : \sigma > \sigma_{max} = \max\{\sigma_i\}, -\infty < \omega < \infty\}$.

Definition

F.Y.I.

Two-sided and one-sided Laplace transforms

The **two-sided** Laplace transform of a **non causal** signal

$$f(t) = f_{ac}(t) + f_c(t) = f(t)u(-t) + f(t)u(t)$$

can be expressed using the **one-sided** Laplace transform

$$F(s) = \mathcal{L}[f_{ac}(-t)]_{(-s)} + \mathcal{L}[f_c(t)]$$

The **region of convergence** of $F(s)$ is $\text{ROC} = \text{ROC}_c \cap \text{ROC}_{ac}$.
Here ROC_c is the region of convergence of $\mathcal{L}[f(t)\,u(t)]$ and ROC_{ac} the region of convergence of $\mathcal{L}[f(t)\,u(-t)]$.

Causal and anti-causal decomposition

F.Y.I.

- A **non-causal signal** can be **decomposed** as

$$f(t) = f_{ac}(t) + f_c(t) = f(t)u(-t) + f(t)u(t)$$

- At time $t = 0$, $u(0) = 0.5$ **insures** $f(0) = f_{ac}(0) + f_c(0)$.
- The **two-sided** Laplace transform of $f(t)$ is

$$\begin{aligned} F(s) &= \int_{-\infty}^{0} f(t)u(-t)\, e^{-st}\, dt + \int_{0}^{\infty} f(t)u(t)\, e^{-st}\, dt \\ &= \int_{0}^{\infty} f(-t)u(t)\, e^{st}\, dt + \int_{0}^{\infty} f(t)u(t)\, e^{-st}\, dt \\ &= \mathcal{L}[f(-t)u(t)]_{(-s)} + \mathcal{L}[f(t)u(t)] = \mathcal{L}[f_{ac}(-t)]_{(-s)} + \mathcal{L}[f_c(t)] \end{aligned}$$

- The **two-sided** Laplace transform can be computed using **only** the **one-sided transform** with ROC the **intersection** of the ROCs of the causal and the anti-causal Laplace transforms.

Region of convergence and poles

F.Y.I.

If $\{\sigma_i\}$ are the **real parts** of the **poles** of $F(s) = \mathcal{L}[f(t)]$ then:

- If $f(t)$ is **anti-causal**, i.e. $f(t) = 0$ for $t > 0$ the ROC $= \{(\sigma, \omega) : \sigma < \sigma_{min} \ = min\{\sigma_i\}, -\infty < \omega < \infty\}$.
- If $f(t)$ is **non-causal**, i.e. $f(t) = f_c(t) + f_{ac}(t)$ then ROC $=$ ROC$_c \cap$ ROC$_{ac}$.

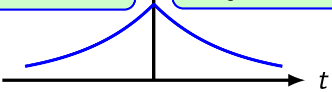
ROC of a non-causal function: example

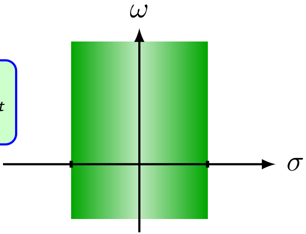
F.Y.I.

$$f(t) = e^{-a|t|} = e^{at}u(-t) + e^{-at}u(t), a > 0$$

$t \leq 0, f(t) = e^{at}$
$\int_{-\infty}^{0} e^{at}e^{-\sigma t}dt = \int_{-\infty}^{0} e^{(a-\sigma)t}dt$
converges for $\sigma < a$

$t \geq 0, f(t) = e^{-at}$
$\int_{0}^{\infty} e^{-at}e^{-\sigma t}dt = \int_{0}^{\infty} e^{-(a+\sigma)t}dt$
converges for $\sigma > -a$

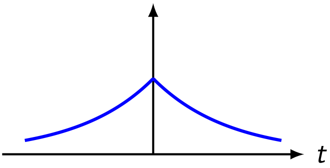


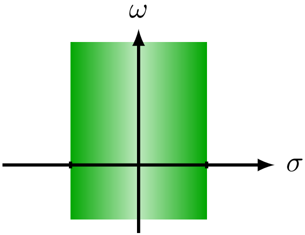


Non-causal function: example

F.Y.I.

$f(t) = e^{-a|t|}, a > 0$





The function $f(t)$ can be **decomposed** in its **causal** and **anti-causal** components, i.e.

$$f(t) = e^{-a|t|} = e^{at}u(-t) + e^{-at}u(t) = f_{ac}(t) + f_c(t).$$

The Laplace transform $F(s)$ can be **written** as

$$F(s) = \mathcal{L}[f(t)] = \mathcal{L}[f_{ac}(-t)]_{(-s)} + \mathcal{L}[f_c(t)].$$

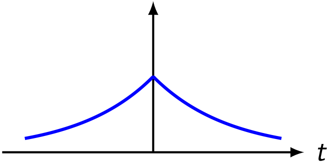
This **yields**

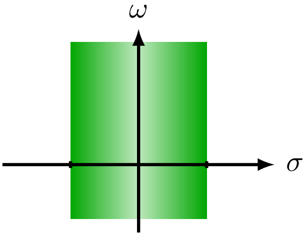
$$\mathcal{L}[e^{-a|t|}] = \mathcal{L}[e^{-at}u(t)]_{(-s)} + \mathcal{L}[e^{-at}u(t)].$$

Non-causal function: example

F.Y.I.

$f(t) = e^{-a|t|}, a > 0$





Remember that

$$\mathcal{L}[e^{-at}u(t)] = \frac{1}{s+a}$$

The Laplace transform $F(s)$ can therefore be **written** as

$$\begin{aligned} F(s) &= \mathcal{L}[e^{-at}u(t)]_{(-s)} + \mathcal{L}[e^{-at}u(t)], \\ &= \frac{1}{-s+a} + \frac{1}{s+a} = \frac{-2a}{s^2-a^2} = F_{ac}(s) + F_c(s). \end{aligned}$$

Linearity

Linearity

For signals $f(t)$ and $g(t)$ with **Laplace transforms**

- $\mathcal{L}[f(t)] = F(s)$,
- $\mathcal{L}[g(t)] = G(s)$ and
- and constants α and $\beta \in \mathbb{C}$

we have

$$\mathcal{L}[\alpha f(t) + \beta g(t)] = \alpha \, \mathcal{L}[f(t)] + \beta \, \mathcal{L}[g(t)] = \alpha F(s) + \beta G(s)$$

with ROC the **intersection of the regions of convergence** of $\mathcal{L}[f(t)]$ and $\mathcal{L}[g(t)]$.

Cosine: Laplace transform

Remember that

$$\mathcal{L}[e^{j\omega_0 t}u(t)] = \frac{1}{s - j\omega_0}$$

with **ROC** $\mathcal{R}_e[s] = \sigma > 0$. The **region of convergence is the open right-half plane s-plane**.

Using **Euler's identity** and the **linearity** of the Laplace transform, we obtain

$$\begin{aligned}
\mathcal{L}[\cos(\omega_0 t)u(t)] &= \mathcal{L}\left[{}^1/_2\left(e^{j\omega_0 t}u(t) + e^{-j\omega_0 t}u(t)\right)\right] \\
&= {}^1/_2\left(\mathcal{L}[e^{j\omega_0 t}u(t)] + \mathcal{L}[e^{-j\omega_0 t}u(t)]\right) \\
&= \frac{1}{2}\left(\frac{1}{s - j\omega_0} + \frac{1}{s + j\omega_0}\right) \\
&= \frac{1}{2}\ \frac{(s + j\omega_0) + (s - j\omega_0)}{s^2 + \omega_0^2} = \frac{s}{s^2 + \omega_0^2}
\end{aligned}$$

with **region of convergence the open right-half plane s-plane**.

Sine: Laplace transform

Remember that

$$\mathcal{L}[e^{j\omega_0 t}u(t)] = \frac{1}{s - j\omega_0}$$

with **ROC** $\mathcal{R}_e[s] = \sigma > 0$. The **region of convergence is the open right-half plane s-plane**.

Using **Euler's identity** and the **linearity** of the Laplace transform, we obtain

$$\begin{aligned}
\mathcal{L}[\sin(\omega_0 t)u(t)] &= \mathcal{L}\left[{}^1/_{2j}\left(e^{j\omega_0 t}u(t) - e^{-j\omega_0 t}u(t)\right)\right] \\
&= {}^1/_{2j}\left(\mathcal{L}[e^{j\omega_0 t}u(t)] - \mathcal{L}[e^{-j\omega_0 t}u(t)]\right) \\
&= \frac{1}{2j}\left(\frac{1}{s - j\omega_0} - \frac{1}{s + j\omega_0}\right) \\
&= \frac{1}{2j}\,\frac{(s + j\omega_0) - (s - j\omega_0)}{s^2 + \omega_0^2} = \frac{\omega_0}{s^2 + \omega_0^2}
\end{aligned}$$

with **region of convergence the open right-half plane s-plane**.

One-sided Laplace transforms[41]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$		
$\mathcal{L}[\delta(t)] = 1$ $\mathcal{L}[u(t)] = \frac{1}{s}$ $\mathcal{L}[t\,u(t)] = \frac{1}{s^2}$ $\mathcal{L}[e^{at}\,u(t)] = \frac{1}{s-a}$ $\mathcal{L}[e^{j\omega_0 t}\,u(t)] = \frac{1}{s-j\omega_0}$ $\mathcal{L}[\sin(\omega_0 t)\,u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$ $\mathcal{L}[\cos(\omega_0 t)\,u(t)] = \frac{s}{s^2+\omega_0^2}$		

[41] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

Frequency shifting

Frequency shifting

For a function $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$ and a **real**[42] λ,

$$\begin{aligned} F(s-\lambda) &= \int_0^\infty f(t)\, e^{-(s-\lambda)t} dt = \int_0^\infty \left(f(t)e^{\lambda t}\right) e^{-st} dt \\ &= \mathcal{L}[f(t)\, e^{\lambda t}] \end{aligned}$$

with ROC the region of convergence of $\mathcal{L}[f(t)u(t)]$ **shifted** λ to the **right** if $\lambda > 0$ and to the **left** if $\lambda < 0$.

[42] Note that property holds also for **complex** λ.

Frequency shifting

The **frequency shifting** property can be used to compute (old and) **new transforms**, i.e.

- $\mathcal{L}[e^{at}u(t)] = \dfrac{1}{(s-a)}$, a real
- $\mathcal{L}[t\,e^{at}u(t)] = \dfrac{1}{(s-a)^2}$, a real
- $\mathcal{L}[e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)] = \dfrac{\omega_0}{(s-\sigma_0)^2+\omega_0^2} = \dfrac{\omega_0}{(s-\sigma_0-j\omega_0)(s-\sigma_0+j\omega_0)}$
- $\mathcal{L}[e^{\sigma_0 t}\cos(\omega_0 t)\,u(t)] = \dfrac{s-\sigma_0}{(s-\sigma_0)^2+\omega_0^2} = \dfrac{s-\sigma_0}{(s-\sigma_0-j\omega_0)(s-\sigma_0+j\omega_0)}$

One-sided Laplace transforms[43]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\, e^{\lambda t} u(t)] = F(s-\lambda)$	
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{(s-a)}$	
$\mathcal{L}[t\, u(t)] = \frac{1}{s^2}$	$\mathcal{L}[t\, e^{at}\, u(t)] = \frac{1}{(s-a)^2}$	
$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{s-a}$		
$\mathcal{L}[e^{j\omega_0 t}\, u(t)] = \frac{1}{s-j\omega_0}$		
$\mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2}$	
$\mathcal{L}[\cos(\omega_0 t)\, u(t)] = \frac{s}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \cos(\omega_0 t)\, u(t)] = \frac{s-\sigma_0}{(s-\sigma_0)^2+\omega_0^2}$	

[43] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

Time shifting

Time shifting

For a function $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$, we have

$$\begin{aligned}\mathcal{L}[f(t-\tau)u(t-\tau)] &= \int_0^{\infty} f(t-\tau)u(t-\tau)e^{-st}dt \\ &= \int_{\tau}^{\infty} f(t-\tau)e^{-st}dt \\ &= e^{-s\tau}\int_0^{\infty} f(\bar{t})e^{-s\bar{t}}d\bar{t} \\ &= e^{-s\tau}\mathcal{L}[f(t)u(t)] = e^{-s\tau}F(s)\end{aligned}$$

with the **same region of convergence** as $\mathcal{L}[f(t)u(t)]$.

We have used the **substitution** $\bar{t} = t - \tau$.

One-sided Laplace transforms[44]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\, e^{\lambda t} u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau} F(s)$
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{(s-a)}$	$\mathcal{L}[u(t-\tau)] = \frac{e^{-s\tau}}{s}$
$\mathcal{L}[t\, u(t)] = \frac{1}{s^2}$	$\mathcal{L}[t\, e^{at}\, u(t)] = \frac{1}{(s-a)^2}$	$\mathcal{L}[(t-\tau)\, u(t-\tau)] = \frac{e^{-s\tau}}{s^2}$
$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{s-a}$		$\mathcal{L}[e^{a(t-\tau)}\, u(t-\tau)] = \frac{e^{-s\tau}}{s-a}$
$\mathcal{L}[e^{j\omega_0 t}\, u(t)] = \frac{1}{s-j\omega_0}$		
$\mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2}$	$\mathcal{L}[\sin(\omega_0(t-\tau))\, u(t-\tau)] = \frac{\omega_0\, e^{-s\tau}}{s^2+\omega_0^2}$
$\mathcal{L}[\cos(\omega_0 t)\, u(t)] = \frac{s}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \cos(\omega_0 t)\, u(t)] = \frac{s-\sigma_0}{(s-\sigma_0)^2+\omega_0^2}$	$\mathcal{L}[\cos(\omega_0(t-\tau))\, u(t-\tau)] = \frac{s\, e^{-s\tau}}{s^2+\omega_0^2}$

[44] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

Differentiation

Derivative of $f(t)$

For a signal $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$,

$$\mathcal{L}\left[\frac{df(t)}{dt}u(t)\right] = s\,F(s) - f(0^-).$$

In most cases, 0^- can be **replaced** by 0.

Differentiation: proof

We have

$$\mathcal{L}\left[\frac{df(t)}{dt}u(t)\right] = \int_{0^-}^{\infty} \frac{df(t)}{dt} e^{-st} dt.$$

Using **integration by parts** with

$$v = e^{-st} \quad \text{and} \quad \frac{dw}{dt} = \frac{df(t)}{dt},$$

and subsequently **differentiating** and **integrating**

$$\frac{dv}{dt} = -s\, e^{-st} \quad \text{and} \quad w = f(t),$$

we **obtain**

$$\begin{aligned} \int_{0^-}^{\infty} v \, \frac{dw}{dt} \, dt &= vw|_{0^-}^{\infty} - \int_{0^-}^{\infty} w \, \frac{dv}{dt} \, dt \\ &= -f(0^-) + s \int_{0^-}^{\infty} f(t) e^{-st} dt = s\, F(s) - f(0^-). \end{aligned}$$

Differentiation

Derivatives of $f(t)$

For a signal $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$,

$$\mathcal{L}[f'(t)u(t)] = sF(s) - f(0).$$

This property can be used to find the **Laplace transform** of the **successive derivatives** of $f(t)$, i.e.

$$\begin{aligned}
\mathcal{L}[f''(t)u(t)] &= s\,\mathcal{L}[f'(t)u(t)] - f'(0) \\
&= s\,(s\,F(s) - f(0)) - f'(0) \\
&= s^2F(s) - s\,f(0) - f'(0)
\end{aligned}$$

Differentiation

F.Y.I.

Derivatives of $F(s)$

For a function $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$,

$$\frac{dF(s)}{ds} = \int_0^\infty f(t)\,\frac{d(e^{-st})}{ds}\,dt = \int_0^\infty (-t\,f(t))\,e^{-st}\,dt = -\mathcal{L}[t\,f(t)u(t)],$$

i.e.

$$\frac{dF(s)}{ds} = -\mathcal{L}[t\,f(t)u(t)].$$

In **general**,

$$\boxed{\frac{d^nF(s)}{ds^n} = (-1)^n\,\mathcal{L}[t^nf(t)u(t)]}$$

Differentiation

F.Y.I.

This **property** can be used to compute **new transforms**, i.e.

- $\mathcal{L}[t^n\, u(t)] = (-1)^n \dfrac{d^n}{ds^n}\left(\dfrac{1}{s}\right) = \dfrac{n!}{s^{n+1}}$
- $\mathcal{L}[t^n\, e^{at}\, u(t)] = (-1)^n \dfrac{d^n}{ds^n}\left(\dfrac{1}{s-a}\right) = \dfrac{n!}{(s-a)^{n+1}}$
- $\mathcal{L}[t \sin(\omega_0 t) u(t)] = -\dfrac{d}{ds}\left(\dfrac{\omega_0}{(s^2+\omega_0^2)}\right) = \dfrac{2\,\omega_0 s}{(s^2+\omega_0^2)^2}$
- $\mathcal{L}[t \cos(\omega_0 t) u(t)] = -\dfrac{d}{ds}\left(\dfrac{s}{(s^2+\omega_0^2)}\right) = \dfrac{s^2-\omega_0^2}{(s^2+\omega_0^2)^2}$

Integration

F.Y.I.

Integration of $f(t)$

The Laplace transform of the **integral** of a causal signal $f(t)$ is given by

$$\mathcal{L}\left[\left(\int_{0^-}^{t} f(\overline{t})\, d\overline{t}\right) u(t)\right] = \frac{F(s)}{s}$$

where $\mathcal{L}[f(t)u(t)] = F(s)$.

Integration of $F(s)$

For a function $f(t)$ with Laplace transform $\mathcal{L}[f(t)u(t)] = F(s)$, we have

$$\int_{s}^{\infty} F(u)\, du = \mathcal{L}\left[\frac{f(t)}{t}\right].$$

Time scaling

F.Y.I.

Time scaling

If the **time scale**[45] of a signal $f(t)$ is **contracted** (expanded), the **frequency scale** of the Laplace transform $F(s)$ is **expanded** (contracted).

For a function $f(t)$ with **Laplace transform** $\mathcal{L}[f(t)u(t)] = F(s)$, we have

$$\mathcal{L}[f(at)u(t)] = \int_0^{\infty} f(at)\, e^{-st}\, dt = \frac{1}{a} \int_0^{\infty} f(\overline{t})\, e^{-\frac{s}{a}\overline{t}}\, d\overline{t} = \frac{1}{a} F\left(\frac{s}{a}\right)$$

Here we have used the **substitution** $\overline{t} = at$.

[45] a is real and $a \neq 0$!

Convolution integral and Laplace transform

Convolution integral

The **Laplace transform** of the **convolution integral** of a **causal** signal $x(t)$, with Laplace transform $X(s)$, and a **causal** signal $h(t)$, with Laplace transform $H(s)$, is given by

$$\mathcal{L}[y(t)] = \mathcal{L}[h(t) * x(t)] = H(s)X(s).$$

Corollary

The **response** $y(t)$ of a LTI system with **causal impulse response** $h(t)$ to a **causal input** $x(t)$ is given by

$$y(t) = \mathcal{L}^{-1}[Y(s)] = \mathcal{L}^{-1}[H(s)X(s)] = h(t) * x(t).$$

Once $Y(s)$ is found, $y(t)$ is computed by means of the **inverse Laplace transform**. This typically involves a **partial fraction expansion**, i.e. decomposition into a sum of rational components of which the inverse transform can be found directly.

Convolution integral and Laplace transform: proof

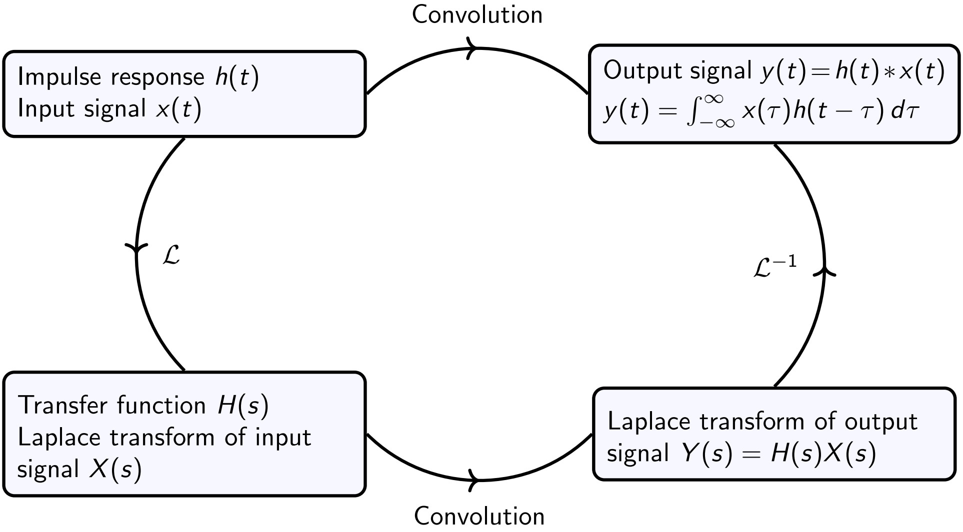
F.Y.I.

$$
\begin{aligned}
Y(s) = \mathcal{L}[y(t)] &= \int_0^\infty y(t)\, e^{-st}\, dt \\
&= \int_0^\infty \left(\int_0^\infty h(\tau) x(t-\tau)\, d\tau \right) e^{-st}\, dt \\
&= \int_0^\infty \int_0^\infty h(\tau) x(t-\tau)\, e^{-st}\, dt\, d\tau
\end{aligned}
$$

Change variable $\sigma = t - \tau$, $d\sigma = dt$, we have

$$
\begin{aligned}
Y(s) &= \int_0^\infty \int_0^\infty h(\tau) x(\sigma)\, e^{-s(\sigma+\tau)}\, d\sigma\, d\tau \\
&= \int_0^\infty h(\tau)\, e^{-s\tau} d\tau \int_0^\infty x(\sigma)\, e^{-s\sigma}\, d\sigma \\
&= H(s)\, X(s)
\end{aligned}
$$

Convolution integral and Laplace transform



Transfer function

Transfer function

The **transfer function** $H(s) = \mathcal{L}[h(t)]$, the **Laplace transform** of the **impulse response** $h(t)$ of a LTI system can be expressed as the **ratio**

$$H(s) = \frac{\mathcal{L}[y(t)]}{\mathcal{L}[x(t)]} = \frac{\mathcal{L}[\text{output signal}]}{\mathcal{L}[\text{input signal}]} \; (= P(s)\,)$$ [46]

This function is called **transfer function** as it **transfers** the Laplace transform of the input to the output.

The transfer function **characterizes the system** by means of its **poles** and **zeros**.

Thus it becomes a very **important tool** for the **analysis** and **synthesis** of (control) systems.

[46] We will sometimes use $P(s)$ to represent the process by its transfer function. The downside is that it is not clear anymore from the notations that the transfer function $P(s)$ is the Laplace transform of the impulse response of the system.

Transfer function and transform of a signal

Note that $^1/_s$ represents

- the **Laplace transform of a step input** signal $x(t) = u(t)$, i.e.
$$X(s) = \mathcal{L}[x(t)] = \mathcal{L}[u(t)] = \frac{1}{s},$$
- the **transfer function of an integrating process**[47], i.e. the Laplace transform of the **impulse response** $h(t) = u(t)$. We have that
$$H(s) = \mathcal{L}[h(t)] = \mathcal{L}[u(t)] = \frac{1}{s}.$$

[47] e.g. a surge tank or "cuvier" in French

Transfer function and transform of a signal

Note that ${}^{\omega_0}\!/\!{}_{s^2+\omega_0^2}$ represents

- the **Laplace transform of a sinusoidal input** signal $x(t) = \sin(\omega_0 t)\, u(t)$, i.e.

$$X(s) = \mathcal{L}[x(t)] = \mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2},$$

- the **transfer function of an oscillating process**[48] **without damping**, i.e. the Laplace transform of the **impulse response** $h(t) = \sin(\omega_0 t)\, u(t)$. We have that

$$H(s) = \mathcal{L}[h(t)] = \mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2}.$$

[48] e.g. a pendulum

Transfer function and transform of a signal

Note that $\omega_0/(s-\sigma_0)^2+\omega_0^2$ represents

- the **Laplace transform of a damped sinusoidal input** signal $x(t) = e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)$, i.e.

$$X(s) = \mathcal{L}[x(t)] = \mathcal{L}[e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2},$$

- the **transfer function of an oscillating process**[49] **with damping**, i.e. the Laplace transform of the **impulse response** $h(t) = e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)$. We have that

$$H(s) = \mathcal{L}[h(t)] = \mathcal{L}[e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2}.$$

[49] e.g. a pendulum with friction

One-sided Laplace transforms[50]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\,e^{\lambda t}u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau}F(s)$
$\mathcal{L}[\delta(t)] = 1$ $\mathcal{L}[u(t)] = \frac{1}{s}$ $\mathcal{L}[t\,u(t)] = \frac{1}{s^2}$		

[50] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

One-sided Laplace transforms[50]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\, e^{\lambda t} u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau} F(s)$
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{(s-a)}$	
$\mathcal{L}[t\, u(t)] = \frac{1}{s^2}$		
$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{s-a}$		
$\mathcal{L}[e^{j\omega_0 t}\, u(t)] = \frac{1}{s-j\omega_0}$		

[50] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

One-sided Laplace transforms[50]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\, e^{\lambda t} u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau} F(s)$
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{(s-a)}$	
$\mathcal{L}[t\, u(t)] = \frac{1}{s^2}$		
$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{s-a}$		
$\mathcal{L}[e^{j\omega_0 t}\, u(t)] = \frac{1}{s-j\omega_0}$		
$\mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$		
$\mathcal{L}[\cos(\omega_0 t)\, u(t)] = \frac{s}{s^2+\omega_0^2}$		

[50] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

One-sided Laplace transforms[50]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\, e^{\lambda t} u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau} F(s)$
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{(s-a)}$	
$\mathcal{L}[t\, u(t)] = \frac{1}{s^2}$	$\mathcal{L}[t\, e^{at}\, u(t)] = \frac{1}{(s-a)^2}$	
$\mathcal{L}[e^{at}\, u(t)] = \frac{1}{s-a}$		
$\mathcal{L}[e^{j\omega_0 t}\, u(t)] = \frac{1}{s-j\omega_0}$		
$\mathcal{L}[\sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \sin(\omega_0 t)\, u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2}$	
$\mathcal{L}[\cos(\omega_0 t)\, u(t)] = \frac{s}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t} \cos(\omega_0 t)\, u(t)] = \frac{s-\sigma_0}{(s-\sigma_0)^2+\omega_0^2}$	

[50] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

One-sided Laplace transforms[50]		
Original transform $\mathcal{L}[f(t)u(t)] = F(s)$	**Frequency shifting** $\mathcal{L}[f(t)\,e^{\lambda t}u(t)] = F(s-\lambda)$	**Time shifting** $\mathcal{L}[f(t-\tau)u(t-\tau)] = e^{-s\tau}F(s)$
$\mathcal{L}[\delta(t)] = 1$		
$\mathcal{L}[u(t)] = \frac{1}{s}$	$\mathcal{L}[e^{at}\,u(t)] = \frac{1}{(s-a)}$	$\mathcal{L}[u(t-\tau)] = \frac{e^{-s\tau}}{s}$
$\mathcal{L}[t\,u(t)] = \frac{1}{s^2}$	$\mathcal{L}[t\,e^{at}\,u(t)] = \frac{1}{(s-a)^2}$	$\mathcal{L}[(t-\tau)\,u(t-\tau)] = \frac{e^{-s\tau}}{s^2}$
$\mathcal{L}[e^{at}\,u(t)] = \frac{1}{s-a}$		$\mathcal{L}[e^{a(t-\tau)}\,u(t-\tau)] = \frac{e^{-s\tau}}{s-a}$
$\mathcal{L}[e^{j\omega_0 t}\,u(t)] = \frac{1}{s-j\omega_0}$		
$\mathcal{L}[\sin(\omega_0 t)\,u(t)] = \frac{\omega_0}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t}\sin(\omega_0 t)\,u(t)] = \frac{\omega_0}{(s-\sigma_0)^2+\omega_0^2}$	$\mathcal{L}[\sin(\omega_0(t-\tau))\,u(t-\tau)] = \frac{\omega_0\,e^{-s\tau}}{s^2+\omega_0^2}$
$\mathcal{L}[\cos(\omega_0 t)\,u(t)] = \frac{s}{s^2+\omega_0^2}$	$\mathcal{L}[e^{\sigma_0 t}\cos(\omega_0 t)\,u(t)] = \frac{s-\sigma_0}{(s-\sigma_0)^2+\omega_0^2}$	$\mathcal{L}[\cos(\omega_0(t-\tau))\,u(t-\tau)] = \frac{s\,e^{-s\tau}}{s^2+\omega_0^2}$

[50] You must be able to obtain all transforms and prove all properties from the definition of the Laplace transform.

Stability in the Laplace domain

Reminder:

- The **ROC** of the **Laplace transform** of the **impulse response** of a **causal** system, i.e. its transfer function, is

$$\boxed{\text{ROC } = \left\{ s = \sigma + j\omega \text{ such that } \int_0^{\infty} |h(t)e^{-\sigma t}| dt < \infty \right\}}$$

- A **causal** LTI system is **BIBO stable** if and only if

$$\boxed{\int_0^{\infty} |h(t)| dt < \infty}$$

Note that

- If the **ROC** includes the **imaginary axis**, i.e. $\sigma = 0$ in ROC, then the system is **BIBO stable**.
- If the system is **BIBO stable** then the **ROC** includes the **imaginary axis**, i.e. $\sigma = 0$ in ROC.

Stability in the Laplace domain

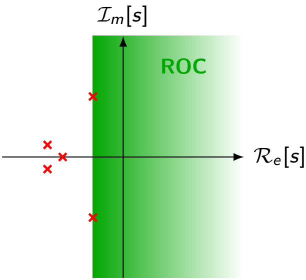
Stability in the Laplace domain

The **causal** system is **stable** if it can be **described** by a **transfer function** $H(s)$ with a **region of convergence** that **includes the imaginary axis**, i.e. $\sigma = 0$ in ROC.

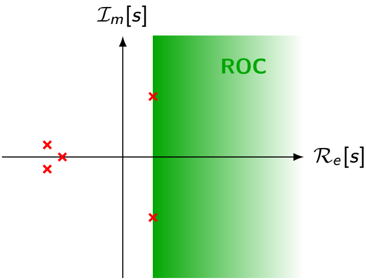
For a causal system, the condition implies that all the poles of $H(s)$ are strictly in the open left half s-plane.

Stability in the *s*-domain: illustration for a causal system

Assuming a **causal** system, the **ROC** of the transfer function $H(s)$ is $\sigma > \sigma_{max}$ where σ_{max} is the **largest real part** of the poles of $H(s)$. The ROC extends to the **right** of the pole with the **largest real part**.



Stable system



Unstable system

Stability of a system

- A system is **stable** if its impulse response has a **zero steady-state response**. The poles of the transfer function, $H(s)$, are **all** in the **open left half s-plane**.
- A system is **marginally stable** if the impulse response as a **constant** or a **sinusoidal** steady-state response. The transfer function, $H(s)$, has poles in the **open left half s-plane** and a **simple real pole** at the **origin** or **simple complex conjugate poles** on the $j\omega$-axis.
- A system is **unstable** if its impulse response **grows** as $t \to \infty$. The transfer function, $H(s)$, has **multiple poles** on the $j\omega$-axis and/or **poles in the right half s-plane**.

Stability in the Laplace domain

Control engineering joke

A group of Polish tourists is flying on a small airplane through the Grand Canyon on a sightseeing tour. The tour guide anounces: "On the right of the airplane, you can see the famous Bright Angle Falls". The tourists leap out of their seats and crowd to the windows on the right side. This causes a dynamic imbalance, and the plane violently rolls to the side and crashes into the canyon wall. All aboard are lost.

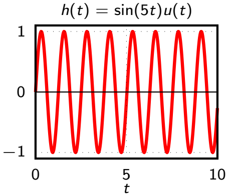
The moral to this episode is: always keep your poles to the left side of the plane.

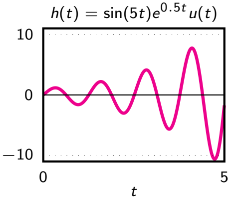
Dynamic behaviour

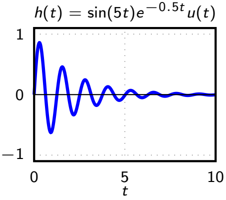
The **two properties** that have been used in the next slide are

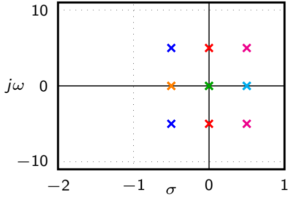
- $\mathcal{L}[e^{at}u(t)] = \dfrac{1}{(s-a)}$, a real
- $\mathcal{L}[e^{\sigma_0 t}\sin(\omega_0 t)\, u(t)] = \dfrac{\omega_0}{(s-\sigma_0)^2+\omega_0^2} = \dfrac{\omega_0}{(s-\sigma_0-j\omega_0)(s-\sigma_0+j\omega_0)}$

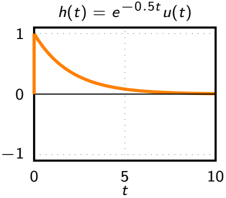
Dynamic behaviour

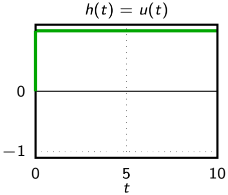


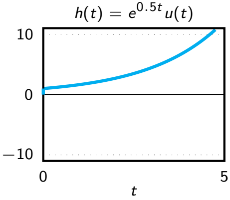












Initial value theorem

From the **Laplace transform** $Y(s)$, it is possible to establish the **initial value** of $y(t)$ in the **vicinity** of $t = 0^+$.

Initial value theorem

Supposing that

- $y(t)$ and its derivatives have Laplace transforms,
- the limit $y(0^+) = \lim_{t \to 0^+} y(t)$ exists,

we have

$$\boxed{y(0^+) = \lim_{s \to \infty} sY(s)}$$

Final value theorem

From the **Laplace transform** $Y(s)$, it is possible to establish the **steady-state value**[51] of $y(t)$ when $t \to \infty$.

Final value theorem

Assuming $\mathcal{R}_e[\text{poles}(sY(s))] < 0$, i.e. all poles of $sY(s)$ are in the **left half plane**, we have

$$\boxed{\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s)}$$

[51] La valeur finale atteinte en **régime permanent** par $y(t)$.

Initial and final value theorems: application

Consider

- a **lead-lag system** $H(s) = K\,\dfrac{T_1\,s + 1}{T_2\,s + 1}$ and
- and **unit step input** $x(t)$, i.e. $X(s) = \dfrac{1}{s}$.

The Laplace transform of the **response** is

$$Y(s) = H(s)X(s) = K\,\frac{T_1\,s + 1}{s(T_2\,s + 1)}.$$

We can use the **initial** and **final** value theorems to get an **idea of the response in time domain**, i.e.

- **Initial value theorem**: the response in the **vicinity** of $t = 0^+$ is
$$y(0^+) = \lim_{s\to\infty} sY(s) = K\,\frac{T_1}{T_2}.$$
- **Final value theorem**: **steady-state response** is
$$y(\infty) = \lim_{s\to 0} sY(s) = K.$$

Static gain

The **static gain** of a system described by the **transfer function** $H(s)$ is obtained by evaluating the transfer function at $s = 0$, i.e.

$$\boxed{\textbf{Static gain} \text{ of the system } H(s) \text{ is } H(s)|_{s=0} = H(0)}$$

The **static gain** tells the **ratio** of the **output** and the **input** under **steady state condition**.

If the input $x(t) = x_0$ is **constant** and the system is **stable** then the output will reach the **steady state value**

$$y_0 = H(s)|_{s=0}\, x_0$$

when **all transients have disappeared**.

The transfer function can thus be viewed as a **generalization** of the concept of gain.

General ideas

Assume the signal we wish to find has a **rational Laplace transform**, i.e.

$$F(s) = \frac{n(s)}{d(s)}$$

where $n(s)$ and $d(s)$ are **polynomials** in s with **real-valued** coefficients.

In order for the **partial fraction expansion** to be **possible**, it is required that $F(s)$ be **strictly proper** rational, which means that the degree of the numerator polynomial $n(s)$ is less than that of the denominator polynomial $d(s)$, i.e. the **relative degree** (number of poles minus number of zeros) is at least 1.

If $F(s)$ is **not strictly proper**, then we need to do **Eucledian division** until we obtain a **strictly proper rational function**, i.e.

$$F(s) = g_0 + g_1 s + \cdots + g_m s^m + \frac{b(s)}{d(s)}$$

where the degree of $b(s)$ is now **less** than that of $d(s)$ so that we **can** perform partial expansion.

General ideas

- The poles of $F(s)$ provide the basic characteristics of the $f(t)$. If $F(s) = H(s)$, the poles provide the **essential characteristics** of the system described by the transfer $H(s)$.
- If $n(s)$ and $d(s)$ are polynomials in s with real coefficients, then the zeros and poles of $F(s)$ are **real** and/or **complex conjugate pairs**, and can be **simple** or **multiple**.
- The basic idea of the partial expansion is to decompose strictly proper rational functions into a **sum of rational components** of which the inverse transform can be found directly in tables.
- In the inverse, $u(t)$ should be included since the result of the inverse is **causal**, i.e. the function $u(t)$ is an **integral part of the inverse**.

Roots and factorisation of a polynomial

- The **roots**[52] $\{p_k\}$ of a **polynomial** $d(s)$ in s are the **values** for which
$$d(p_k) = 0.$$
- A **polynomial** in s of degree n with **roots** $\{p_k\}$ $(k = 1, \cdots, n)$ can be **factored** as follows
$$d(s) = d_n\, s^n + d_{n-1}\, s^{n-1} + \cdots + d_0 = d_n \prod_{k=1}^{n}(s - p_k).$$

[52] racines

Partial fraction expansion: simple real poles

Simple real poles

If $F(s)$ is a strictly **strictly proper rational function**

$$F(s) = \frac{n(s)}{\Pi_k (s - p_k)}$$

where the $\{p_k\}$ are **simple real poles** of $F(s)$. The **partial fraction expansion** and its **inverse** Laplace transform yields

$$F(s) = \sum_k \frac{A_k}{s - p_k} \Longleftrightarrow f(t) = \sum_k A_k \, e^{p_k t} u(t).$$

where the **expansion coefficients** A_k are computed as

$$A_k = F(s)(s - p_k)|_{s=p_k}.$$

Partial fraction expansion: simple complex conjugate poles

Simple complex conjugate poles

The **partial fraction expansion** of a **strictly proper rational** function[53]

$$F(s) = \frac{n(s)}{(s-\sigma_0)^2 + \omega_0^2} = \frac{n(s)}{(s-\sigma_0-j\omega_0)(s-\sigma_0+j\omega_0)}$$

with **complex conjugate poles** $\{p_{1,2} = \sigma_0 \pm j\omega_0\}$ is given by

$$F(s) = \frac{A}{(s-\sigma_0-j\omega_0)} + \frac{A^\star}{(s-\sigma_0+j\omega_0)}$$

$$\text{where } A = F(s)(s-\sigma_0-j\omega_0)|_{s=\sigma_0+j\omega_0} = K\, e^{j\phi}$$

so that the **inverse** Laplace transform yields

$$f(t) = 2K\, e^{\sigma_0 t}\cos(\omega_0 t + \phi)\, u(t).$$

[53] The denominator is obtained by **completing the squares** !

Partial fraction expansion: simple complex conjugate poles

Simple complex conjugate poles

An **equivalent partial fraction** expansion[54] consists in **expressing** the numerator $n(s)$ of the strictly proper rational function $F(s)$ as $n(s) = a + b\,(s - \sigma_0)$, for some constants a and b, i.e.

$$F(s) = \frac{a + b\,(s - \sigma_0)}{(s - \sigma_0)^2 + \omega_0^2} = \frac{a}{\omega_0}\,\frac{\omega_0}{(s - \sigma_0)^2 + \omega_0^2} + b\,\frac{(s - \sigma_0)}{(s - \sigma_0)^2 + \omega_0^2}.$$

The **inverse** Laplace transform is a **sum** of a **sine** and a **cosine** multiplied by a **decaying exponential**, i.e.

$$f(t) = e^{\sigma_0 t}\left[\frac{a}{\omega_0}\sin\omega_0 t + b\cos\omega_0 t\right] u(t).$$

Note that this form is equivalent to the one in the previous slide.

[54] This is often <u>**easier**</u> than to work with complex numbers !

Partial fraction expansion: example

Find the **inverse Laplace transform** of the function

$$F(s) = \frac{4}{s^3 + 2s^2 + 4s}.$$

Completing the squares, we obtain

$$F(s) = \frac{4}{s(s^2 + 2s + 4)} = \frac{4}{s((s+1)^2 + 3)}.$$

We will perform a **partial fraction expansion**

$$F(s) = \frac{A}{s} + \frac{B\,s + C}{(s+1)^2 + 3}.$$

We have that

$$A = F(s)s|_{s=0} = 1.$$

Bringing back to same denominator, we obtain the following **constraint**

$$n(s) = 4 = A(s^2 + 2s + 4) + B\,s^2 + C\,s$$

yielding $B = -1$ and $C = -2$.

Partial fraction expansion: example

We have thus obtained the **partial fraction expansion**

$$F(s) = \frac{1}{s} - \frac{s+2}{(s+1)^2+3}.$$

Equivalently, we obtain

$$F(s) = \frac{1}{s} - \frac{s+1}{(s+1)^2+3} - \frac{1}{\sqrt{3}} \frac{\sqrt{3}}{(s+1)^2+3}.$$

The **inverse Laplace transform** is therefore

$$f(t) = \left(1 - e^{-t}\left(\cos(\sqrt{3}t) + \frac{1}{\sqrt{3}}\sin(\sqrt{3}t)\right)\right) u(t).$$

Partial fraction expansion: multiple real poles

Multiple real poles

If $F(s)$ is a strictly proper rational function with **multiple real poles**, the **partial fraction expansion** is

$$F(s) = \frac{n(s)}{(s-a)^r} = \frac{c_1}{(s-a)} + \frac{c_2}{(s-a)^2} + \cdots + \frac{c_r}{(s-a)^r}$$

and the **inverse** Laplace transform is

$$f(t) = \left[c_1 + c_2 t + \cdots + c_r \frac{t^{r-1}}{(r-1)!} \right] e^{at} \, u(t)$$

with

$$\begin{aligned} c_{r-k} &= \frac{1}{k!} \frac{d^k}{ds^k} \left[F(s)(s-a)^r \right] \Big|_{s=a}, \quad k = 1, \cdots, r-1 \\ c_r &= F(s)(s-a)^r \big|_{s=a} \end{aligned}$$

Partial fraction expansion: example

Find the **inverse Laplace transform** of the function

$$F(s) = \frac{4}{s\,(s+2)^2}.$$

We will perform a **partial fraction expansion**

$$F(s) = \frac{A}{s} + \frac{B}{s+2} + \frac{C}{(s+2)^2}.$$

We have that

$$A = F(s)\,s|_{s=0} = \left.\frac{4}{(s+2)^2}\right|_{s=0} = 1.$$

$$C = F(s)(s+2)^2|_{s=-2} = \left.\frac{4}{s}\right|_{s=-2} = -2.$$

$$B = \left[F(s)(s+2)^2\right]'|_{s=-2} = \left.-\frac{4}{s^2}\right|_{s=-2} = -1.$$

Partial fraction expansion: example

We have thus obtained the **partial fraction expansion**

$$F(s) = \frac{1}{s} + \frac{-1}{s+2} + \frac{-2}{(s+2)^2}.$$

The **inverse Laplace transform** is therefore

$$f(t) = \left(1 - e^{-2t} - 2t\, e^{-2t}\right) u(t).$$

Analysis of LTI systems

The **complete response** $y(t)$ of a system represented by an nth-order **linear differential equation** with **constant** coefficients,

$$a_0\, y(t) + a_1 \frac{dy(t)}{dt} + \ldots + a_n \frac{d^n y(t)}{dt^n} = b_0\, x(t) + b_1 \frac{dx(t)}{dt} + \ldots + b_m \frac{d^m x(t)}{dt^m}$$

where $x(t)$ is the **input** and $y(t)$ is the **output** of the system, and **initial conditions**

$$\{y^{(k)}(0),\ 0 \leq k \leq n-1\},$$

is obtained by **inverting the Laplace transform**

$$Y(s) = \frac{b(s)}{a(s)}\, X(s) + \frac{1}{a(s)}\, I(s)$$

where $Y(s) = \mathcal{L}[y(t)]$ and $X(s) = \mathcal{L}[x(t)]$.

Here we have used the notation $y^{(k)}(t) = \frac{d^k y(t)}{dt^k}$.

Analysis of LTI systems

The **complete response** $y(t)$ of a system represented by an nth-order **linear differential equation** with **constant** coefficients,

$$\sum_{k=0}^{n} a_k\, y^{(k)}(t) = \sum_{l=0}^{m} b_l\, x^{(l)}(t)$$

where $x(t)$ is the **input** and $y(t)$ is the **output** of the system, and **initial conditions**

$$\{y^{(k)}(0),\ 0 \leq k \leq n-1\},$$

is obtained by **inverting the Laplace transform**

$$Y(s) = \frac{b(s)}{a(s)}\, X(s) + \frac{1}{a(s)}\, I(s)$$

where $Y(s) = \mathcal{L}[y(t)]$ and $X(s) = \mathcal{L}[x(t)]$.

Here we have used the notation $y^{(k)}(t) = \frac{d^k y(t)}{dt^k}$.

Analysis of LTI systems

The complete response $y(t)$ is obtained by **inverting the Laplace transform**

$$Y(s) = \frac{b(s)}{a(s)} X(s) + \frac{1}{a(s)} I(s)$$

which gives

$$y(t) = y_{zs}(t) + y_{zi}(t)$$

where

- $y_{zs}(t) = \mathcal{L}^{-1}\left[H(s)X(s)\right]$ is the **zero-state response**,
- $y_{zi}(t) = \mathcal{L}^{-1}\left[H_i(s)I(s)\right]$ is the **zero-input response**, with

$$H(s) = \frac{b(s)}{a(s)} \text{ et } H_i(s) = \frac{1}{a(s)}.$$

The associated **transfer function** is $H(s)$.

Transient and steady-state responses

F.Y.I.

Consider a **stable** system with the following response in the Laplace domain

$$Y(s) = H(s)\,X(s) = \frac{b(s)}{a(s)}\,X(s) \;\; \text{(zero initial conditions).}$$

Expanding $H(s)$ and $X(s)$ into **partial fractions**[55]

$$H(s) = \frac{b(s)}{a(s)} = \frac{b(s)}{a_n\,(s-p_1)\cdots(s-p_n)}$$

$$X(s) = \frac{n(s)}{d(s)} = \frac{n(s)}{d_n\,(s-q_1)\cdots(s-q_l)}$$

yields

$$Y(s) = \sum_{i=1}^{n} \frac{c_i}{s-p_i} + \sum_{j=1}^{l} \frac{k_j}{s-q_j}.$$

[55] For convenience, we suppose that all poles are **simple** and **real**. The result **generalises** to multiple and/or complex poles.

Transient and steady-state responses

F.Y.I.

The inverse Laplace transform of a **stable** system yields

$$y(t) = \underbrace{\sum_{i=1}^{n} c_i \, e^{p_i t} \, u(t)}_{\text{Transient response}} + \underbrace{\sum_{j=1}^{l} k_j \, e^{q_j t} \, u(t)}_{\text{Steady-state response}}$$

- The **natural** response only depends on the natural modes of the system. As the system is **stable**, the **natural response decays** and it disappears when $t \to \infty$. It is then called a **transient** response.
- Assuming that the system is **stable**, then the system response in the long run is determined by its **steady state response only**.

  The **steady-state** response then depends on the input signal **only**.

Steady-state response: **general case**

F.Y.I.

- If the poles (simple or multiple, real or complex) of the Laplace transform of the output, $Y(s)$, of an LTI system are in the **open left half s-plane** (i.e. no poles on the $j\omega$-axis), the **steady-state response is zero**.
- **Simple complex conjugate poles** or a **simple real pole** at the **origin** of the s-plane cause a **steady-state response**:
  - If the pole of $Y(s)$ is $s = 0$, we know that its inverse transform is of the **form** $A\,u(t)$.
  - If the poles of $Y(s)$ are the **complex conjugates** $s = \pm j\omega_0$, the corresponding inverse transform is a **sinusoidal signal**.
- However, **multiple poles** on the $j\omega$-axis, or any **poles in the right half s-plane** will give inverses that **grow** as $t \to \infty$.

Frequency response

F.Y.I.

There is a **tight link** between the **transfer function** and the **frequency response** of a system, defined as the **steady-state response** of the system to a **sinusoidal input signal**.

Consider $X(s) = \frac{\omega}{s^2+\omega^2}$. The **response** of the system is

$$Y(s) = H(s)\,\frac{\omega}{s^2+\omega^2} = \frac{b(s)}{a(s)}\,\frac{\omega}{s^2+\omega^2} = \frac{b(s)}{a(s)}\,\frac{\omega}{(s-j\omega)(s+j\omega)}$$

Expanding into **partial fractions** and assuming a **stable** system, we obtain

$$Y(s) = \underbrace{\frac{c_1}{(s-j\omega)} + \frac{c_2}{(s+j\omega)}}_{\text{steady-state response}} + \underbrace{\frac{\overline{b}(s)}{a(s)}}_{\text{transient response}}$$

$$c_1 = \frac{b(s)}{a(s)}\,\frac{\omega}{(s+j\omega)}\bigg|_{s=j\omega} = \frac{H(j\omega)}{2j},\;\; c_2 = \frac{b(s)}{a(s)}\,\frac{\omega}{(s-j\omega)}\bigg|_{s=-j\omega} = -\frac{H(-j\omega)}{2j}$$

Frequency response

F.Y.I.

The **steady-state response** to a **sinusoidal input** is

$$
\begin{aligned}
y_{ss}(t) &= \mathcal{L}^{-1}\left[\frac{H(j\omega)}{2j(s-j\omega)} - \frac{H(-j\omega)}{2j(s+j\omega)}\right] \\
&= \left(\frac{H(j\omega)}{2j}\,e^{j\omega t} - \frac{H(-j\omega)}{2j}\,e^{-j\omega t}\right) u(t)
\end{aligned}
$$

Writing $H(j\omega)$ in **polar form** $H(j\omega) = |H(\omega)|e^{j\,\arg[H(j\omega)]}$ yields

$$
\begin{aligned}
y_{ss}(t) &= \frac{|H(j\omega)|}{2j}\left(e^{j(\omega t+\arg[H(j\omega)])} - e^{-j(\omega t+\arg[H(j\omega)])}\right) u(t), \\
&= |H(j\omega)|\sin(\omega t + \arg[H(j\omega)])\,u(t).
\end{aligned}
$$

Remember that $H(j\omega)$ is to be **interpreted** as $H(s)|_{s=j\omega}$.

Frequency response

The **steady-state response** to a **sinusoidal input** $x(t) = \sin(\omega t)\, u(t)$ is

$$\boxed{y_{ss}(t) = |H(j\omega)| \sin(\omega t + \arg[H(j\omega)])\, u(t)}$$

where

- $H(j\omega)$ is to be **interpreted** as $H(s)|_{s=j\omega}$,
- $|H(j\omega)|$ is the **magnitude** or **modulus** as a function of ω,
- $\arg[H(j\omega)]$ is the **phase shift** as a function of ω.

The linear system acts as a **frequency filter**. Varying the frequency ω from 0 to infinity, we obtain **an infinite number of sinusoidal input steady-state responses** that can be represented **graphically**.

Frequency response

The **steady-state response** to a **sinusoidal input** $x(t) = \sin(\omega t)\, u(t)$ is

$$\boxed{y_{ss}(t) = |H(j\omega)| \sin(\omega t + \arg[H(j\omega)])\, u(t)}$$

Frequency-domain representations of the system:

- **Bode diagram**: $\{\omega, |H(j\omega)|, \arg[H(j\omega)]\}$, i.e. the **gain** and **phase shift** as a function of **frequency** using logarithmic or semi-logarithmic scales.
- **Nyquist diagram**: $\{\mathcal{R}_e[H(j\omega)], \mathcal{I}_m[H(j\omega)]\}$, i.e. in **Cartesian** coordinates, the **real** part of $H(j\omega)$ **versus** the **imaginary** part of $H(j\omega)$ using frequency as a parameter in the plot.
- **Black-Nichols diagram**: $\{|H(j\omega)|, \arg[H(j\omega)]\}$, i.e. in **polar** coordinates, the **gain versus** the **phase shift** using frequency as a parameter in the plot.

First order system: Laplace domain

In general, a **first order system** is described by a **first order differential equation**

$$\boxed{T\,\frac{dy(t)}{dt} + y(t) = K\,x(t)}$$

with T the **time constant** of the system and K is the **static gain** of the system.

In the Laplace domain, we have

$$T\,(sY(s) - y(0)) + Y(s) = K\,X(s)$$

This yields

$$\boxed{Y(s) = \underbrace{\frac{K}{Ts+1}\,X(s)}_{\text{Zero-state response}} + \underbrace{\frac{T}{Ts+1}\,y(0)}_{\text{Zero-input response}}}$$

First order system: Laplace domain

In general, a **first order system** is described by a **first order differential equation**

$$\boxed{T\,\frac{dy(t)}{dt} + y(t) = K\,x(t)}$$

with T the **time constant** of the system and K is the **static gain** of the system.

In the Laplace domain assuming **zero initial conditions**, we have

$$\boxed{H(s) = \frac{Y(s)}{X(s)} = \frac{K}{Ts+1}}$$

First order system: impulse response

The **impulse response** of a first order system

$$H(s) = \frac{K}{Ts + 1}$$

is

$$Y(s) = H(s)X(s) = H(s)$$

The **time-domain response** is

$$y(t) = h(t) = \frac{K}{T}\, e^{-\frac{t}{T}}\, u(t).$$

First order system: step response

The **step response** of a first order system

$$H(s) = \frac{K}{Ts + 1}$$

is

$$Y(s) = H(s)X(s) = \frac{H(s)}{s} = K\left(\frac{1}{s} - \frac{T}{Ts + 1}\right)$$

The **time-domain response** is

$$y(t) = s(t) = K\left(1 - e^{-\frac{t}{T}}\right) u(t).$$

First order system: Bode diagram[56]

$$Y(s) = \frac{K}{Ts+1} X(s) = H(s)X(s)$$

Note that

$$\begin{aligned} H(j\omega) = \left.\frac{K}{Ts+1}\right|_{s=j\omega} &= \frac{K}{j\omega T+1} = K\,\frac{1-j\omega T}{1+\omega^2T^2} \\ &= K\,\frac{1}{1+\omega^2T^2} - j\,K\,\frac{\omega T}{1+\omega^2T^2} \\ &= \mathcal{R}_e[H(j\omega)] + j\,\mathcal{I}_m[H(j\omega)]. \end{aligned}$$

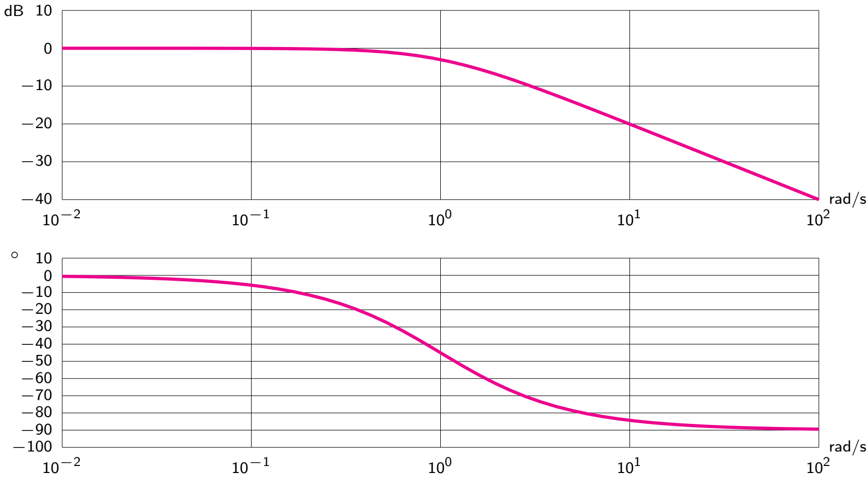
Therefore

$$|H(j\omega)| = \sqrt{\mathcal{R}_e^2[H(j\omega)] + \mathcal{I}_m^2[H(j\omega)]} = \frac{K}{\sqrt{1+\omega^2T^2}},$$

$$\arg[H(j\omega)] = \arctan\left(\frac{\mathcal{I}_m[H(j\omega)]}{\mathcal{R}_e[H(j\omega)]}\right) = -\arctan(\omega T).$$

[56] For a **stable system**, i.e. $T > 0$.

First order system: Bode diagram[57]



[57] $K = 1,\ T = 1$

First order system: Bode diagram

One has

$$H(j\omega) = H(s)|_{s=j\omega} = \frac{K}{j\omega T + 1} = \frac{K}{j\frac{\omega}{\omega_c} + 1}$$

- When $\omega << \frac{1}{T} = \omega_c$. Then

$$|H(j\omega)| \approx K \text{ and } \arg[H(j\omega)] \approx 0.$$

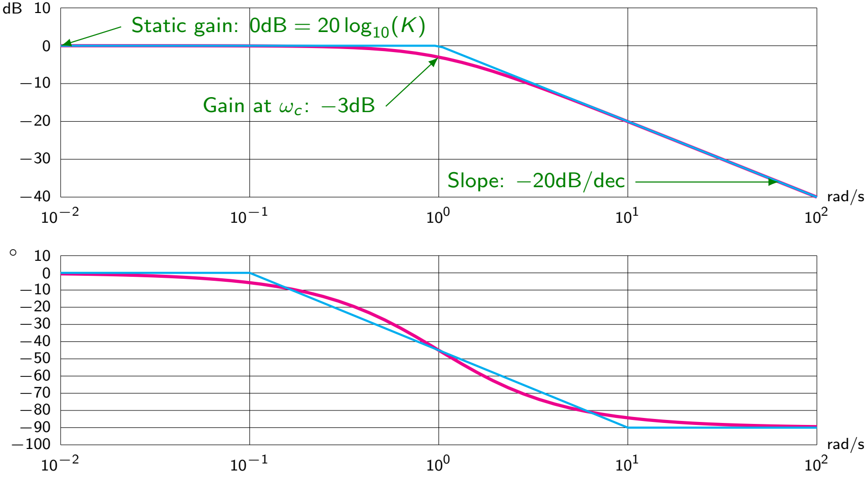
- When $\omega >> \frac{1}{T} = \omega_c$. Then

$$|H(j\omega)| \approx \frac{K}{\omega T} \text{ and } \arg[H(j\omega)] \approx -90^\circ.$$

- When $\omega = \frac{1}{T} = \omega_c$. Then

$$|H(j\omega)| = \frac{K}{\sqrt{2}} \text{ and } \arg[H(j\omega)] = -45^\circ.$$

First order system: Bode diagram[58]



[58] $K = 1,\ T = 1$

Dead-time

Time domain:

$$y(t) = x(t-\theta)$$

Laplace domain:

$$Y(s) = e^{-\theta s} X(s) = H(s)X(s)$$

Note that

$$e^{-\theta s}\Big|_{s=j\omega} = e^{-j\omega\theta} = \cos(\omega\theta) - j\sin(\omega\theta) = \mathcal{R}_e[H(j\omega)] + j\,\mathcal{I}_m[H(j\omega)].$$

Therefore

$$|H(j\omega)| = |e^{-j\omega\theta}| = \sqrt{\mathcal{R}_e^2[H(j\omega)] + \mathcal{I}_m^2[H(j\omega)])} = 1,$$

$$\arg[H(j\omega)] = \arctan\left(\frac{\mathcal{I}_m[H(j\omega)]}{\mathcal{R}_e[H(j\omega)]}\right) = -\omega\theta.$$

Note that in the previous formula, phase is expressed in **radians** !

Dead-time

Time domain:

$$y(t) = x(t-\theta)$$

Laplace domain:

$$Y(s) = e^{-\theta s}X(s) = H(s)X(s)$$

Note that

$$e^{-\theta s}\Big|_{s=j\omega} = e^{-j\omega\theta} = \cos(\omega\theta) - j\sin(\omega\theta) = \mathcal{R}_e[H(j\omega)] + j\,\mathcal{I}_m[H(j\omega)].$$

Therefore

$$|H(j\omega)| = |e^{-j\omega\theta}| = \sqrt{\mathcal{R}_e^2[H(j\omega)] + \mathcal{I}_m^2[H(j\omega)])} = 1,$$

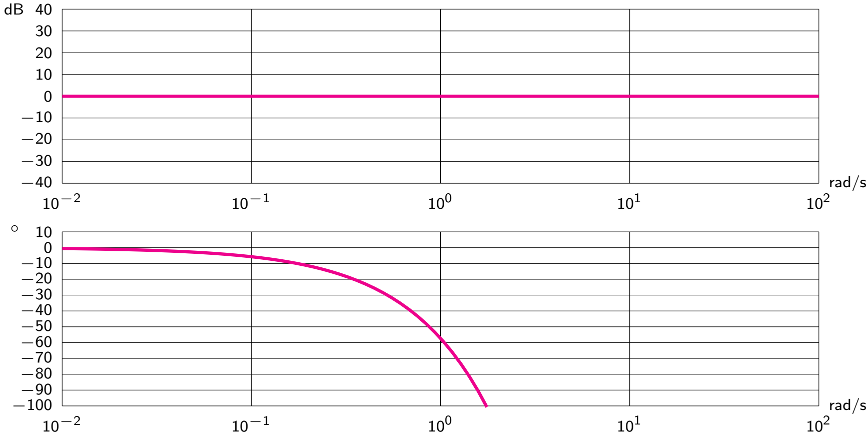
$$\arg[H(j\omega)] = \arctan\left(\frac{\mathcal{I}_m[H(j\omega)]}{\mathcal{R}_e[H(j\omega)]}\right) = -\omega\theta.$$

Obvious !

Note that in the previous formula, phase is expressed in **radians** !

Dead-time: Bode diagram[59]

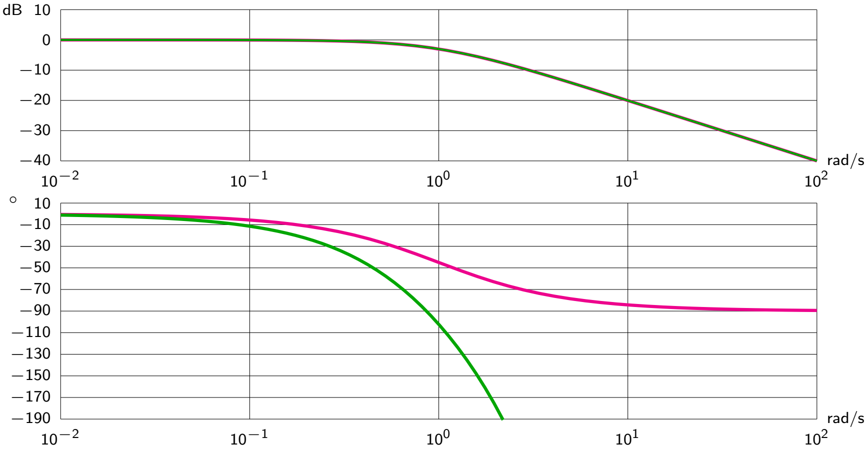
$$Y(s) = e^{-\theta s}\, X(s)$$



[59] $\theta = 1$

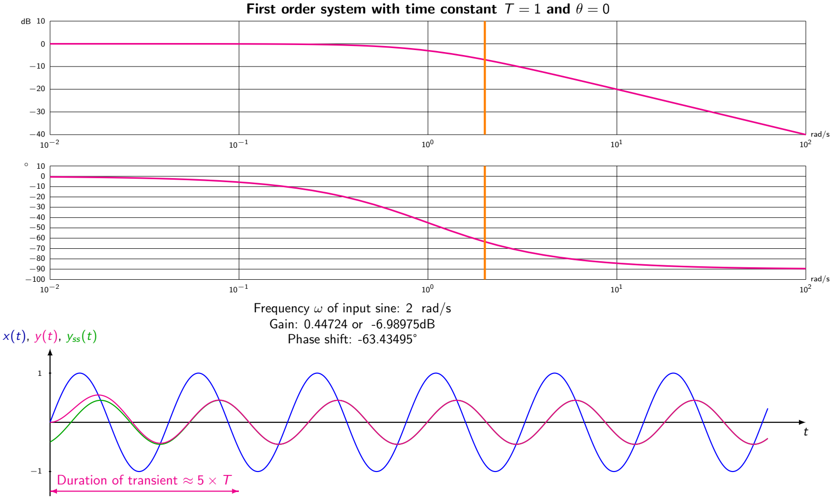
First order system with delay[60]: Bode diagram

$$Y(s) = \frac{K}{Ts+1}\, e^{-\theta s}\, X(s)$$



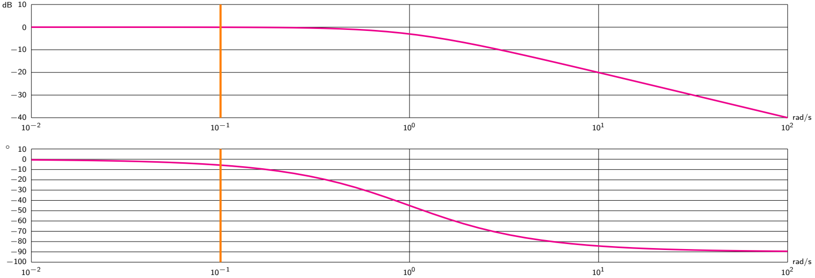
[60] $K = 1,\ T = 1,\ \theta = 0$ and $\theta = 1$

Response to input sine with zero initial conditions

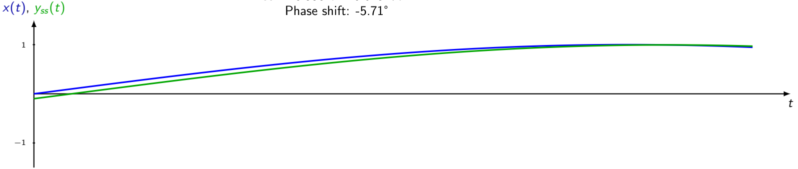


Bode diagram and **steady-state** response to input sine

First order system with time constant $T = 1$ and $\theta = 0$



Frequency ω of input sine: 0.10 rad/s
Gain: 0.995 or -0.04517dB
Phase shift: -5.71°



Filtering

F.Y.I.

- An RC system has a **low-pass behaviour**, i.e. **low frequencies are passed** and **high frequencies are attenuated**.
- Similarly, a first-order system described by the transfer function[61]

$$H_{LP}(s) = \frac{1}{Ts+1}$$

has a **low-pass behaviour**.
- The first-order system described by the transfer-function

$$H_{HP}(s) = 1 - H_{LP}(s) = 1 - \frac{1}{Ts+1} = \frac{Ts}{Ts+1}$$

has a **high-pass behaviour**.

[61] Here $K = 1$ as in general **no DC amplification/attenuation** is needed for **filtering applications**.

Filtering

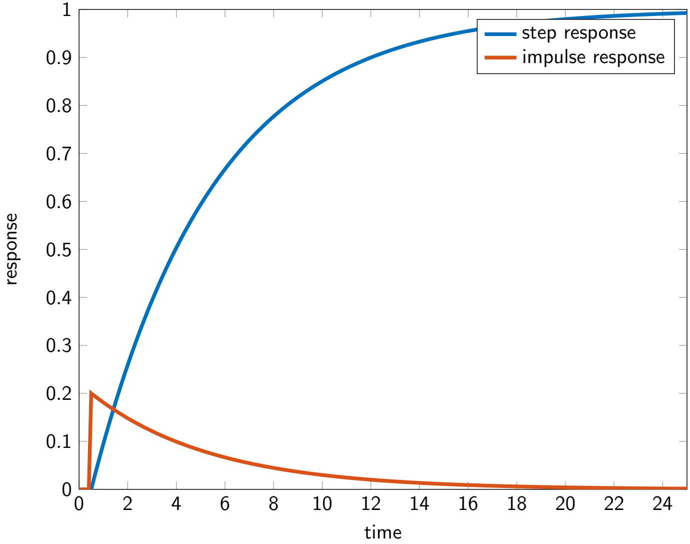
F.Y.I.

- The **cut-off frequency** is the frequency beyond which the filter will not pass signals. It is usually measured at a **specific attenuation** such as 3 dB.
- The **Roll-off** is the rate at which attenuation increases beyond the cut-off frequency. For a **first-order filter the roll-off** is -20 dB/decade.
- Important filter families such as **Butterworth, Chebyshev** filters allow us to design n-order filters with a **higher roll-off** of -20n dB/decade. These filters can also be used to implement **band-pass** and **band-stop** filter characteristics.
- In signal processing, a **filter** is a system that removes some undesired components from a signal. Most often, filtering removes some frequencies or frequency bands.

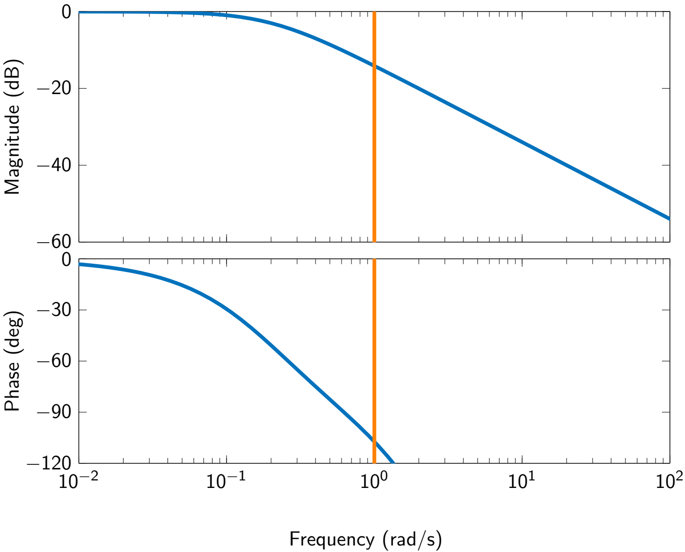
Step and impulse responses and Bode diagram

```
% pkg load control % uncomment if running octave with control toolbox

K = 1;
T = 5;
theta = 0.5;
num = [0 1];
den = [5 1];

sys = tf(num,den,'InputDelay',theta); % H(s) = K e^-(theta s)/(Ts + 1)
t = 0:0.1:25;

y1 = step(sys,t);
y2 = impulse(sys,t);

figure(1),
plot(t,[y1 y2])
legend('step response','impulse response')
xlabel('time');
ylabel('response')

figure(2),
bode(sys),
ylim([-90 0])
```

Step and impulse responses



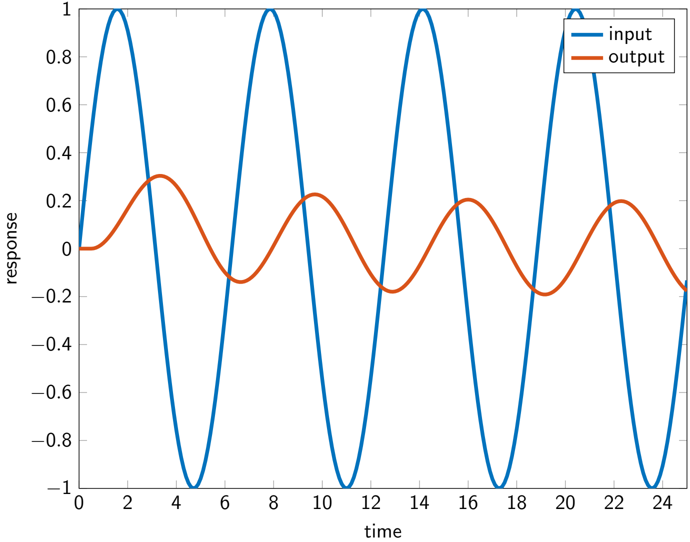
Bode diagram



Simulated responses

```
% pkg load control % uncomment if running octave with control toolbox

K = 1;
T = 5;
theta = 0.5;
num = [0 1];
den = [5 1];
sys = tf(num,den,'InputDelay',theta); % H(s) = K e^-(theta s)/(Ts + 1)
t = 0:0.1:25;
u = sin(t);
y = lsim(sys,u,t);
plot(t,[u; y'])
legend('input','output')
xlabel('time');
ylabel('response')
```

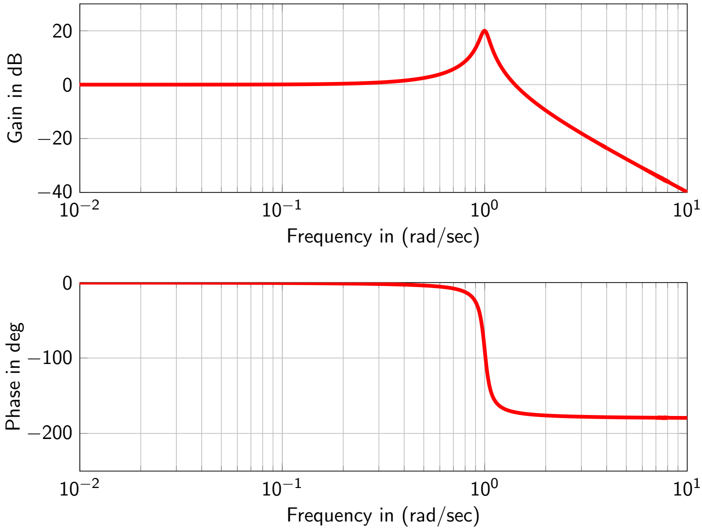
Simulated responses



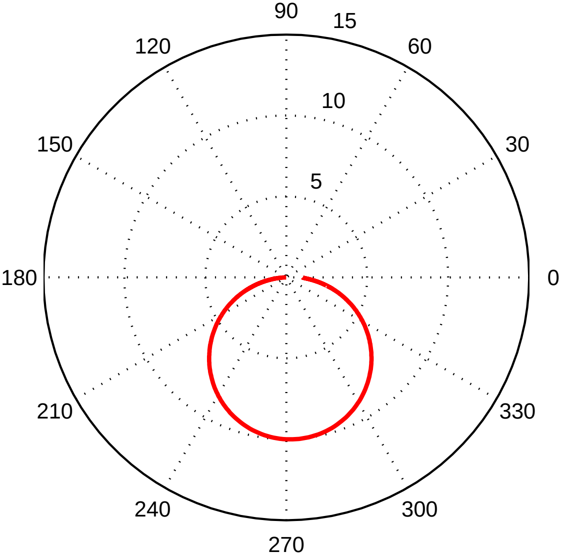
Bode and Nyquist plots

```
% pkg load control % uncomment if running octave with control toolbox

num = [0 0 1]; den = [1 0.1 1];  % Transfer function H(s) = 1/(s^2 + 0.1s + 1)
sys = tf(num,den);
w = 0:0.001:10;
[amplitude,phase] = bode(sys,w);
amplitude = reshape(amplitude,[length(amplitude) 1]);
phase = reshape(phase,[length(phase) 1]);

% Bode diagram
figure(1)
subplot(211)
semilogx(w,20*log10(amplitude),'r','Linewidth',2);
grid on; axis([0.01 10 -40 30]);
ylabel('Gain in dB'); xlabel('Frequency in rad / sec');
subplot(212)
semilogx(w,phase ,'r','Linewidth',2);
grid on; axis([0.01 10 -250 0]);
ylabel('Phase in deg'); xlabel('Frequency in rad / sec');

% Nyquist diagram
figure(2)
h1 = polar(phase*(pi/180), amplitude );
set(h1 ,'color','r','linewidth',2);
grid
```

Bode diagram



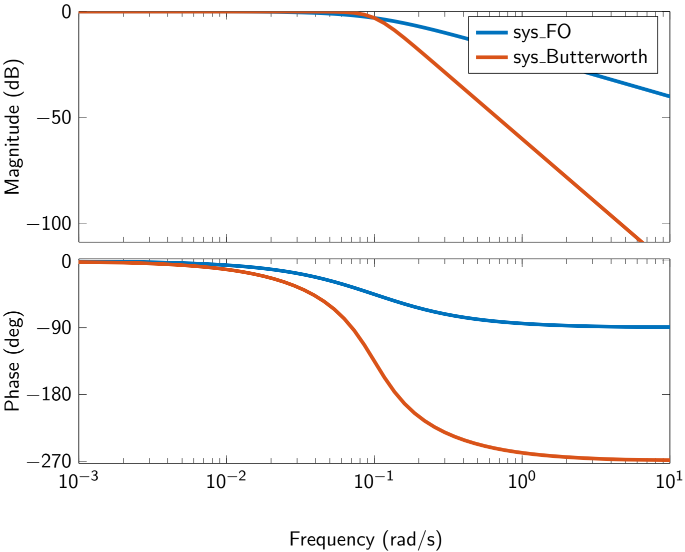
Nyquist plot



Low-pass filtering

```
wb = 0.1; T = 1/wb;

% First order low-pass
num = [0 1]; den = [T 1];
sys_FO = tf(num,den);

% Butterworth low pass filter of order n
n = 3;
[num,den] = butter(n,wb,'s');
sys_Butterworth = tf(num,den);

bode(sys_FO,sys_Butterworth)
```

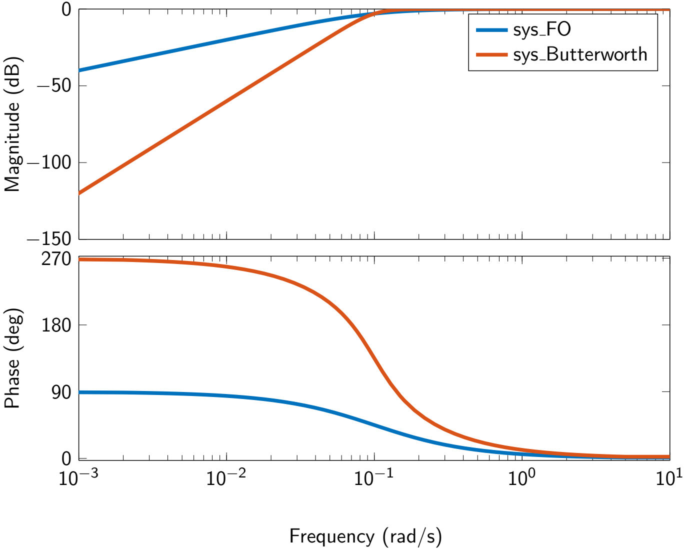
Low-pass filtering



High-pass filtering

```
wb = 0.1; T = 1/wb;

% First order high-pass
num = [T 0]; den = [T 1];
sys_FO = tf(num,den);

% Butterworth high-pass filter of order n
n = 3;
[num,den] = butter(n,wb,'high','s');
sys_Butterworth = tf(num,den);

bode(sys_FO,sys_Butterworth)
```

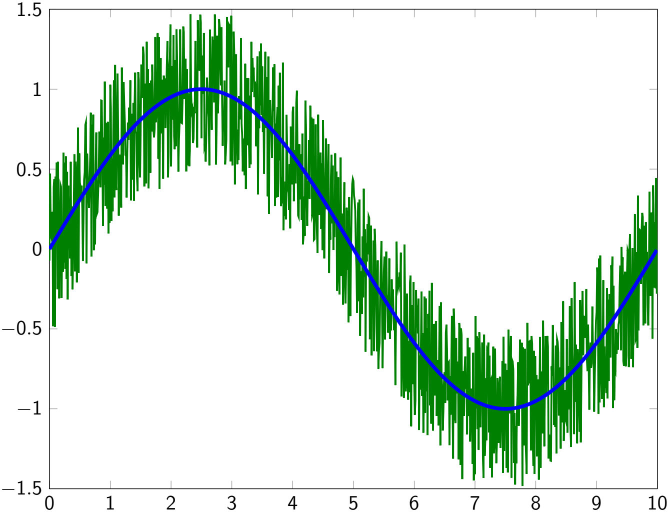
High-pass filtering



Filtering: data generation

```
clear all
close all

N = 1000;
Ts = 0.01;
t = (0:Ts:(N-1)*Ts)';

% Sinusoid
ws = 2*pi*0.1;
ys = sin(ws*t);

% Noise with high-pass filtering
yn = rand(size(t)) - 0.5;
n = 3;
[num_hp,den_hp] = butter(n,10*ws*Ts,'high','s');
yn = filter(num_hp,den_hp,yn);

figure
plot(t,ys,'b',t,ys+yn,'color',[0 0.5 0]);
title('Sinusoid + noise')
```

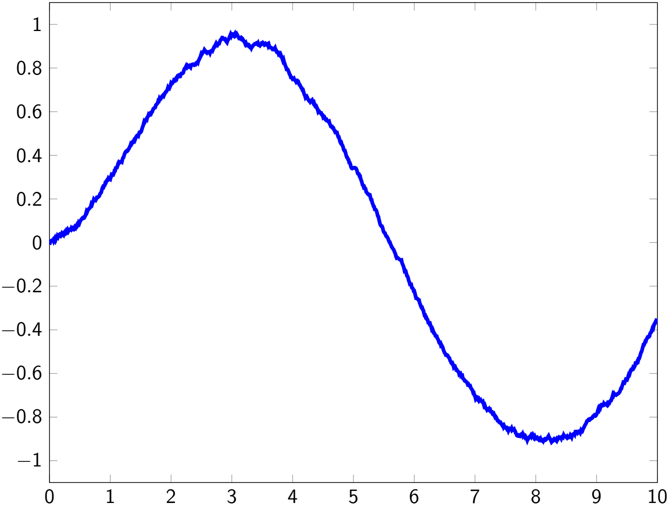
Filtering: data generation



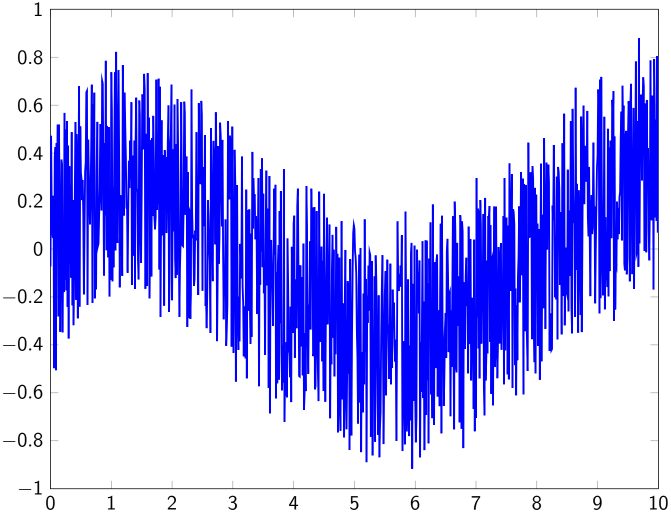
First-order filtering

```
wb = 2.5*ws; T = 1/wb;

% Low-pass
num = [0 1];
den = [T 1];
sys = tf(num,den);
sysd = c2d(sys,Ts);
[numd,dend] = tfdata(sysd,'v'); % 'v' makes sure numd en dend are not cells
yf_lp = filter(numd,dend,ys+yn);

figure
plot(t,yf_lp); title('Low frequency component');

% High pass
num = [T 0];
den = [T 1];
sys = tf(num,den);
sysd = c2d(sys,Ts);
[numd,dend] = tfdata(sysd,'v'); % 'v' makes sure numd en dend are not cells
yf_hp = filter(numd,dend,ys+yn);

figure
plot(t,yf_hp); title('High frequency component');
```

First-order filtering: low frequency component



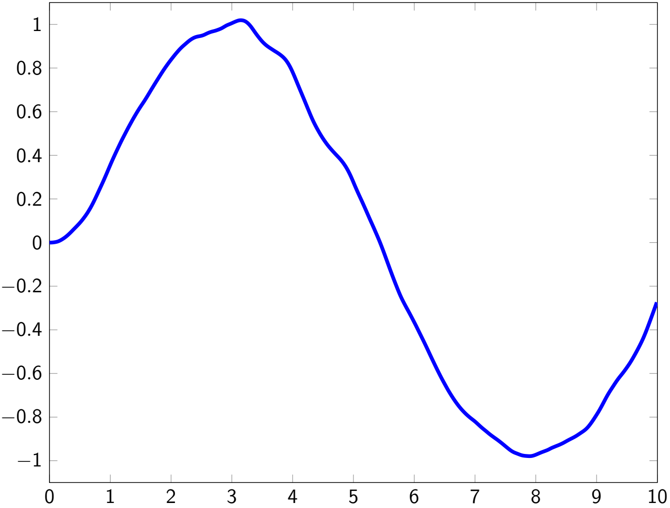
First-order filtering: high frequency component



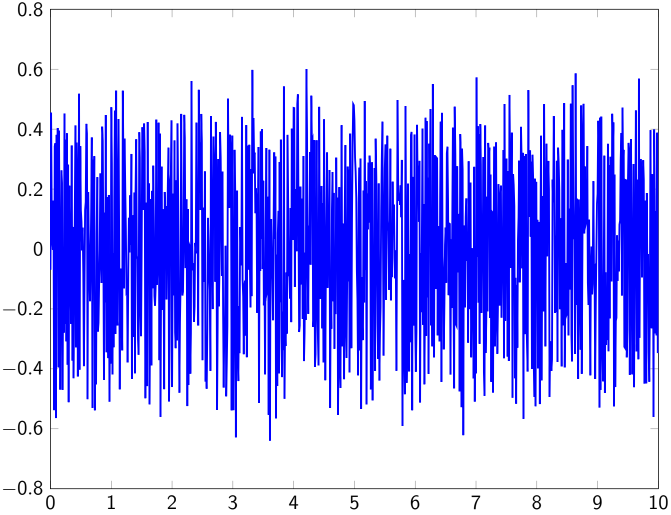
Butterworth filtering

```
wb = 2.5*ws;

n = 3; % Order of Butterworth filter

% Low-pass
[numd,dend] = butter(n,wb*Ts);
yf_lp = filter(numd,dend,ys+yn);

figure
plot(t,yf_lp); title('Low frequency component');

% High pass
[numd,dend] = butter(n,wb*Ts,'high');
yf_hp = filter(numd,dend,ys+yn);

figure
plot(t,yf_hp); title('High frequency component');
```

Butterworth filtering: low frequency component



Butterworth filtering: high frequency component



Partial fractions

Note the partial expansion of

$$\begin{aligned} P(s) &= \frac{5s+3}{s^3+3s^2-4} = \frac{5s+3}{(s-1)(s+2)^2} \\ &= \frac{-\frac{8}{9}}{s+2} + \frac{\frac{7}{3}}{(s+2)^2} + \frac{\frac{8}{9}}{s-1} \end{aligned}$$

```
num = [5 3];          % 5s + 3
den = [1 3 0 -4];     % s^3 + 3s^2 - 4
[R,P] = residue(num,den)
```

```
R =

  -0.8889
   2.3333
   0.8889

P =

   -2
   -2
    1
```

Laplace transforms

```
% pkg load symbolic % uncomment for use with octave with symbolic toolbox
syms t
f = exp(-t/2) + sin(t);
laplace(f)

syms s
F = 1/(s^2 - 1);
ilaplace(F)
```

```
ans =

1/(s + 1/2) + 1/(s^2 + 1)

ans =

exp(t)/2 - exp(-t)/2
```

One-sided Laplace transforms

$f(t)$	$\mathcal{L}[f(t)u(t)]$	ROC
$\delta(t)$	1	$\mathbb{C}$
1	$^1/_s$	$\mathcal{R}_e[s] > 0$
e^{at}	$\dfrac{1}{s-a}$	$\mathcal{R}_e[s] > a$
t^n	$\dfrac{n!}{s^{n+1}}$	$\mathcal{R}_e[s] > 0$
$\sin(\omega_0 t)$	$\dfrac{\omega_0}{s^2+\omega_0^2}$	$\mathcal{R}_e[s] > 0$
$\cos(\omega_0 t)$	$\dfrac{s}{s^2+\omega_0^2}$	$\mathcal{R}_e[s] > 0$

One-sided Laplace transforms		
$f(t)$	$\mathcal{L}[f(t)u(t)]$	ROC
$t^n\, e^{at}$	$\dfrac{n!}{(s-a)^{n+1}}$	$\mathcal{R}_e[s] > a$
$1 - e^{at}$	$\dfrac{-a}{s(s-a)}$	$\mathcal{R}_e[s] > \max(0, a)$
$e^{\sigma_0 t}\, \sin \omega_0 t$	$\dfrac{\omega_0}{(s-\sigma_0)^2 + \omega_0^2}$	$\mathcal{R}_e[s] > \sigma_0$
$e^{\sigma_0 t}\, \cos \omega_0 t$	$\dfrac{s-\sigma_0}{(s-\sigma_0)^2 + \omega_0^2}$	$\mathcal{R}_e[s] > \sigma_0$

Basic properties of one-sided Laplace transforms

Properties	$f(t)$	$F(s)$
Causal functions and constants	$\alpha f(t),\ \beta g(t)$	$\alpha F(s),\ \beta G(s)$
Linearity	$\alpha f(t) + \beta g(t)$	$\alpha F(s) + \beta G(s)$
Time shifting	$f(t-\tau)$	$e^{-\tau s} F(s)$
Frequency shifting	$e^{\lambda t} f(t)$	$F(s-\lambda)$
Multiplication by t	$tf(t)$	$-\dfrac{dF(s)}{ds}$

Basic properties of one-sided Laplace transforms		
Properties	$f(t)$	$F(s)$
Convolution	$f(t) * g(t)$	$F(s)G(s)$
Derivative	$\dfrac{df(t)}{dt}$	$sF(s) - f(0^-)$
Integral	$\int_{0^-}^{t} f(\overline{t})d\overline{t}$	$\frac{F(s)}{s}$
Expansion / contraction	$f(at), (a \neq 0)$	$\dfrac{1}{a} F\left(\dfrac{s}{a}\right)$

Basic properties of one-sided Laplace transforms	
Properties	$f(t)$ and $F(s)$
Initial value	$f(0^+) = \lim_{s\to\infty} sF(s)$
Final value f	$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$

5. Fourier frequency analysis

Some linear algebra
Projection on a function space
Fourier series
Alternative fourier series
Basic properties of Fourier series
Fourier transform
Matlab
Basic properties of the Fourier transform

Notations

F.Y.I.

- Scalar: $x = 1$
- Vector: $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- Unless specified otherwise, $\mathbf{x}$ is a column vector
- Unless specified otherwise, $\mathbf{x}^T$ is a row vector
- Matrix: $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Projection onto a subspace of $\mathbb{R}^m$

F.Y.I.

Inner product in $\mathbb{R}^m$

The **inner**, **scalar** or **dot product** of two vectors $\mathbf{x} = [x_1 \cdots x_m]^T$ and $\mathbf{y} = [y_1 \cdots y_m]^T \in \mathbb{R}^m$ is defined as

$$\mathbf{x}.\mathbf{y} = \sum_{i=1}^{m} x_i \, y_i = \mathbf{x}^T \mathbf{y}.$$

In particular, $\|\mathbf{x}\|^2 = \mathbf{x}.\mathbf{x}$ where $\|\mathbf{x}\|$ is the **norm** or **length** of $\mathbf{x}$.

Orthogonal vectors in $\mathbb{R}^m$

The vector $\mathbf{x}$ and $\mathbf{y}$ are **orthogonal** if their **inner product** is **zero**, i.e.

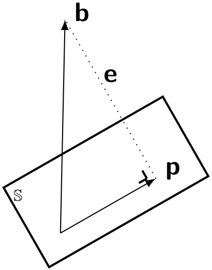
$$\mathbf{x}.\mathbf{y} = 0.$$

Projection[62] onto a subspace of $\mathbb{R}^m$

F.Y.I.

Consider n **linearly independent** column vectors

$$\mathbf{a}_1, \cdots, \mathbf{a}_n \in \mathbb{R}^m.$$



These vectors **span** an n dimensional subspace $\mathbb{S}$ of $\mathbb{R}^m$. They therefore form a **basis** of the subspace $\mathbb{S}$.

Consider a vector $\mathbf{b} \in \mathbb{R}^m$.

Find the vector $\mathbf{p} \in \mathbb{S}$ **closest** to the vector $\mathbf{b}$ in the **least square sense**. The vector p is the **projection** of b onto the **subspace** $\mathbb{S}$.

By definition, the vector $\mathbf{p} \in \mathbb{S}$ is a **linear combination** of the vectors $\mathbf{a}_i$, i.e.

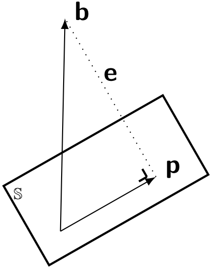
$$\mathbf{p} = \hat{x}_1\,\mathbf{a}_1 + \cdots + \hat{x}_n\,\mathbf{a}_n = \mathbf{A}\,\hat{\mathbf{x}}$$

where $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_n] \in \mathbb{R}^{m\times n}$ and $\hat{\mathbf{x}} = [\hat{x}_1 \ \cdots \ \hat{x}_n]^T$. The **coefficients** $\hat{x}_i \in \mathbb{R}$ have to be **determined**.

[62] Picture the situation with $m = 3$ and $n = 2$!

Projection onto a subspace of $\mathbb{R}^m$

F.Y.I.



By definition, the vector $\mathbf{p} \in \mathbb{S}$ is a **linear combination** of the vectors $\mathbf{a}_i$, i.e.

$$\mathbf{p} = \hat{x}_1\,\mathbf{a}_1 + \cdots + \hat{x}_n\,\mathbf{a}_n = \mathbf{A}\,\hat{\mathbf{x}}$$

where $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_n] \in \mathbb{R}^{m\times n}$ and $\hat{\mathbf{x}} = [\hat{x}_1 \ \cdots \ \hat{x}_n]^T$. The **coefficients** $\hat{x}_i \in \mathbb{R}$ have to be **determined**.

The **error** $\mathbf{e} = \mathbf{b} - \mathbf{p} = \mathbf{b} - \mathbf{A}\,\hat{\mathbf{x}}$ is **orthogonal** to $\mathbf{p}$

$$\mathbf{p}^T\mathbf{e} = \hat{\mathbf{x}}^T\mathbf{A}^T(\mathbf{b} - \mathbf{A}\,\hat{\mathbf{x}}) = 0.$$

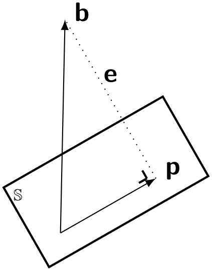
This **implies**

$$\mathbf{A}^T\mathbf{b} = \mathbf{A}^T\mathbf{A}\,\hat{\mathbf{x}}, \quad \boxed{\hat{\mathbf{x}} = [\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{b}}$$

$$\text{and } \mathbf{p} = \mathbf{A}\,\hat{\mathbf{x}} = \mathbf{A}[\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{b}.$$

Projection onto a subspace of $\mathbb{R}^m$

F.Y.I.



Suppose the vectors $\{\mathbf{a}_i\}$ form an **orthonormal basis** of $\mathbb{S}$, i.e.

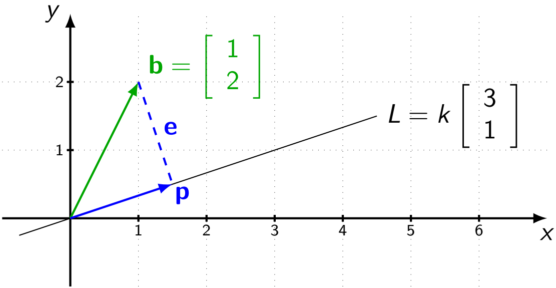
$$\mathbf{a}_i \cdot \mathbf{a}_j = \mathbf{a}_i^T \mathbf{a}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \Longrightarrow \mathbf{A}^T \mathbf{A} = \mathbf{I}.$$

The **coordinates** of $\mathbf{p}$ in the basis of $\mathbb{S}$ **simplify** to

$$\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b} \text{ or } \boxed{\hat{x}_i = \mathbf{a}_i^T \mathbf{b} = \mathbf{a}_i \cdot \mathbf{b}}$$

Projection on a line in $\mathbb{R}^2$

F.Y.I.



Projection of $\mathbf{b} \in \mathbb{R}^2$ on the line L, 1 dimensional subspace of $\mathbb{R}^2$. An orthonormal basis of L is $\mathbf{a} = [3/\sqrt{10}, 1/\sqrt{10}]^T$. The coordinates of the projection $\mathbf{p}$ in the orthonormal basis are

$$\hat{x} = \mathbf{a}^T\mathbf{b} = \mathbf{a} \cdot \mathbf{b} = 5/\sqrt{10} = \sqrt{10}/2 \Longrightarrow \mathbf{p} = \hat{x}\,\mathbf{a} = [3/2, 1/2]^T.$$

Objective

F.Y.I.

How can we **represent** a **function** $f(t)$ on an interval of length T using n **individual basis functions**

$$\{\phi_1(t), \cdots, \phi_n(t)\} \text{ ?}$$

We will examine the **approximate representation** of the function $f(t)$ as a **linear combination** of the **basis functions**, i.e.

$$f(t) \approx \sum_{i=1}^{n} c_i \, \phi_i(t).$$

Once the **basis functions** $\phi_i(t)$ have been **selected**, the task becomes **finding** the **expansion coefficients** c_i.

Signal space $\mathbb{L}_2(T)$

F.Y.I.

Signal space $\mathbb{L}_2(T)$

The **set** of **real** or **complex valued functions** of time defined on an interval of length T that are **square integrable** form the **signal space** $\mathbb{L}_2(T)$ with **norm**

$$\|s(t)\|^2 = \int_T s(t)s^\star(t)dt = \int_T |s(t)|^2 dt.$$

Inner product on $\mathbb{L}_2(T)$

The **inner product** of two function $s(t)$ and $r(t) \in \mathbb{L}_2(T)$ is

$$< s, r > \; = \int_T s(t)r^\star(t)dt, \text{ in particular } < s, s > \; = \|s(t)\|^2.$$

Signal space $\mathbb{L}_2(T)$

F.Y.I.

Orthogonal functions

Two functions $s(t)$ and $r(t)$ are **orthogonal** if their **inner product** is **zero**, i.e.

$$< s, r > \; = \int_T s(t) r^\star(t) dt = 0.$$

Orthogonal basis of a subspace of $\mathbb{L}_2(T)$

The functions $\{\phi_i(t)\} \in \mathbb{L}_2(T)$ $(i = 1, \cdots, n)$ form an **orthogonal basis** of a subspace $\mathbb{F} \subset \mathbb{L}_2(T)$ if

$$< \phi_i, \phi_j > \begin{cases} = 0, & \text{for } i \neq j, \\ \neq 0, & \text{for } i = j. \end{cases}$$

Moreover, if $< \phi_i, \phi_i > \; = 1$ for $i = 1, \cdots, n$ then the **basis** is **orthonormal**.

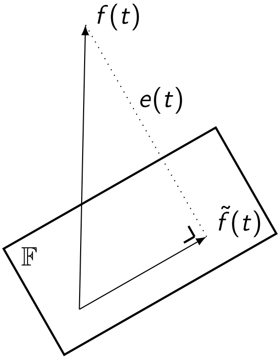
Projection on a subspace of $\mathbb{L}_2(T)$

F.Y.I.

Given n functions $\{\phi_1(t), \cdots, \phi_n(t)\}$ forming a **basis** of the **subspace** $\mathbb{F} \subset \mathbb{L}_2(T)$.

Consider a function $f(t) \in \mathbb{L}_2(T)$.

Find the function $\tilde{f}(t) \in \mathbb{F}$ **closest** to $f(t)$ in the **least square sense**. The function $\tilde{f}(t)$ is in fact the **projection** of $f(t)$ on $\mathbb{F} \subset \mathbb{L}_2(T)$.



The function $\tilde{f}(t) \in \mathbb{F}$ is a **linear combination** of the vectors $\{\phi_i(t)\}$, i.e. $\tilde{f}(t) = \sum_{i=1}^{n} c_i\, \phi_i(t)$ where the **coefficients** $c_i \in \mathbb{C}$ $(i = 1, \cdots, n)$ need to be **determined**.

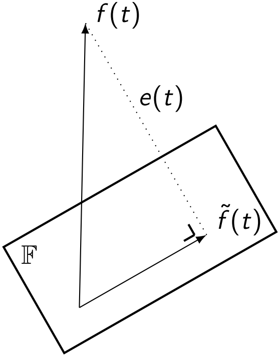
The approximation **error** $e(t) = f(t) - \tilde{f}(t)$ is **orthogonal** to $\tilde{f}(t)$. We have that

$$\begin{aligned} < e, \tilde{f} > &= < f - \tilde{f}, \tilde{f} >, \\ &= < f, \tilde{f} > - < \tilde{f}, \tilde{f} > = 0 \end{aligned}$$

$$\Rightarrow < f, \tilde{f} > = < \tilde{f}, \tilde{f} > \cdot$$

Projection on a subspace of $\mathbb{L}_2(T)$

F.Y.I.



The **error** $e(t) = f(t) - \tilde{f}(t)$ is **orthogonal** to $\tilde{f}(t) = \sum_{i=1}^{n} c_i\, \phi_i(t)$. We have that

$$
\begin{aligned}
< f, \tilde{f} > &= < \tilde{f}, \tilde{f} > \\
\sum_{i=1}^{n} c_i < f, \phi_i > &= \sum_{i=1}^{n} c_i < \phi_i, \tilde{f} > .
\end{aligned}
$$

This implies

$$
< f, \phi_i > = < \phi_i, \tilde{f} > = \sum_{j=1}^{n} c_j < \phi_i, \phi_j > .
$$

Conclusion

$$
\boxed{< f, \phi_i > = \sum_{j=1}^{n} c_j < \phi_i, \phi_j >}
$$

There are n **complex equations** with n **complex unknowns**. It is therefore **possible** to determine the coefficients $c_i \in \mathbb{C}$.

Projection on a subspace of $\mathbb{L}_2(T)$

F.Y.I.

Orthogonal basis

It is **advantageous** to work with an **orthogonal basis** as

$$< \phi_i, \phi_j > \; = 0, \text{ for } i \neq j.$$

The approximation equation **becomes** $< f, \phi_i > \; = c_i < \phi_i, \phi_i >$, i.e.

$$\boxed{c_i = \frac{< f, \phi_i >}{< \phi_i, \phi_i >}}$$

Orthonormal basis

If the basis is **orthonormal**, we obtain[63] $\boxed{c_i = \; < f, \phi_i >}$

[63] This should look familiar to you !

F.Y.I.

Projection on a subspace of $\mathbb{L}_2(T)$

Given n functions $\{\phi_1(t), \cdots, \phi_n(t)\}$ forming an **orthonormal basis** of the subspace $\mathbb{F} \subset \mathbb{L}_2(T)$, i.e.

$$< \phi_i, \phi_j > = \int_T \phi_i(t)\phi_j^\star(t)\, dt = \begin{cases} 0, & \text{for } i \neq j, \\ 1, & \text{for } i = j. \end{cases}$$

Consider a function $f(t) \in \mathbb{L}_2(T)$. We are interested in $\tilde{f}(t) \in \mathbb{F}$ **closest** to $f(t)$ in the **mean square sense**. The function $\tilde{f}(t)$ is the **projection** of $f(t)$ on $\mathbb{F} \subset \mathbb{L}_2(T)$ and can be written as

$$\boxed{\tilde{f}(t) = \sum_{i=1}^{n} c_i\, \phi_i(t)}$$

with

$$c_i = \, < f, \phi_i > \quad \Longrightarrow \quad \boxed{c_i = \int_T f(t)\phi_i^\star(t)\, dt}$$

Choice of the basis function

F.Y.I.

- There exist a **multitude** of **orthogonal function sets**: Legendre and Laguerre polynomials, etc.
- The **choice** of the type of basis function depends on the **nature of the problem** that is considered.
- It might even be possible that a non orthogonal basis is best suited to represent a given function.
- In the sequel, the basis that is considered is the **Fourier basis**, i.e. a **set of complex exponential functions**.

Standard $\mathbb{R}^3$ basis and Fourier basis

- Standard $\mathbb{R}^3$ basis

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \; \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \; \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Orthonormal basis, i.e.

$$\mathbf{e}_k . \mathbf{e}_l = \begin{cases} 1, & k = l, \\ 0, & k \neq l. \end{cases}$$

- Scalar or inner product

$$\mathbf{v}.\mathbf{w} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} . \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = v_1 w_1 + v_2 w_2 + v_3 w_3$$

- Decomposition in $\mathbb{R}^3$ basis

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = c_1 \, \mathbf{e}_1 + c_2 \, \mathbf{e}_2 + c_3 \, \mathbf{e}_3 = \sum_{k=1}^{3} c_k \, \mathbf{e}_k$$

$$c_k = \mathbf{v}.\mathbf{e}_k = v_k$$

Standard $\mathbb{R}^3$ basis and Fourier basis

- Standard $\mathbb{R}^3$ basis

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Orthonormal basis, i.e.

$$\mathbf{e}_k . \mathbf{e}_l = \begin{cases} 1, & k = l, \\ 0, & k \neq l. \end{cases}$$

- Scalar or inner product

$$\mathbf{v}.\mathbf{w} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} . \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = v_1 w_1 + v_2 w_2 + v_3 w_3$$

- Decomposition in $\mathbb{R}^3$ basis

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = c_1\,\mathbf{e}_1 + c_2\,\mathbf{e}_2 + c_3\,\mathbf{e}_3 = \sum_{k=1}^{3} c_k\,\mathbf{e}_k$$

$$c_k = \mathbf{v}.\mathbf{e}_k = v_k$$

- Fourier basis of complex exponentials

$$F_k(t) = e^{jk\omega_0 t}, \ k \text{ integer}$$

- Orthonormal basis, i.e.

$$< F_k, F_l > = \begin{cases} 1, & k = l, \\ 0, & k \neq l. \end{cases}$$

- Inner product

$$< F_k, F_l > = \frac{1}{T} \int_T F_k(t) F_l^\star(t) dt$$

- Decomposition in Fourier basis

$$f(t) = \sum_{k=-\infty}^{\infty} c_k\, F_k(t)$$

$$c_k = < f, F_k > = \frac{1}{T} \int_T f(t) e^{-jk\omega_0 t} dt$$

Fourier basis

Fourier basis

The **complex Fourier basis** is composed of the **complex exponentials**

$$F_k(t) = e^{jk\omega_0 t},\ k \text{ integer and } \omega_0 = \frac{2\pi}{T}.$$

The **Fourier basis functions** are **orthonormal** over a period T, i.e.

$$< F_k, F_l > = \frac{1}{T}\int_{t_0}^{t_0+T} F_k(t)F_l^\star(t)dt = \begin{cases} 1, & k = l, \\ 0, & k \neq l. \end{cases}$$

Fourier series representation

Fourier series representation

The **Fourier series** representation of a **periodic** signal $f(t)$, of period T, is given by an **infinite sum** of **weighted complex exponentials** with **frequencies multiples** of the **signal's fundamental frequency** ω_0, i.e.

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \, F_k(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{jk\omega_0 t}, \ \omega_0 = \frac{2\pi}{T}$$

where the **Fourier coefficients** are found according to

$$c_k \ = \ < f, F_k > \ = \ \frac{1}{T} \int_{t_0}^{t_0+T} f(t) F_k^{\star} dt = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-jk\omega_0 t} dt$$

for $k = 0, \pm 1, \pm 2, \cdots$ and any t_0. The **Fourier coefficients** c_k are **complex** and can be obtained from any period of $f(t)$.

Fourier series representation

Fourier series representation

The **Fourier series** representation of a **periodic** signal $f(t)$, of period T, is given by an **infinite sum** of **weighted complex exponentials** with **frequencies multiples** of the **signal's fundamental frequency** ω_0, i.e.

$$\boxed{f(t) = \sum_{k=-\infty}^{\infty} c_k\, e^{jk\omega_0 t},\ \omega_0 = \frac{2\pi}{T}}$$

where the **Fourier coefficients** are found according to

$$\boxed{c_k = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)e^{-jk\omega_0 t}dt}$$

for $k = 0, \pm1, \pm2, \cdots$ and any t_0. The **Fourier coefficients** c_k are **complex** and can be obtained from any period of $f(t)$.

Convergence of Fourier series

Convergence

The **Fourier series** of a **piecewise smooth** (continuous or discontinuous) **periodic** signal $f(t)$ **converges** for all values of t.

The mathematician **Dirichlet** showed that for the Fourier series to **converge**[64] to the periodic signal $f(t)$, the signal should satisfy the following **sufficient** (not necessary) **conditions** over a **period**:

- be **absolutely integrable**,
- have a **finite number** of **maxima** and **minima**,
- have a **finite number** of **discontinuities**.

[64] We can speak of **pointwise convergence**.

Convergence of Fourier series

Convergence

The infinite series equals $f(t)$ at **every continuity point** and equals the **average**

$$0.5(f(t_0^+) + f(t_0^-))$$

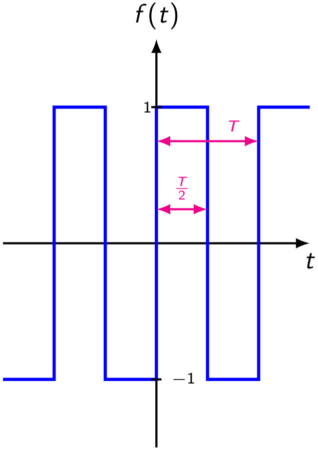
of the right limit $f(t_0^+)$ and the left limit $f(t_0^-)$ at every **discontinuity point** t_0. Here

$$f(t_0^+) = \lim_{\substack{t \to t_0 \\ >}} f(t) \text{ and } f(t_0^-) = \lim_{\substack{t \to t_0 \\ <}} f(t).$$

If $f(t)$ is **continuous** everywhere, the series **converges absolutely and uniformly**.

Although the Fourier series converges to the arithmetic average at discontinuities, a **ringing** before and after the discontinuity points can be observed. This is called the **Gibbs phenomenon**.

Fourier series: example



The **Fourier coefficients** c_k are

$$
\begin{aligned}
c_k &= \frac{1}{T}\int_0^T f(t)e^{-jk\omega_0 t}dt \\
&= \frac{1}{T}\left(\int_0^{\frac{T}{2}} 1e^{-j2k\pi\frac{t}{T}}dt + \int_{\frac{T}{2}}^{T} -1e^{-j2k\pi\frac{t}{T}}dt\right) \\
&= -\frac{1}{T}\frac{T}{j2k\pi}\left([e^{-j2k\pi\frac{t}{T}}]_0^{\frac{T}{2}} - [e^{-j2k\pi\frac{t}{T}}]_{\frac{T}{2}}^{T}\right) \\
&= -\frac{1}{j2k\pi}\left(e^{-jk\pi} - 1 - \underbrace{e^{-j2k\pi}}_{=1} + e^{-jk\pi}\right) \\
&= -\frac{1}{jk\pi}\left(e^{-jk\pi} - 1\right)
\end{aligned}
$$

$$\boxed{c_k = 0 \text{ for } k \textbf{ even} \text{ or } \textbf{zero} \text{ and } c_k = \frac{2}{jk\pi} \text{ for } k \textbf{ odd}.}$$

Fourier series: example

$$\begin{aligned}
f(t) &= \sum_{k=-\infty}^{\infty} c_k\, e^{j2k\pi\frac{t}{T}},\ c_k = \begin{cases} 0, & \text{k } \textbf{even} \text{ or } \textbf{zero} \\ \dfrac{2}{jk\pi}, & \text{k } \textbf{odd} \end{cases} \\
&= \frac{2}{j\pi}\Big(\cdots - \frac{1}{5}e^{-j10\pi\frac{t}{T}} - \frac{1}{3}e^{-j6\pi\frac{t}{T}} - e^{-j2\pi\frac{t}{T}} \\
&\qquad e^{j2\pi\frac{t}{T}} + \frac{1}{3}e^{j6\pi\frac{t}{T}} + \frac{1}{5}e^{j10\pi\frac{t}{T}} + \cdots\Big)
\end{aligned}$$

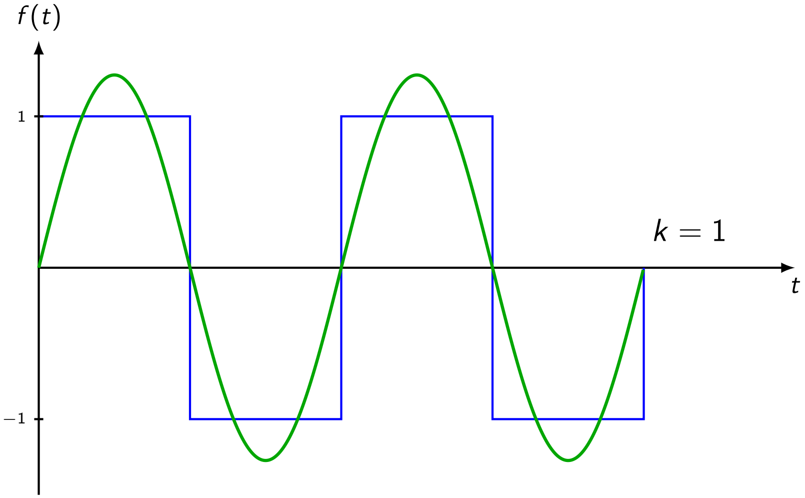
Note that

$$\frac{1}{k}\left(e^{j2k\pi\frac{t}{T}} - e^{-j2k\pi\frac{t}{T}}\right) = \frac{2j}{k}\sin(2k\pi\frac{t}{T})$$

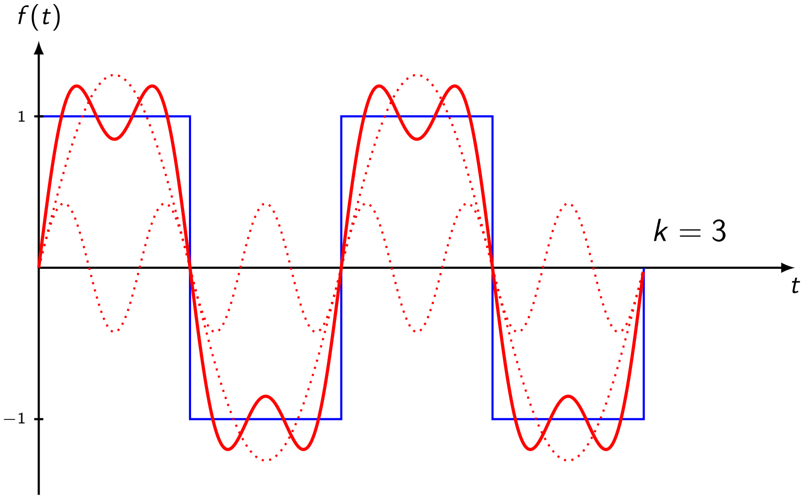
Therefore

$$\boxed{f(t) = \sum_{\substack{k>0 \\ \text{odd}}}^{\infty} \frac{4}{k\pi}\sin\left(2k\pi\frac{t}{T}\right)}$$

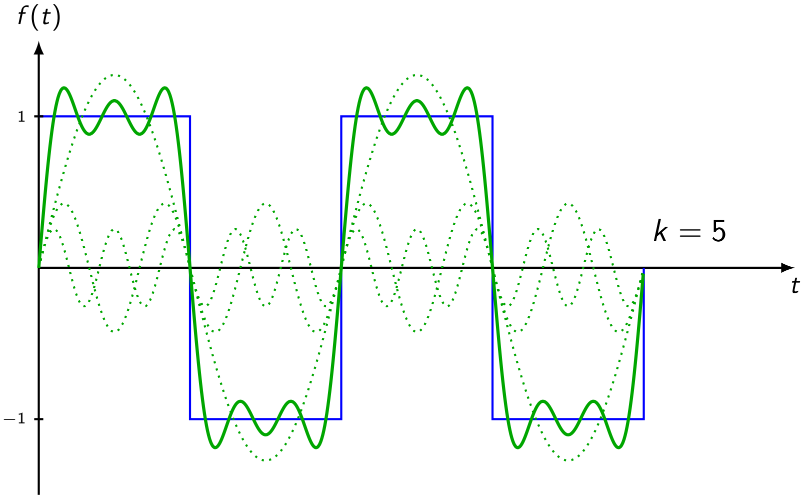
Fourier series representation: example



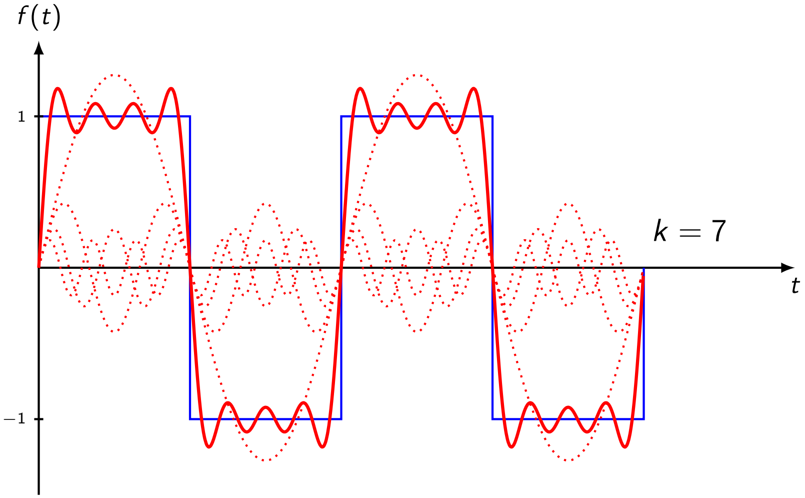
Fourier series representation: example



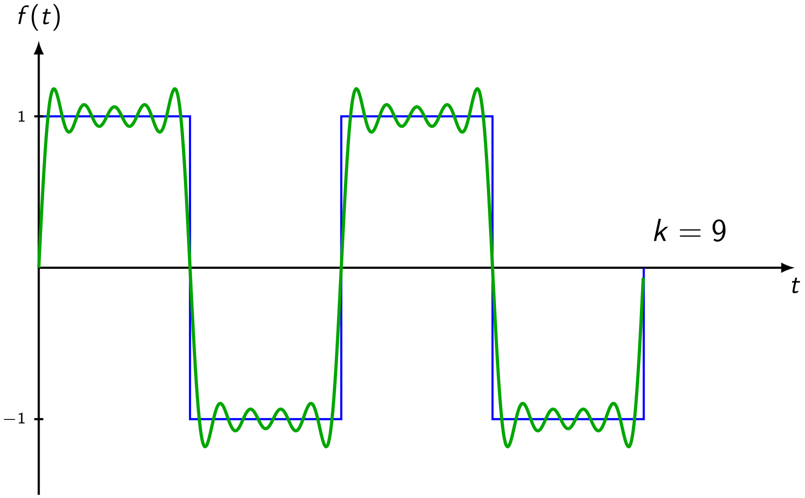
Fourier series representation: example



Fourier series representation: example



Fourier series representation: example



Fourier series representation

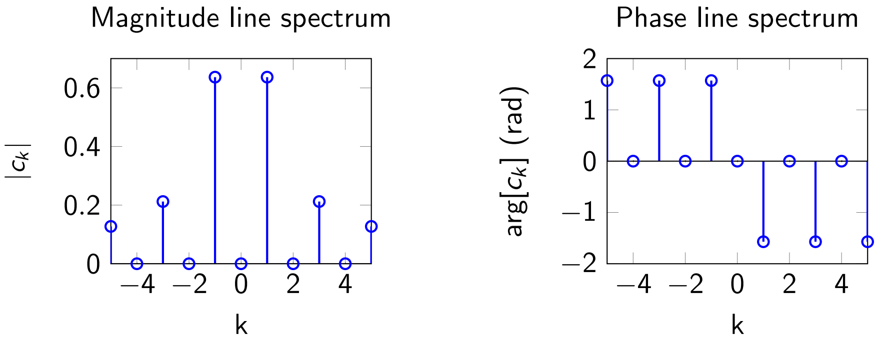
Representation

A **periodic** signal $f(t)$ of period T, is represented in the **frequency domain** by its

- **magnitude line spectrum**: magnitude $|c_k|$ versus $k\omega_0$,
- **phase line spectrum**: argument $\arg[c_k]$ versus $k\omega_0$.

The **power line spectrum** $|c_k|^2$ versus $k\omega_0$ of $f(t)$ displays the distribution of the **power** of the signal over frequency.

Fourier series representation: example



Fourier series representation: general case

Properties of the Fourier series

For a signal $f(t)$ of **period** T, we can **always** write

$$\begin{aligned} f(t) &= \sum_{k=-\infty}^{\infty} c_k\, e^{jk\omega_0 t} \\ &= c_0 + \sum_{k=-\infty}^{-1} c_k\, e^{jk\omega_0 t} + \sum_{k=1}^{\infty} c_k\, e^{jk\omega_0 t} \\ &= c_0 + \sum_{k=1}^{\infty} \left[c_k\, e^{jk\omega_0 t} + c_{-k}\, e^{-jk\omega_0 t} \right] \end{aligned}$$

Fourier series representation

DC component and harmonics

- $c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)dt$ is the **DC or average component** of $f(t)$.
- $c_1 e^{j\omega_0 t} + c_{-1} e^{-j\omega_0 t}$ is the **fundamental component** of $f(t)$.
- $c_k e^{jk\omega_0 t} + c_{-k} e^{-jk\omega_0 t}$ is the **kth harmonic component**[65] of $f(t)$.

[65] $k > 1$

Fourier series representation: real-valued functions

Symmetry of line spectra

For a **real-valued** periodic signal $f(t)$, of period T, represented in the **frequency domain** by the **Fourier coefficients** $\{c_k = |c_k|e^{j\arg[c_k]}\}$ at **harmonic frequencies** $\{k\omega_0 = \frac{2k\pi}{T}\}$, we have that

$$c_k = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)e^{-jk\omega_0 t}dt = c^{\star}_{-k}.$$

This yields

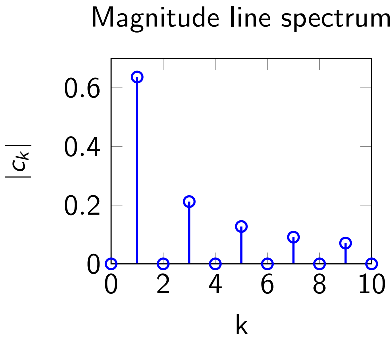
$$c_{-k} = |c_{-k}|e^{j\arg[c_{-k}]} = c_k^{\star} = |c_k|e^{-j\arg[c_k]}$$

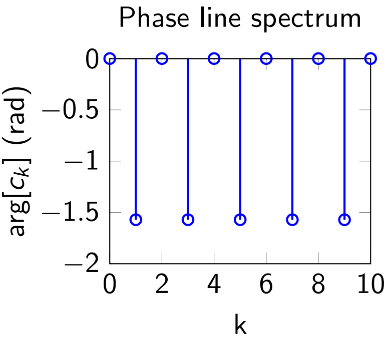
and therefore

- $|c_k| = |c_{-k}|$, i.e. the **magnitude** line **spectrum** is **even**,
- $\arg[c_k] = -\arg[c_{-k}]$, i.e. the **phase** line **spectrum** is **odd**.

For **real-valued signals** we only need to display the spectra for $k \geq 0$.

Fourier series and real-valued functions: example





Fourier series representation: real-valued functions

The Fourier series of a **real-valued** function $f(t)$ of period T is

$$
\begin{aligned}
f(t) &= \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} |c_k| e^{j\,\mathrm{arg}[c_k]} e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} |c_k| e^{j(k\omega_0 t + \mathrm{arg}[c_k])} \\
&= c_0 + \sum_{k=1}^{\infty} |c_k| e^{j(k\omega_0 t + \mathrm{arg}[c_k])} + |c_{-k}| e^{j(-k\omega_0 t + \mathrm{arg}[c_{-k}])} \\
&= c_0 + \sum_{k=1}^{\infty} |c_k| \left(e^{j(k\omega_0 t + \mathrm{arg}[c_k])} + e^{-j(k\omega_0 t + \mathrm{arg}[c_k])} \right) \\
&= c_0 + \sum_{k=1}^{\infty} 2\,|c_k| \cos(k\omega_0 t + \mathrm{arg}[c_k])
\end{aligned}
$$

Fourier series representation: real-valued functions

DC component and harmonics

- $c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)dt$ is the **DC or average component** of $f(t)$.
- $2\,|c_1|\cos(\omega_0 t + \arg[c_1])$ is the **fundamental component** of $f(t)$.
- $2\,|c_k|\cos(k\omega_0 t + \arg[c_k])$ is the **kth harmonic component**[66] of $f(t)$.

[66] $k > 1$

Fourier series representation

Properties of the Fourier series: reflection

Assume the **Fourier series** of a function $f(t)$ of period T is

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{jk\omega_0 t}, \;\; \omega_0 = \frac{2\pi}{T}$$

where the **Fourier coefficients** are computed according to

$$c_k \; = \; \frac{1}{T} \int_{t_0}^{t_0+T} f(t) e^{-jk\omega_0 t} dt.$$

The **Fourier series** of the **reflected function** $f(-t)$ is

$$f(-t) = \sum_{k=-\infty}^{\infty} c_k \, e^{-jk\omega_0 t} = \sum_{k=-\infty}^{\infty} c_{-k} \, e^{jk\omega_0 t}.$$

Even real-valued functions

Properties of the Fourier series for real-valued **even** functions

For a **real-valued even** function $f(t)$ of period T, we have

$$f(-t) = f(t) \Rightarrow c_k = c^\star_{-k} = c_{-k} \Rightarrow c_k = c^\star_k.$$

The **Fourier coefficients** $\{c_k\}$ are **real**. The **Fourier series** yields

$$\begin{aligned} f(t) &= c_0 + \sum_{k=1}^{\infty} \left[c_k \, e^{jk\omega_0 t} + c_{-k} \, e^{-jk\omega_0 t} \right], \\ &= c_0 + \sum_{k=1}^{\infty} c_k \left[\, e^{jk\omega_0 t} + \, e^{-jk\omega_0 t} \right], \\ &= c_0 + \sum_{k=1}^{\infty} 2 \, c_k \cos(k\omega_0 t). \end{aligned}$$

Odd real-valued functions

Properties of the Fourier series for real-valued **odd** functions

For a **real-valued odd** function $f(t)$ of period T, we have

$$f(-t) = -f(t) \Rightarrow c_k = c_{-k}^{\star} = -c_{-k} \Rightarrow c_k = -c_k^{\star}.$$

The **Fourier coefficients** $\{c_k\}$ are **purely imaginary**. The **Fourier series** yields

$$\begin{aligned} f(t) &= \sum_{k=1}^{\infty} \left[c_k \, e^{jk\omega_0 t} + c_{-k} \, e^{-jk\omega_0 t} \right], \\ &= \sum_{k=1}^{\infty} c_k \left[e^{jk\omega_0 t} - e^{-jk\omega_0 t} \right], \\ &= \sum_{k=1}^{\infty} 2j \, c_k \sin(k\omega_0 t). \end{aligned}$$

Fourier coefficients from Laplace

The **computation** of the c_k **Fourier coefficients** requires **integration** that for some signals can be rather **complicated**. The integration can be **avoided** whenever we know the **Laplace transform** of a **period** of the signal $f(t)$.

Fourier coefficients from Laplace transform

If the Laplace transform $F_T(s)$ of **one period** of the **periodic** signal $f(t)$ is **known**, then the **Fourier coefficients** of $f(t)$ are given by

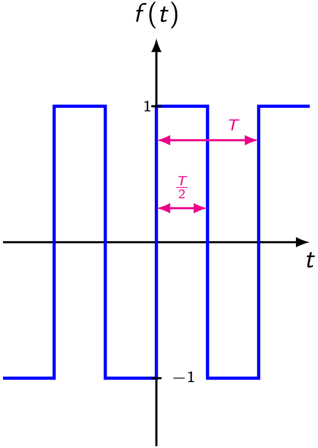
$$\boxed{c_k = \frac{1}{T}\left(F_T(s)|_{s=jk\omega_0}\right),\ \omega_0 = \frac{2\pi}{T}}$$

Note that

$$F_T(s) = \mathcal{L}[f_T(t)]$$

where $f_T(t) = f(t)\left(u(t-t_0) - u(t-t_0-T)\right)$ is the **restriction** of $f(t)$ over **any period**.

Fourier series representation: example



Note that

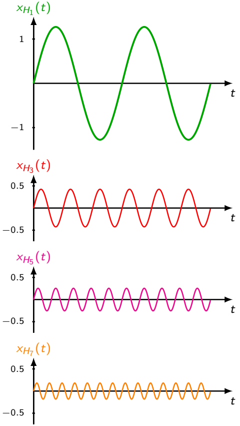
$$\begin{aligned} f_T(t) &= u(t) - 2u(t - \tfrac{T}{2}) + u(t-T) \text{ and} \\ F_T(s) &= \frac{1}{s}\left(1 - 2e^{-\frac{T}{2}s} + e^{-Ts}\right) \end{aligned}$$

The **Fourier coefficients** are

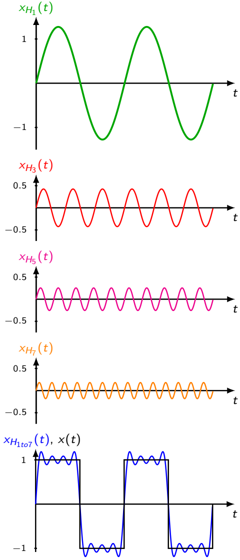
$$\begin{aligned} c_k &= \frac{1}{T}\frac{1}{s}\left(1 - 2e^{-\frac{T}{2}s} + e^{-Ts}\right)\Big|_{s=jk\omega_0=\frac{j2k\pi}{T}} \\ &= \frac{1}{j2k\pi}\left(1 - 2e^{-jk\pi} + \underbrace{e^{-j2k\pi}}_{=1}\right) \\ &= \frac{1}{jk\pi}\left(1 - e^{-jk\pi}\right) \end{aligned}$$

$c_k = 0$ for k **even** or **zero** and $c_k = \dfrac{2}{jk\pi}$ for k **odd**.

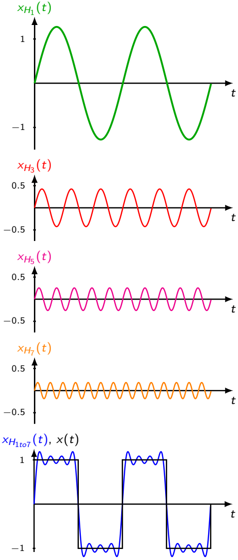
Fourier series: non-sinusoidal periodic signals

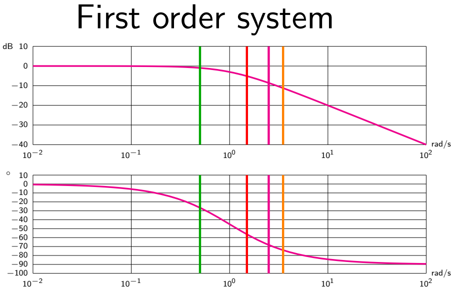


Fourier series: non-sinusoidal periodic signals

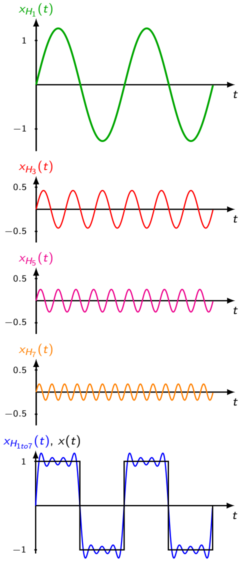


Fourier series: non-sinusoidal periodic signals

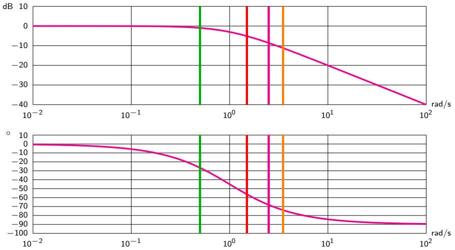


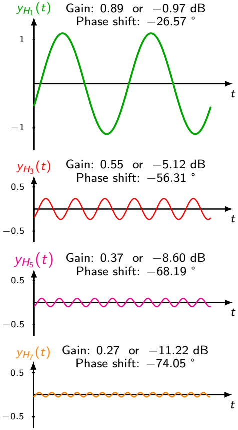


Fourier series: non-sinusoidal periodic signals

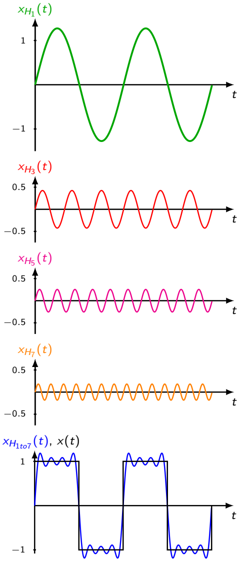


First order system

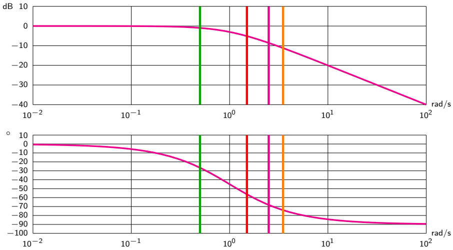


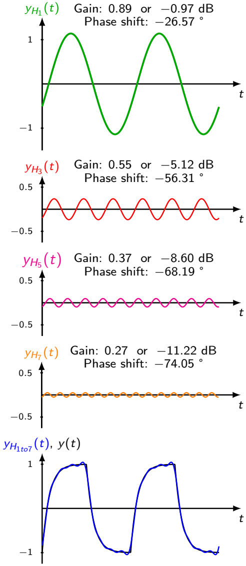


Fourier series: non-sinusoidal periodic signals



First order system





Fourier series representation: real-valued functions

F.Y.I.

The Fourier series of a **real-valued** function $f(t)$ of period T is

$$
\begin{aligned}
f(t) &= \sum_{k=-\infty}^{\infty} c_k\, e^{jk\omega_0 t} = c_0 + \sum_{k=-\infty}^{-1} c_k\, e^{jk\omega_0 t} + \sum_{k=1}^{\infty} c_k\, e^{jk\omega_0 t} \\
&= c_0 + \sum_{k=1}^{\infty} \left[c_k\, e^{jk\omega_0 t} + c_{-k}\, e^{-jk\omega_0 t}\right] \\
&= c_0 + \sum_{k=1}^{\infty} \left[c_k\, e^{jk\omega_0 t} + c_k^{\star}\, e^{-jk\omega_0 t}\right] \\
&= c_0 + \sum_{k=1}^{\infty} \left[(c_k + c_k^{\star}) \cos(k\omega_0 t) + j(c_k - c_k^{\star}) \sin(k\omega_0 t)\right] \\
&= c_0 + 2\sum_{k=1}^{\infty} \left[\mathcal{R}_e[c_k] \cos(k\omega_0 t) - \mathcal{I}_m[c_k] \sin(k\omega_0 t)\right]
\end{aligned}
$$

Trigonometric Fourier series

F.Y.I.

Trigonometric Fourier series

The trigonometric Fourier series of a **real-valued** periodic signal $f(t)$ of period T is given by

$$f(t) = a_0 + 2\sum_{k=1}^{\infty}\left[a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)\right]$$

where

$$\begin{aligned} a_k &= \frac{1}{T}\int_0^T f(t)\cos(k\omega_0 t)dt,\ \ k = 0, 1, 2, \cdots \\ b_k &= \frac{1}{T}\int_0^T f(t)\sin(k\omega_0 t)dt,\ \ k = 1, 2, \cdots \end{aligned}$$

and a_0 is the **DC** or **average component**.

The basis functions $\{\cos(k\omega_0 t), \sin(k\omega_0 t)\}$ are **orthonormal**.

Connection between the two forms of Fourier series

F.Y.I.

Connection between c_k and a_k, b_k

$$c_0 = a_0 = \frac{1}{T}\int_0^T f(t)dt$$

$$c_k = a_k - jb_k \Rightarrow \begin{cases} |c_k| = \sqrt{a_k^2 + b_k^2} \\ \angle c_k = -\arctan\left[\frac{b_k}{a_k}\right] \end{cases}$$

Proof: $\mathcal{R}_e[c_k] = a_k$ and $\mathcal{I}_m[c_k] = -b_k \;\Rightarrow\; a_k - jb_k = c_k$.

Basic properties of Fourier series		
Properties	Time domain	Frequency domain
Signals of period T	$f(t)$, $g(t)$	c_k^f, c_k^g
Linearity	$\alpha f(t) + \beta g(t)$	$\alpha c_k^f + \beta c_k^g$
Time shift	$f(t-\tau)$	$e^{-j\tau\omega_0} c_k^f$
Frequency shift	$e^{jm\omega_0 t} f(t)$	c_{k-m}^f
Differentiation	$\dfrac{df(t)}{dt}$	$jk\omega_0\, c_k^f$

Basic properties of Fourier series		
Properties	Time domain	Frequency domain
Parseval's identity	$\frac{1}{T}\int_T \lvert f(t)\rvert^2 dt$	$\sum_k \lvert c_k^f\rvert^2$
Symmetry	real-valued $f(t)$	$\begin{cases} c_k^f = c_{-k}^{f\star} \\ \lvert c_k^f\rvert = \lvert c_{-k}^f\rvert, \\ \arg[c_k^f] = -\arg[c_{-k}^f] \end{cases}$
Simplification	real-valued even $f(t)$	$c_k^f = c_{-k}^{f\star} = c_{-k}^f$ $\{c_k\}$ **real**
Simplification	real-valued odd $f(t)$	$c_k^f = c_{-k}^{f\star} = -c_{-k}^f$ $\{c_k\}$ **purely imaginary**

Introduction

- The frequency representation of signals is a tool of **great significance** in signal processing.
- It would be nice to complete the Fourier representation of signals by extending it to **aperiodic signals**.
- **Idea**: to obtain the Fourier representation of aperiodic signals, we use the Fourier series representation in a **limiting process**, i.e. the **aperiodic signal** $f(t)$ is seen as a **periodic signal of infinite period**. **Mathematically**, this yields

$$\boxed{f(t) = \lim_{T \to \infty} \tilde{f}(t)}$$

where $\tilde{f}(t)$ is a function of **period** T.

Fourier transform

F.Y.I.

$$\tilde{f}(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{jk\omega_0 t}, \;\; \omega_0 = \frac{2\pi}{T} \text{ with } c_k \; = \; \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \tilde{f}(t) e^{-jk\omega_0 t} dt$$

If $F(\omega_k) = Tc_k$ with $\omega_k = k\omega_0 = \frac{2k\pi}{T}$ and $\Delta\omega = \omega_0 = \frac{2\pi}{T}$, we have

$$\begin{aligned} \tilde{f}(t) &= \sum_{k=-\infty}^{\infty} \frac{F(\omega_k)}{T} \, e^{j\omega_k t} = \sum_{k=-\infty}^{\infty} F(\omega_k) \, e^{j\omega_k t} \, \frac{\Delta\omega}{2\pi} \\ F(\omega_k) &= \int_{-\frac{T}{2}}^{\frac{T}{2}} \tilde{f}(t) e^{-j\omega_k t} dt \end{aligned}$$

When $T \to \infty$, $\omega_k \to \omega$, $\Delta\omega \to d\omega$, $\tilde{f}(t) \to f(t)$, the **sum** becomes an **integral**, the **spectral lines** get **closer** and, in the **limit**, a **continuous spectrum** is obtained.

Fourier transform

Fourier transform

The **Fourier transform** of a signal $f(t)$[67] is

$$\boxed{F(\omega) \;=\; \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt = \mathcal{F}[f(t)]}$$

The **inverse Fourier transform** is

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

Existence of the Fourier transform

The **existence conditions** of the Fourier transform are the **Dirichlet conditions**.

[67] that is not necessarily periodic

Fourier transform

F.Y.I.

Fourier transform

The **Fourier transform** of a signal $f(t)$[68] is

$$\boxed{F(f) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi ft}\, dt = \mathcal{F}[f(t)]}$$

The **inverse Fourier transform** is

$$\boxed{f(t) = \int_{-\infty}^{\infty} F(f)\, e^{j2\pi ft}\, df = \mathcal{F}^{-1}[F(f)]}$$

Existence of the Fourier transform

The **existence conditions** of the Fourier transform are the **Dirichlet conditions**.

[68] that is not necessarily periodic

Fourier transform

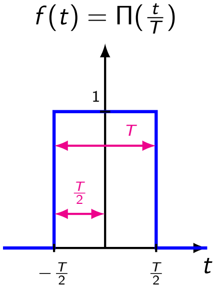
F.Y.I.

Interpretation

The **Fourier transform** of a signal $f(t)$ is the **projection** of $f(t)$ on a **continuous complex exponential basis**, i.e.

$$F(\omega) = \mathcal{F}[f(t)] = < f, e^{j\omega t} >$$

Fourier transform: example[69]



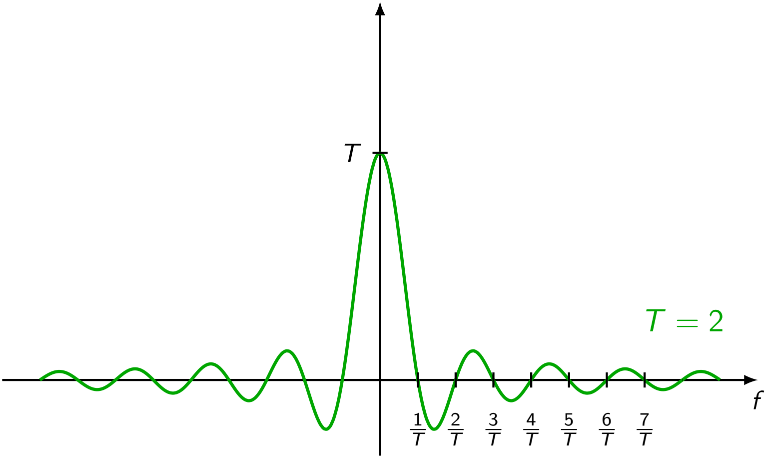
Given the **definition** of the **Fourier transform**, we obtain

$$
\begin{aligned}
F(\omega) &= \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt = \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{-j\omega t}dt \\
&= \frac{1}{-j\omega}\left[e^{-j\omega t}\right]_{-\frac{T}{2}}^{\frac{T}{2}} \\
&= \frac{1}{-j\omega}\left(e^{-j\omega\frac{T}{2}} - e^{j\omega\frac{T}{2}}\right) = \frac{1}{-j\omega}\left(-2j\sin({}^{\omega T}\!/_{2})\right) \\
&= \frac{2}{\omega}\sin({}^{\omega T}\!/_{2}) \\
&= T\,\frac{\sin(\pi f T)}{\pi f T} = T\,\mathrm{sinc}(fT)
\end{aligned}
$$

[69]The **cardinal sine** is defined as $\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$.

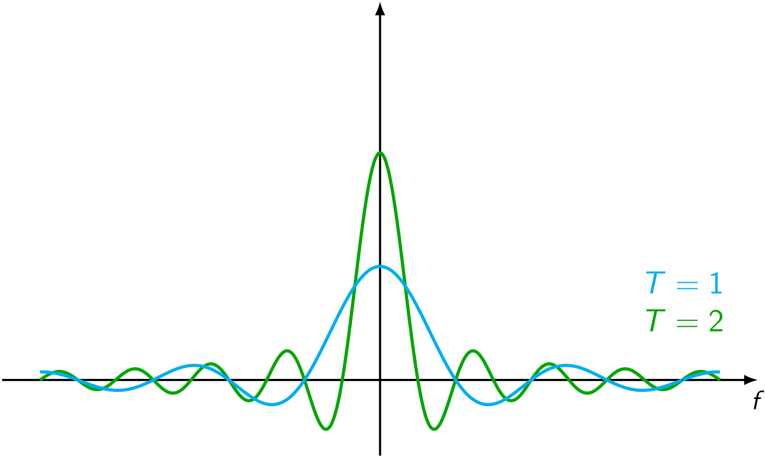
Cardinal sine

$$F(f) = T\left(\frac{\sin(\pi f T)}{\pi f T}\right)$$



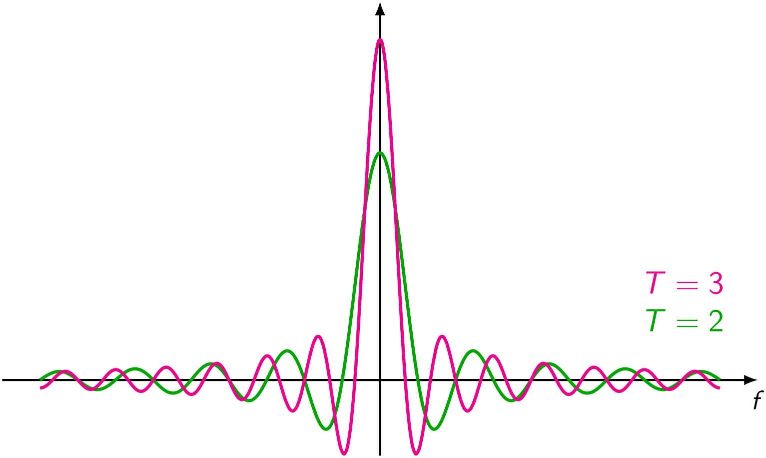
Cardinal sine

$$F(f) = T\left(\frac{\sin(\pi f T)}{\pi f T}\right)$$



Cardinal sine

$$F(f) = T\left(\frac{\sin(\pi f T)}{\pi f T}\right)$$



Fourier transform from Laplace transform

Fourier transforms from Laplace transform

If the **region of convergence** (ROC) of $F(s) = \mathcal{L}[f(t)]$ **contains** the $j\omega$-axis, so that $F(s)$ can be defined for $s = j\omega$, then

$$\mathcal{F}[f(t)] = \mathcal{L}[f(t)]|_{s=j\omega} = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt$$

$$\boxed{\mathcal{F}[f(t)] = F(s)|_{s=j\omega}}$$

Do **not** generalise this property when the ROC does **not** contain the imaginary axis, i.e. this property **can not** be used to compute the Fourier transform of a **step signal**, **sinusoidal signals** or any **periodic** signal[70].

Consider the **Laplace** transform for **transient and steady-state** behaviour, and the **Fourier** transform if only **steady-state** behavior is studied.

[70] For these signals, the Dirichlet conditions are **not** satisfied.

Fourier transform from Laplace transform

Laplace transform

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

$s = j\omega$

$\boxed{\sigma = 0 \text{ in ROC}}$

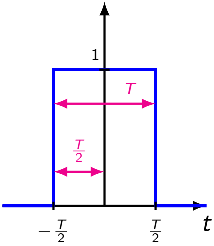
Fourier transform

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

Fourier transform from Laplace transform: example

Note that

$f(t) = \Pi(\frac{t}{T})$



$$
\begin{aligned}
f(t) &= u(t + {}^{T}\!/_{2}) - u(t - {}^{T}\!/_{2}) \text{ and} \\
F(s) &= \frac{1}{s}\left(e^{\frac{T}{2}s} - e^{-\frac{T}{2}s}\right)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{F}[f(t)] &= F(s)|_{s=j\omega} \\
&= \frac{1}{j\omega}\left(e^{\frac{T}{2}j\omega} - e^{-\frac{T}{2}j\omega}\right) \\
&= \frac{2}{\omega}\sin({}^{\omega T}\!/_{2}) \\
&= T\,\frac{\sin(\pi f T)}{\pi f T} = T\,\mathrm{sinc}(fT)
\end{aligned}
$$

Note that the **ROC** of $F(s)$ is the **whole** s-plane as $f(t)$ has **finite support**. The ROC therefore **includes** the $j\omega$-axis !

Fourier transform: frequency shift

Frequency shift

If $F(\omega) = \mathcal{F}[f(t)]$ is the **Fourier transform** of $f(t)$, then we have the **pair**

$$f(t)\, e^{j\omega_0 t} \overset{\mathcal{F}}{\underset{\mathcal{F}^{-1}}{\Longleftrightarrow}} F(\omega - \omega_0)$$

$$\begin{aligned}
\mathcal{F}[f(t)\, e^{j\omega_0 t}] &= \int_{-\infty}^{\infty} f(t)\, e^{j\omega_0 t}\, e^{-j\omega t} dt \\
&= \int_{-\infty}^{\infty} f(t)\, e^{-j(\omega - \omega_0) t} dt \\
&= F(\omega - \omega_0)
\end{aligned}$$

Fourier transform: amplitude modulation

Amplitude modulation consists in **multiplying** an incoming message $f(t)$ by a **sinusoidal signal** of frequency higher than the maximum frequency of the incoming signal. The **modulated** signal is $f(t)\cos(\omega_0 t)$.

Amplitude modulation

If $F(\omega) = \mathcal{F}[f(t)]$ is the **Fourier transform** of $f(t)$, then we have the **pair**

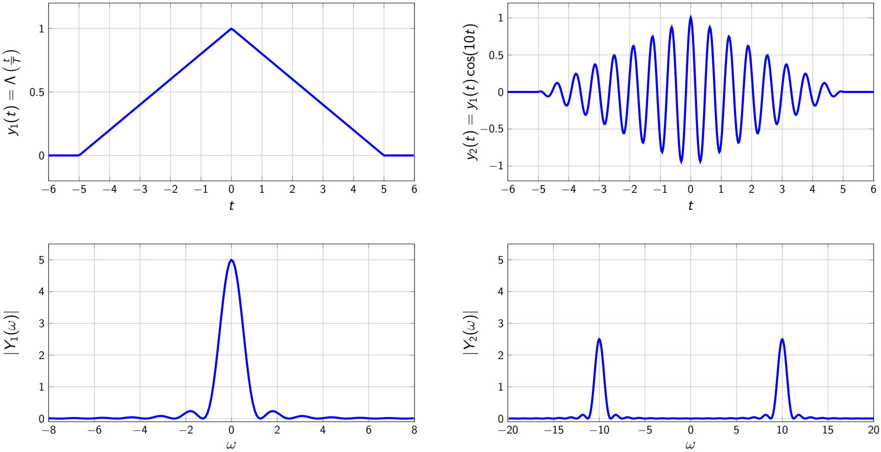
$$\boxed{f(t)\cos(\omega_0 t) \overset{\mathcal{F}}{\underset{\mathcal{F}^{-1}}{\Longleftrightarrow}} 0.5\,(F(\omega-\omega_0) + F(\omega+\omega_0))}$$

That is, the transform of the **modulated signal** is $F(\omega)$ **shifted** to frequencies ω_0 and $-\omega_0$, and **multiplied** by 0.5.

The Fourier pair is easily obtained **using**

$$f(t)\cos(\omega_0 t) = 0.5\, f(t)\,(e^{j\omega_0 t} + e^{-j\omega_0 t}).$$

Amplitude modulation



Applications of amplitude modulation

- **Telecommunications**:
  - **Acoustic** signals are **audible** up to 20kHz.
  - When emitting signals with an **antenna**, the **length** of the antenna is related to the **quarter wavelength**.
  - The **wavelength**[71] is give by $\lambda = \frac{c}{f} = \frac{3 \times 10^8}{f}$ meters.
  - To emit signals with a frequency content up to 30kHz, the wavelength is 10km and the length of the antenna is 2.5km.
  - Amplitude modulation is an **important application** of the Fourier transform, as it allows us to change the original frequencies of a message to **much higher** frequencies, making it possible to transmit the signal over the **airwaves**[72].

---

[71] **Spatial period** of a periodic wave, i.e. the distance over which the wave's shape repeats.

[72] The **amplitude modulated carrier frequencies**, i.e. **AM radio**, are in the **frequency range** 535-1605 kHz.

Applications of amplitude modulation

- **Condition monitoring**: some **mechanical faults** (gearbox, bearings, etc.) introduce **amplitude** (or phase) **modulation** in vibration or current signals.

Time-frequency duality

Time-frequency duality (1)

To the **Fourier transform pair**

$$f(t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} F(\omega) = \mathcal{F}[f(t)]$$

corresponds the following **dual Fourier transform pair**

$$F(t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} 2\pi f(-\omega).$$

This duality property allows us to obtain the **new Fourier transform pairs** from Fourier transform pairs that are **already available** and that would be difficult to obtain directly.

It is thus **one more method** to obtain the Fourier transform, besides the Laplace transform and the integral definition of the Fourier transform.

Time-frequency duality

Time-frequency duality (2)

The **inverse Fourier transform** is obtained using

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

Replacing t by $-\omega$ and multiplying by 2π, we **obtain**

$$\begin{aligned} 2\pi f(-\omega) &= \int_{-\infty}^{\infty} F(\rho)e^{-j\rho\omega}d\rho, \\ &= \int_{-\infty}^{\infty} F(t)e^{-j\omega t}dt, \\ &= \mathcal{F}[F(t)]. \end{aligned}$$

Time-frequency duality: constant signal

Fourier transform of a constant signal

To the **Fourier transform pair**[73]

$$A\,\delta(t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} A$$

corresponds the **dual Fourier transform pair**

$$A \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} 2\pi A\,\delta(-\omega) = 2\pi A\,\delta(\omega).$$

We have the following **interpretation**:

- A **Dirac impulse** contains **all frequencies** which makes it **impossible** to **create** in **practice**.
- A **constant signal only** contains the **zero frequency** !

[73] This pair follows from the property $\mathcal{F}[f(t)] = F(s)|_{s=j\omega}$

Time-frequency duality: cardinal sine

Fourier transform of a cardinal sine

To the **Fourier transform pair**

$$\boxed{\Pi({}^{t}/_{T}) = u(t + {}^{T}/_{2}) - u(t - {}^{T}/_{2}) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} \frac{2}{\omega}\sin({}^{\omega T}/_{2})}$$

corresponds the **dual Fourier transform pair** ($\frac{T}{2}$ replaced by ω_0)

$$\frac{2}{t}\sin(\omega_0 t) \quad \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} \quad 2\pi\,\Pi(\,-{}^{\omega}/_{2\omega_0}) = 2\pi\,\Pi({}^{\omega}/_{2\omega_0}).$$

Therefore

$$\boxed{\frac{\sin(\omega_0 t)}{\pi t} \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} u(\omega + \omega_0) - u(\omega - \omega_0) = \Pi({}^{\omega}/_{2\omega_0})}$$

Time-frequency duality: Fourier transform of a cosine

To the **Fourier transform pair**

$$\delta(t+\tau)+\delta(t-\tau) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} e^{j\omega\tau}+e^{-j\omega\tau} = 2\cos(\omega\tau)$$

corresponds the **dual Fourier transform pair**

$$2\cos(\tau t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} 2\pi\left(\delta(-\omega+\tau)+\delta(-\omega-\tau)\right) = 2\pi\left(\delta(\omega-\tau)+\delta(\omega+\tau)\right)$$

Remplacing τ by ω_0, the **dual Fourier pair** becomes

$$\boxed{\cos(\omega_0 t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} \frac{2\pi}{2}\left(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right)}$$

The Fourier transform **can not** be computed using

- the integral definition since this signal is **not absolutely integrable**,
- the Laplace transform: the $j\omega$-axis is **not included in the ROC**.

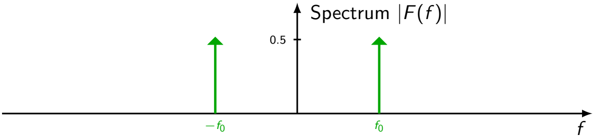
Fourier transform of a cosine

We have

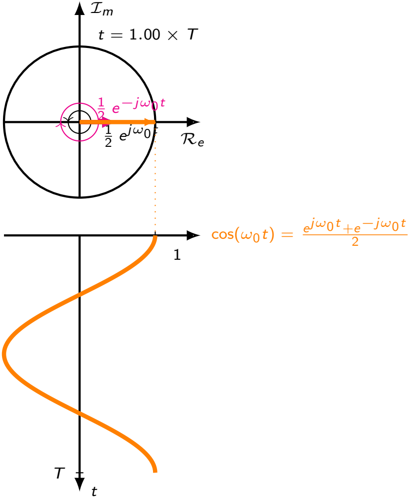
$$\mathcal{F}\left[\cos(\omega_0 t)\right] = \frac{2\pi}{2}\,\left(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)\right)$$

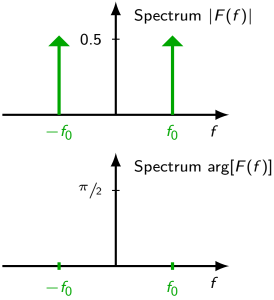
or

$$\boxed{\mathcal{F}\left[\cos(f_0 t)\right] = \frac{1}{2}\,\left(\delta(f - f_0) + \delta(f + f_0)\right)}$$



Cosine: vector sum and spectrum





Time-frequency duality: Fourier transform of a sine

To the **Fourier transform pair**

$$\delta(t+\tau) - \delta(t-\tau) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} e^{j\omega\tau} - e^{-j\omega\tau} = 2j\sin(\omega\tau)$$

corresponds the **dual Fourier transform pair**

$$2j\sin(\tau t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} 2\pi\left(\delta(-\omega+\tau) - \delta(-\omega-\tau)\right) = 2\pi\left(\delta(\omega-\tau) - \delta(\omega+\tau)\right)$$

Remplacing τ by ω_0, the **dual Fourier pair** becomes

$$\boxed{\sin(\omega_0 t) \underset{\mathcal{F}^{-1}}{\overset{\mathcal{F}}{\Longleftrightarrow}} \frac{2\pi}{2j}\left(\delta(\omega-\omega_0) - \delta(\omega+\omega_0)\right)}$$

The Fourier transform **can not** be computed using

- the integral definition since this signal is **not absolutely integrable**,
- the Laplace transform: the $j\omega$-axis is **not included in the ROC**.

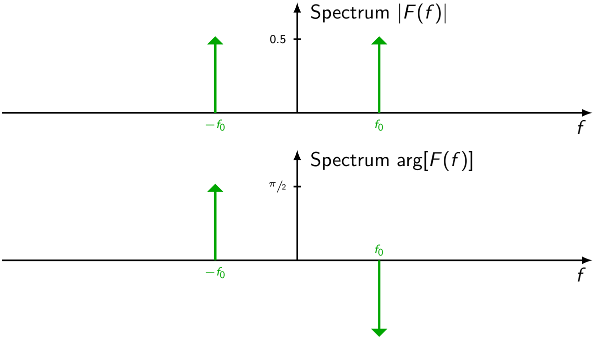
Fourier transform of a sine

We have

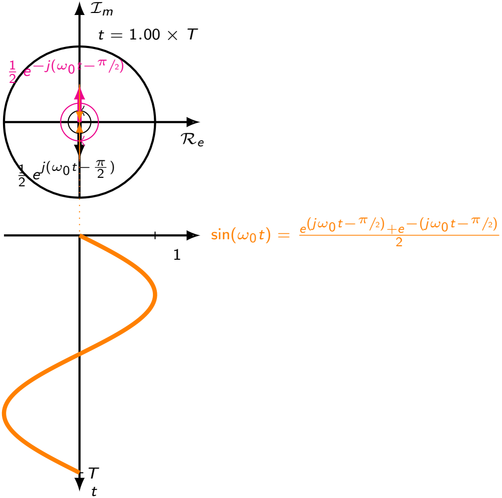
$$\mathcal{F}\left[\sin(\omega_0 t)\right] = \frac{2\pi}{2j}\,\left(\delta(\omega - \omega_0) - \delta(\omega + \omega_0)\right)$$

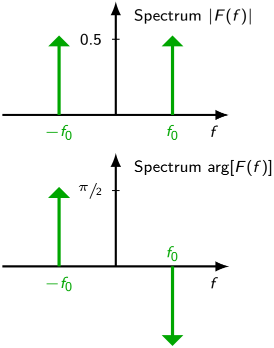
or

$$\boxed{\mathcal{F}\left[\sin(f_0 t)\right] = \frac{1}{2j}\,\left(\delta(f - f_0) - \delta(f + f_0)\right)}$$



Sine: vector sum and spectrum





Fourier transform of periodic signals

By applying the **frequency-shifting property** to compute the **Fourier transform** of **periodic signals**, we are able to **unify** the Fourier representation of **aperiodic** as well as **periodic signals**.

Fourier transform of periodic signals

For a **periodic** signal $f(t)$ of period T, we have the **Fourier pair**

$$\boxed{f(t) = \sum_k c_k \, e^{jk\omega_0 t} \overset{\mathcal{F}}{\underset{\mathcal{F}^{-1}}{\Longleftrightarrow}} F(\omega) = 2\pi \sum_k c_k \, \delta(\omega - k\omega_0)}$$

obtained by **representing** $f(t)$ by its **Fourier series**.

Since a **periodic** signal $f(t)$ is **not absolutely integrable**, its Fourier transform **cannot** be computed using the **integral formula**. The **Fourier series** is used to **circumvent** this problem.

Parseval's energy conservation

F.Y.I.

Parseval's energy conservation

For a **finite-energy** signal $f(t)$ with Fourier transform $F(\omega)$, its **energy** is **conserved** when going from the **time** to the **frequency** domain, or

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega = \int_{-\infty}^{\infty} |F(f)|^2 df.$$

Thus, $|F(\omega)|^2$ is an **energy density** indicating the amount of energy at each of the frequencies ω.

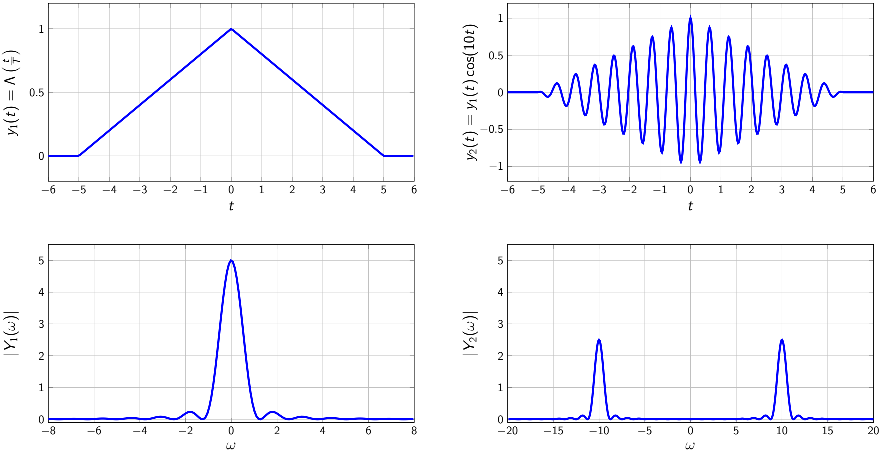
The plot $|F(\omega)|^2$ versus ω is called the **energy spectrum** of $f(t)$, and it displays how the **energy** of the signal is **distributed** over **frequency**.

Amplitude modulation

```
function X = Afourier(x,t0) % Approximate Fourier transform

syms t w
X = int(x*exp(-1i*w*t), t,-t0,t0);
```

```
% Signals and Systems using Matlab by Chaparro: example 5.7

clear all; close all;
syms t w
m=heaviside(t+5)-heaviside(t);
m1=heaviside(t)-heaviside(t-5);
x2=(t+5)*m+m1*(-t+5);x2=x2/5;
x=x2*exp(-1i*10*t)/2; y=x2*exp(+1i*10*t)/2;
Y=Afourier(y,5); X=Afourier(x,5);
X2=Afourier(x2,5);

figure(1)
subplot(221); fplot(matlabFunction(x2), [-6,6]);
grid; axis([-6 6 -0.2 1.2]); xlabel('t'); ylabel('x2(t)');
subplot(222); fplot(matlabFunction(x+y), [-6,6]);
grid; axis([-6 6 -1.2 1.2]); xlabel('t'); ylabel('y2(t)=x2(t)cos(10t)');
subplot(223); fplot(matlabFunction(abs(X2)), [-7.9,8],1000);
grid; axis([-8 8 -0.1 5.5]); xlabel('\omega'); ylabel('|X_2(\omega)|');
subplot(224); fplot(matlabFunction(abs(X)+abs(Y)), [-19.9,20])
grid; axis([-20 20 -0.1 5.5]); xlabel('\omega'); ylabel('|Y_2(\omega)|');
```

Amplitude modulation



Fourier transform pairs	
$f(t)$	$\mathcal{F}[f(t)]$
$A\,\delta(t)$	A
A	$2\pi A\,\delta(\omega)$
$t^n e^{at} u(t),\ a > 0$	$\dfrac{n!}{(j\omega - a)^{n+1}}$
$u(t+T) - u(t-T)$	$2\,\dfrac{\sin(\omega T)}{\omega}$
$\dfrac{\sin(\omega_0 t)}{\pi t}$	$u(\omega + \omega_0) - u(\omega - \omega_0)$

Fourier transform pairs	
$f(t)$	$\mathcal{F}[f(t)]$
$\cos(\omega_0 t)$	$\pi\ (\delta(\omega+\omega_0)+\delta(\omega-\omega_0))$
$\sin(\omega_0 t)$	$j\pi\ (\delta(\omega+\omega_0)-\delta(\omega-\omega_0))$
$f(t)=\sum_k c_k\, e^{jk\omega_0 t}$	$F(\omega)=2\pi\sum_k c_k\,\delta(\omega-k\omega_0)$

Basic properties of the Fourier transform		
Properties	$f(t)$	$F(\omega)$
Signals and constants	$\alpha f(t),\ \beta g(t)$	$\alpha F(\omega),\ \beta G(\omega)$
Linearity	$\alpha f(t) + \beta g(t)$	$\alpha F(\omega) + \beta G(\omega)$
Time shift	$f(t-\tau)$	$e^{-j\tau\omega}\, F(\omega)$
Frequency shift	$e^{j\omega_0 t}\, f(t)$	$F(\omega - \omega_0)$
Modulation	$f(t)\cos(\omega_0 t)$	$0.5\,(F(\omega-\omega_0) + F(\omega+\omega_0))$

Basic properties of the Fourier transform

Properties	$f(t)$	$F(\omega)$
Time convolution	$f(t) * g(t)$	$F(\omega)G(\omega)$
Windowing / multiplication	$f(t)\,\omega(t)$	$\frac{1}{2\pi}[F * W](\omega)$
Time differentiation	$\frac{df(t)}{dt}$	$j\omega F(\omega)$
Integration	$\int_{-\infty}^{t} f(\bar{t})d\bar{t}$	$\frac{F(\omega)}{\omega} + \pi F(0)\delta(\omega)$
Expansion / contraction	$f(at), (a \neq 0)$	$\frac{1}{a} F\left(\frac{\omega}{a}\right)$

Basic properties of the Fourier transform

Properties	$f(t)$	$F(\omega)$
Parseval	$E = \int_{-\infty}^{\infty} \lvert f(t)\rvert^2 dt$	$E = \frac{1}{2\pi}\int_{-\infty}^{\infty} \lvert F(\omega)\rvert^2 d\omega$
Periodic signals	$\sum_k c_k\, e^{jk\omega_0 t}$	$2\pi \sum_k c_k\, \delta(\omega - k\omega_0)$
Symmetry	real-valued $f(t)$	$\begin{cases} \lvert F(\omega)\rvert = \lvert F(-\omega)\rvert, \\ \arg[F(\omega)] = -\arg[F(-\omega)] \end{cases}$
Cosine	real-valued even $f(t)$	$\int_{-\infty}^{\infty} f(t)\cos(\omega t)dt$, real
Sine	real-valued odd $f(t)$	$-j\int_{-\infty}^{\infty} f(t)\sin(\omega t)dt$, imaginary

6. Sampling theory

Ideal impulse sampling
Frequency folding
Analog-to-Digital Conversion
Digital-to-Analog Conversion
ADC followed by DAC

Introduction

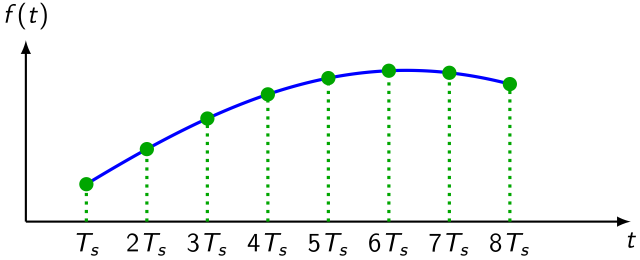
- Since many of the signals found in applications such as communications and control are analog, if we wish to process these signals with a computer it is necessary to **sample**, **quantize**, and **code** them to obtain **digital signals**.
- It is therefore necessary to understand the **bridge** between **analog** and **digital** signals and systems.
- The device that samples, quantizes, and codes an analog signal is called an **Analog-to-Digital Converter (ADC)**, while the device that converts digital signals into analog signals is called a **Digital-to-Analog Converter (DAC)**.
- These devices are far from ideal and thus some **practical aspects of sampling** and **reconstruction** need to be considered: loss or information, **aliasing or frequency folding**, quantisation error, etc.

Uniform sampling

Uniform sampling

Sampling an **analog continuous-time** signal $f(t)$ yields **samples** of the system representing the **amplitude** of the signal at the **sampling instants**.

The samples $f(kT_s)$ are taken on a **regular basis** with a **sampling period** T_s.

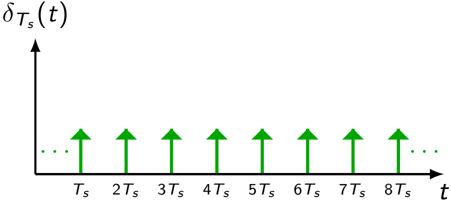


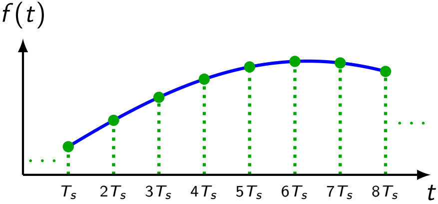
Ideal impulse sampling

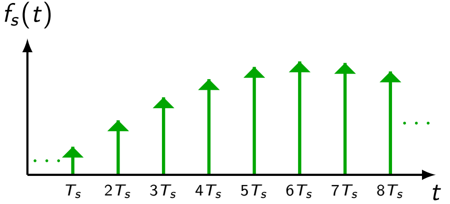
The sampling process is **ideal** if the samples are considered to be **instantaneous** values **precisely** at the **sample instants**.

Ideal sampling can be **modeled** using the **Dirac comb sampling function**.

$$\begin{aligned}\delta_{T_s}(t) &= \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \\ f_s(t) &= f(t)\,\delta_{T_s}(t) \\ &= \sum_{k=-\infty}^{\infty} f(kT_s)\,\delta(t - kT_s)\end{aligned}$$







Fourier series of a Dirac comb

The **Dirac comb sampling function** is **periodic** with period T_s.
It can therefore be written as a **Fourier series**:

$$\begin{aligned}
\delta_{T_s}(t) &= \sum_{k=-\infty}^{\infty} c_k \, e^{jk\omega_s t}, \; \omega_s = \frac{2\pi}{T_s} \\
c_k &= \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta_{T_s}(t) \, e^{-jk\omega_s t} \, dt \\
&= \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \, e^{-jk\omega_s t} \, dt \\
&= \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta(t) \, e^{-jk\omega_s t} \, dt = \frac{1}{T_s}
\end{aligned}$$

Fourier transform of a Dirac comb

F.Y.I.

The **Dirac comb sampling function** has **Fourier series**

$$\boxed{\delta_{T_s}(t) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} e^{jk\omega_s t}, \ \omega_s = \frac{2\pi}{T_s}}$$

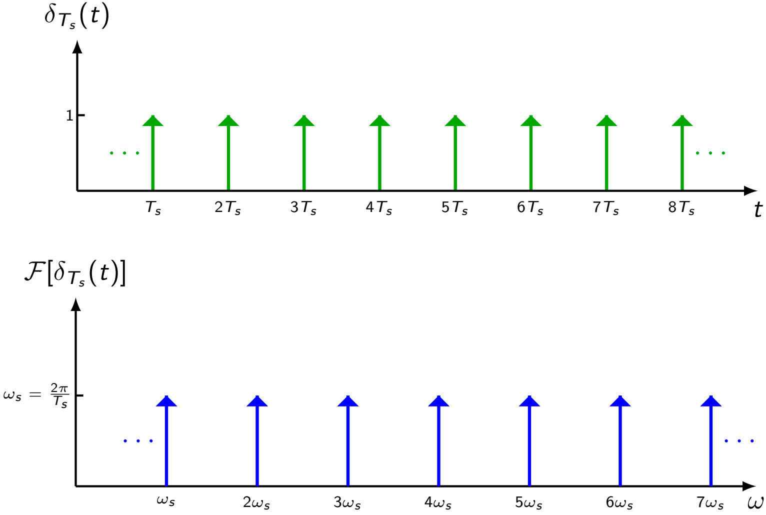
Therefore

$$\begin{aligned} \mathcal{F}[\delta_{T_s}(t)] &= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \mathcal{F}[e^{jk\omega_s t}] \\ &= \frac{2\pi}{T_s}\sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s) \end{aligned}$$

The **Fourier transform** of the **Dirac comb** is a **superposition** of **shifted** pulse $\delta(\omega - k\omega_s)$ multiplied by $2\pi/T_s = \omega_s$.

Dirac comb and its Fourier transform

F.Y.I.



Fourier transform of a sampled signal

The **Dirac comb sampling function** has **Fourier series**

$$\boxed{\delta_{T_s}(t) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} e^{jk\omega_s t},\ \omega_s = \frac{2\pi}{T_s}}$$

We have

$$f_s(t) = f(t)\,\delta_{T_s}(t).$$

Therefore

$$\begin{aligned} F_s(\omega) &= \mathcal{F}[f(t)\,\delta_{T_s}(t)] \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \mathcal{F}[f(t)\,e^{jk\omega_s t}] \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} F(\omega - k\omega_s) \end{aligned}$$

The **Fourier transform** of the **sampled signal** is a **superposition** of **shifted** analog spectra $F(\omega - k\omega_s)$ multiplied by $^1/_{T_s}$.

Uniform sampling and Fourier transform

Sampling a continuous-time $f(t)$ with a **uniform sampling period** T_s yields the signal

$$f_s(t) = \sum_{k=-\infty}^{\infty} f(kT_s)\,\delta(t - kT_s).$$

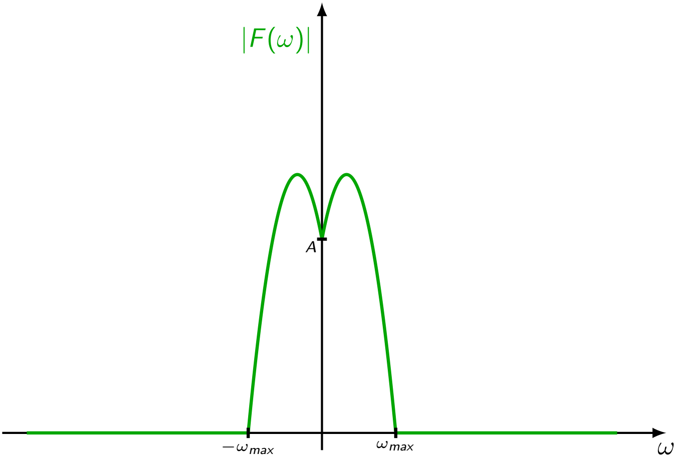
Sampling is equivalent to **modulating** the signal using the **sampling function**

$$\delta_{T_s}(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s).$$

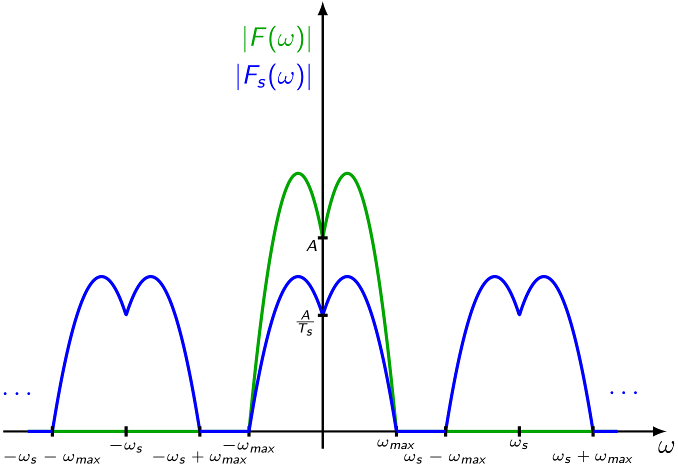
If $F(\omega)$ is the Fourier transform of $f(t)$, the **Fourier transform** of the **sampled signal** $f_s(t)$ is

$$\boxed{F_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} F(\omega - k\omega_s)}$$

Low-pass spectrum of finite support: Nyquist condition



Low-pass spectrum of finite support: Nyquist condition



No overlapping $\Longrightarrow$ **Nyquist condition** is **respected** !

Nyquist condition

Nyquist condition

If the initial signal $f(t)$ has a **low-pass spectrum** of **finite support**, i.e. $F(\omega) = 0$ pour $\omega > \omega_{max}$ where ω_{max} is the **maximum frequency** present in the signal[74], it is possible to **choose** ω_s so that the spectrum of the sampled signal consists of **shifted** but **non-overlapping** scaled versions of $F(\omega)$.

The **Nyquist condition** on ω_s is $\omega_s - \omega_{max} \geq \omega_{max}$, or

$$\omega_s \geq 2\,\omega_{max}.$$

This is sometimes written under the form

$$\boxed{\omega_N = \frac{\omega_s}{2} \geq \omega_{max}}$$

where ω_N is called the **Nyquist frequency**.

[74] Such a signal is called **band-limited**.

Frequency folding or aliasing

When the **Nyquist condition** is **not respected**, it is **impossible** to **distinguish** between a **frequency** ω and its **aliases** $\omega + k\,\omega_s$.

Consequence : Loss of information !

When ?

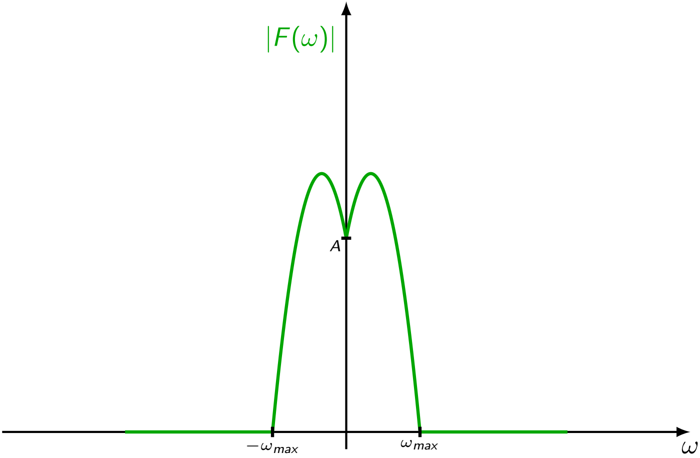
- The initial signal $f(t)$ has a spectrum of **unlimited support**.

  Solution: Pass the signal through an analog **low-pass anti-aliasing filter** to remove all frequency components above the Nyquist frequency.

- The initial signal $f(t)$ is band limited but the **Nyquist sampling condition is not respected**.

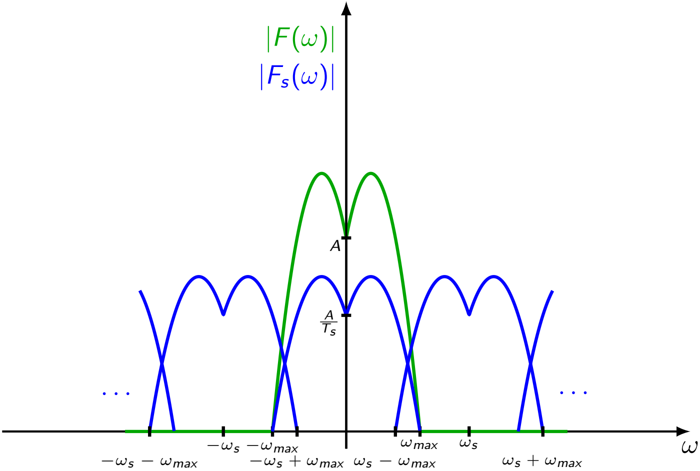
  Solution: **Increase** the sampling frequency.

Frequency folding or aliasing



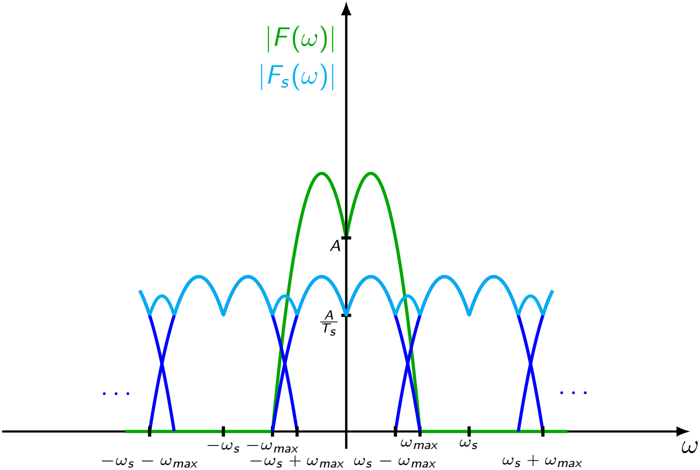
Nyquist condition is **not** respected !

Frequency folding or aliasing



Nyquist condition is **not** respected !

Frequency folding or aliasing

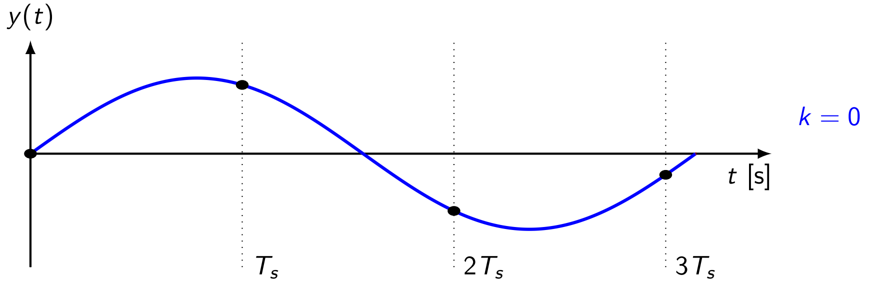


Nyquist condition is **not** respected !

Sampling issues

Several distinct signals can **coincide** on the **same** discrete samples !

Example: $y = \sin(t)$, $T_s = 2, \omega_s = \frac{2\pi}{T_s} = \pi > 2\,\omega_{max} = 2$.

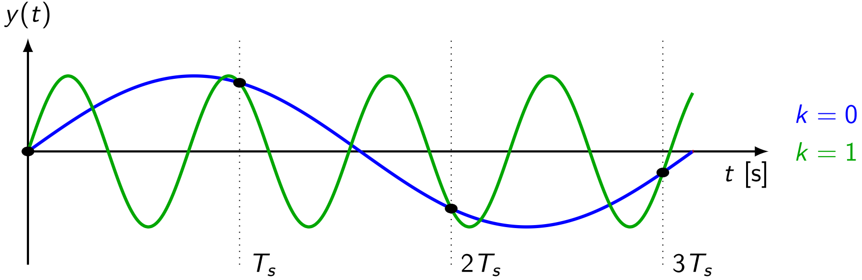


Sampling issues

Several distinct signals can **coincide** on the **same** discrete samples !

Example: $y = \sin(t)$, $T_s = 2, \omega_s = \frac{2\pi}{T_s} = \pi > 2\,\omega_{max} = 2$.

If the signals $y_k(t) = \sin\left((1 + k\,\omega_s)\,t\right)$ **pollute** the **base signal** $y(t)$, they are **identical**[75] to the **base signal** $y(t)$ at the **sample instants**.



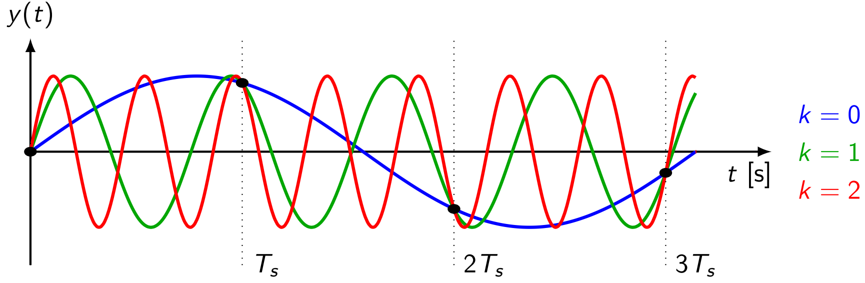
[75] The Nyquist condition is **not** verified anymore !

Sampling issues

Several distinct signals can **coincide** on the **same** discrete samples !

Example: $y = \sin(t)$, $T_s = 2, \omega_s = \frac{2\pi}{T_s} = \pi > 2\,\omega_{max} = 2$.

If the signals $y_k(t) = \sin\left((1 + k\,\omega_s)\,t\right)$ **pollute** the **base signal** $y(t)$, they are **identical**[75] to the **base signal** $y(t)$ at the **sample instants**.



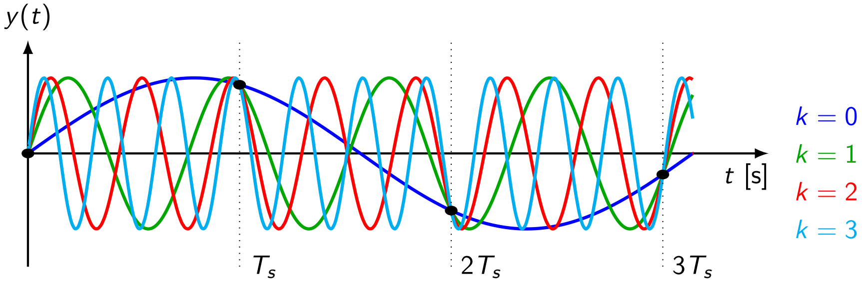
[75] The Nyquist condition is **not** verified anymore !

Sampling issues

Several distinct signals can **coincide** on the **same** discrete samples !

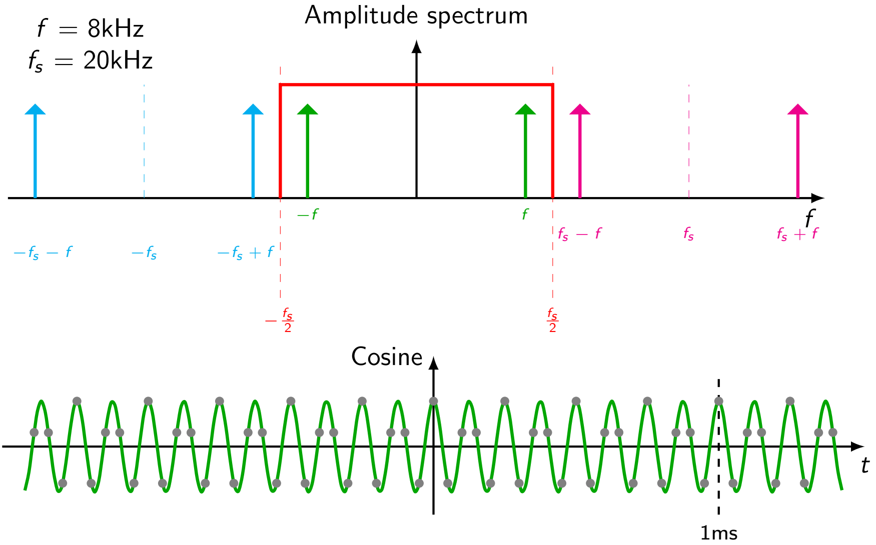
Example: $y = \sin(t)$, $T_s = 2, \omega_s = \frac{2\pi}{T_s} = \pi > 2\,\omega_{max} = 2$.

If the signals $y_k(t) = \sin\left((1 + k\,\omega_s)\,t\right)$ **pollute** the **base signal** $y(t)$, they are **identical**[75] to the **base signal** $y(t)$ at the **sample instants**.

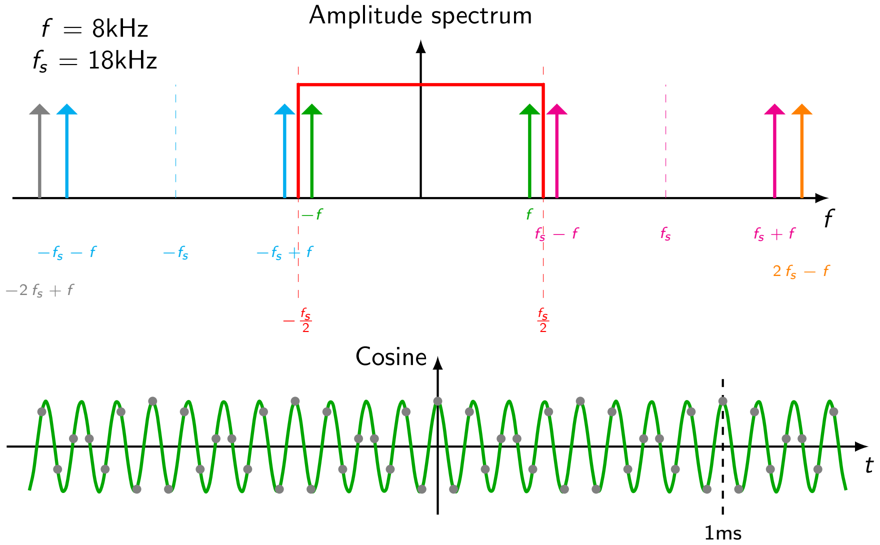


[75] The Nyquist condition is **not** verified anymore !

8kHz cosine sampled at 20kHz

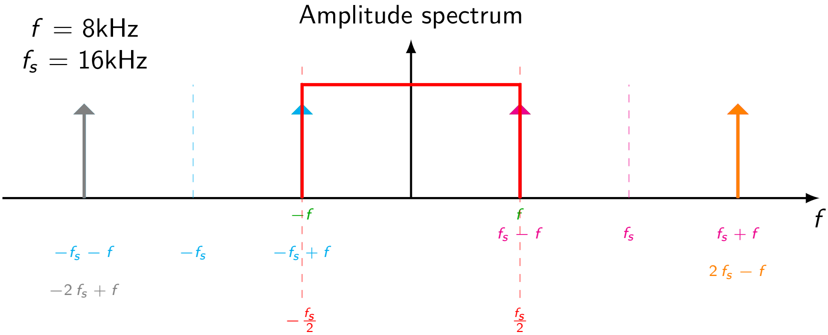


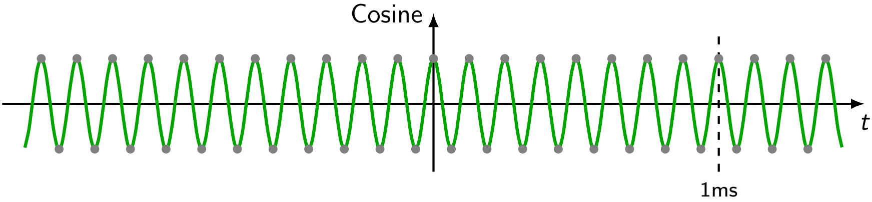
8kHz cosine sampled at 18kHz



8kHz cosine sampled at 16kHz

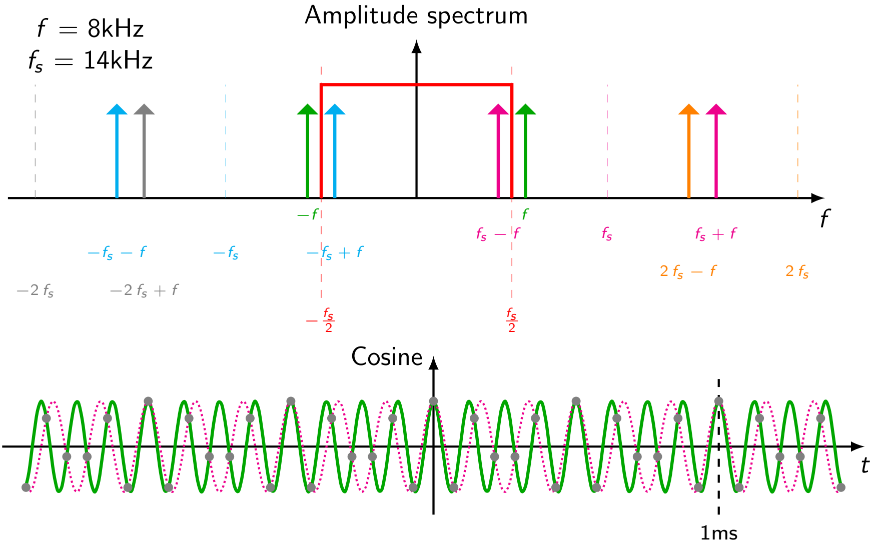
$f = 8\text{kHz}$
$f_s = 16\text{kHz}$





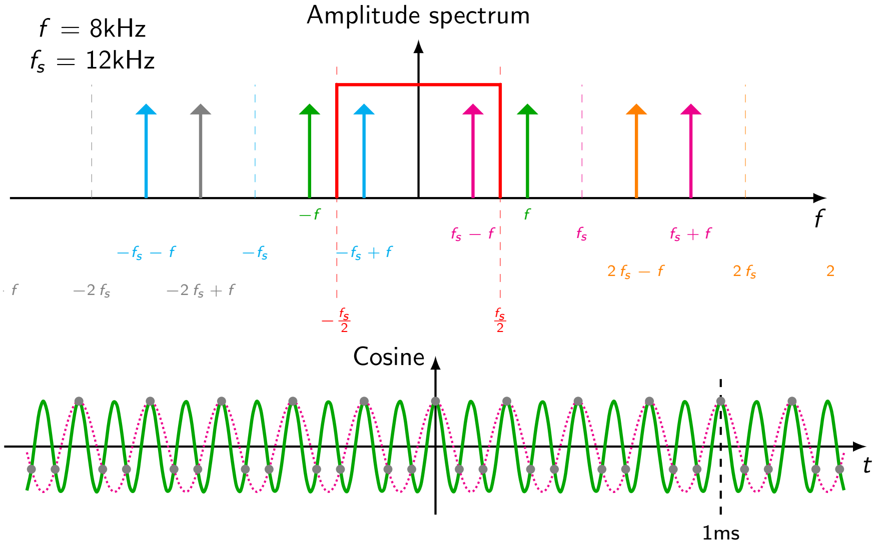
8kHz cosine sampled at 14kHz

$f = 8\text{kHz}$
$f_s = 14\text{kHz}$

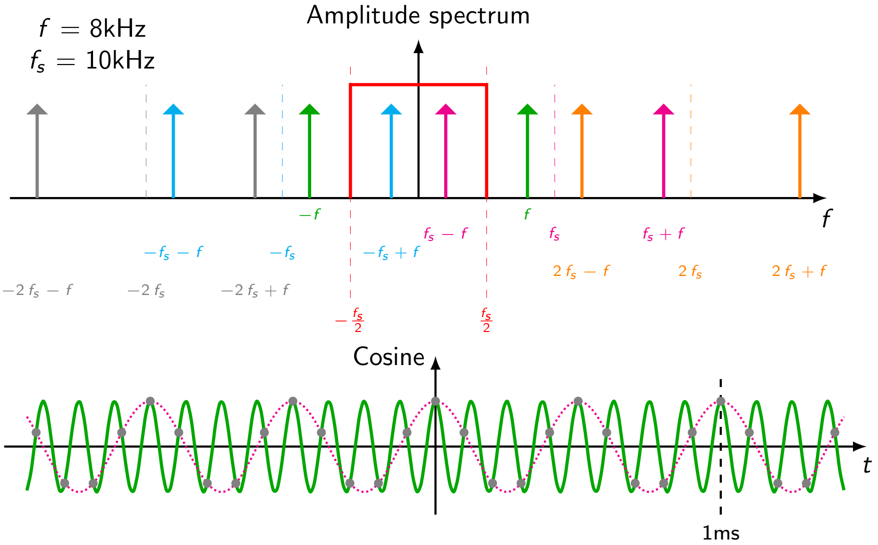


8kHz cosine sampled at 12kHz

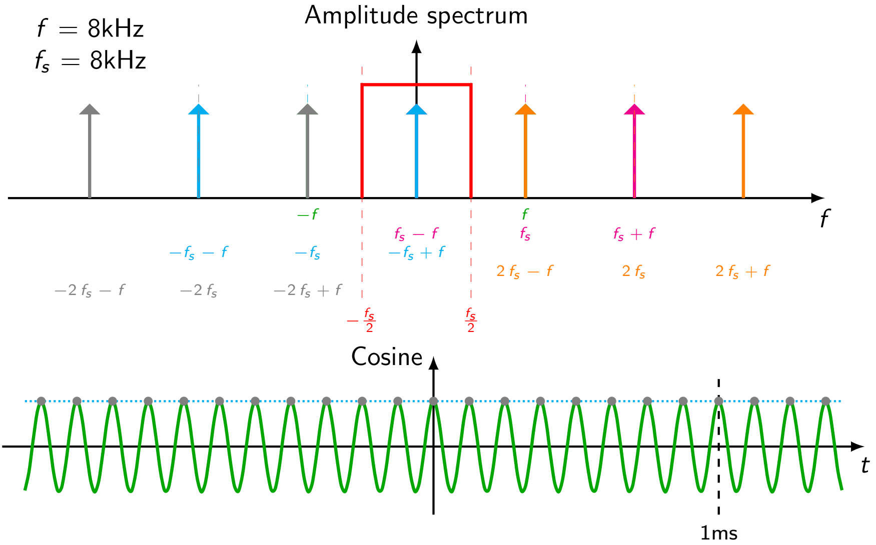
$f = 8\text{kHz}$
$f_s = 12\text{kHz}$



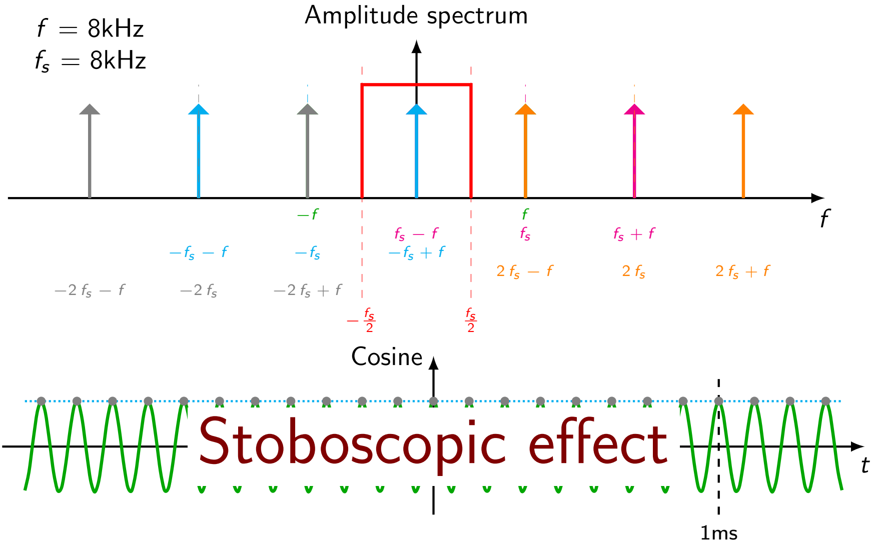
8kHz cosine sampled at 10kHz



8kHz cosine sampled at 8kHz

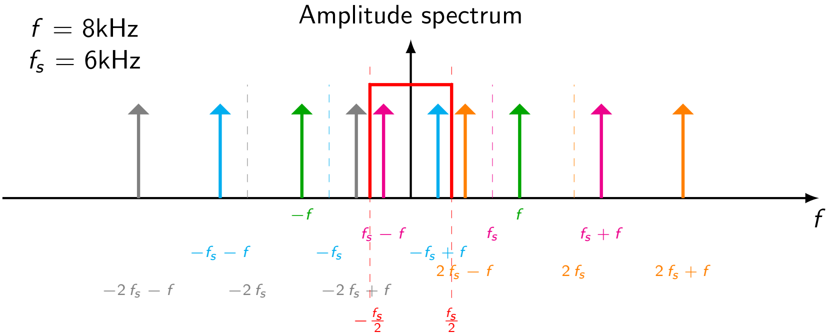


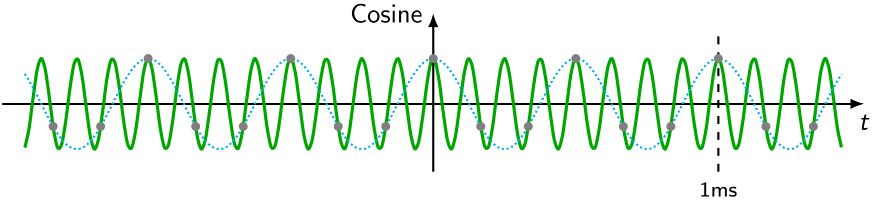
8kHz cosine sampled at 8kHz



8kHz cosine sampled at 6kHz

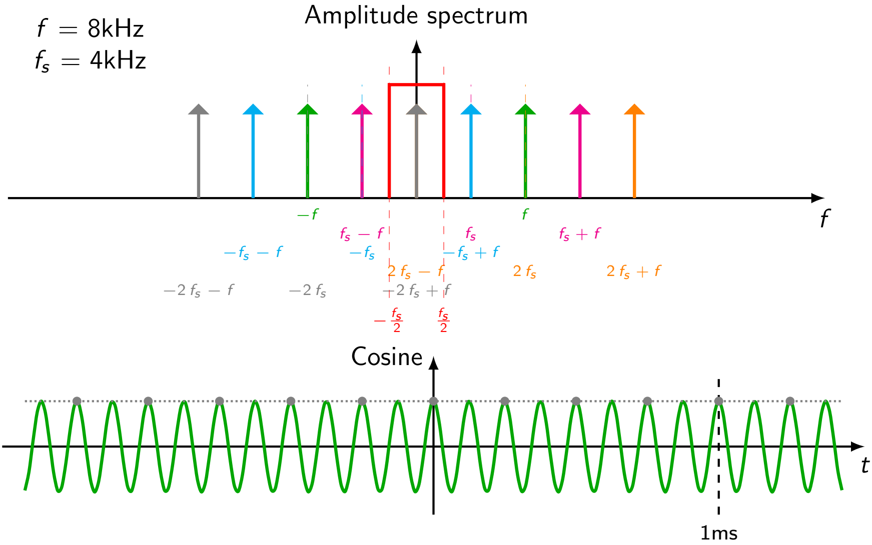
$f = 8\text{kHz}$
$f_s = 6\text{kHz}$





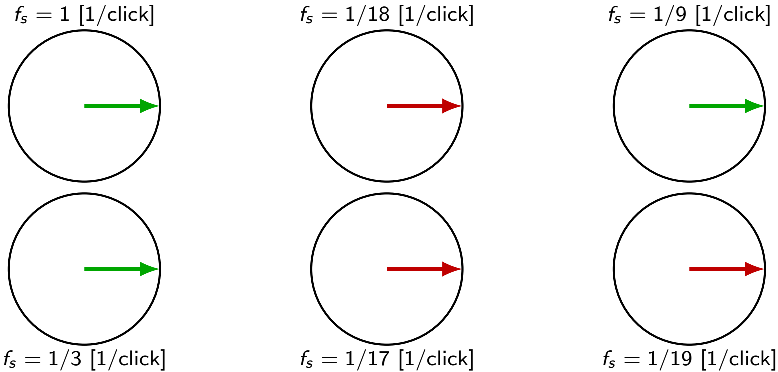
8kHz cosine sampled at 4kHz

$f = 8\text{kHz}$
$f_s = 4\text{kHz}$



Frequency folding: illustration of the stroboscopic effet

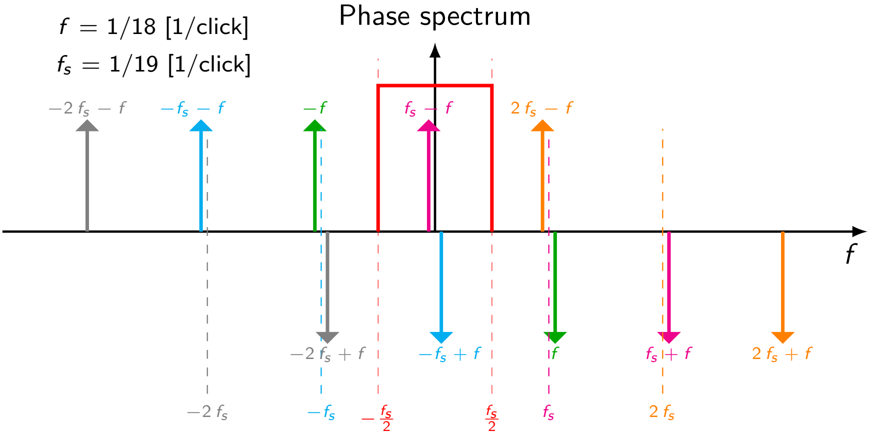
- Top left arrow moving at 20°/click. The period is $360°/(20°/\text{click}) = 18$ clicks, i.e. the **arrow rotation frequency** $f = 1/18$ [1/click].
- The **sampling frequency** f_s, i.e. the **flash frequency** of the strobe light, should **obey the Nyquist condition**, i.e. $f_s \geq 2\,f = 1/9$ [1/click] !



Frequency folding: illustration of the stroboscopic effet

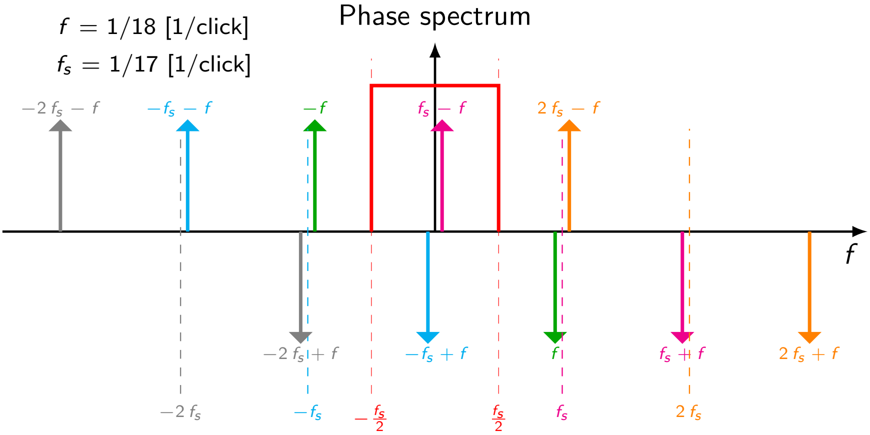
$f = 1/18$ [1/click]

$f_s = 1/19$ [1/click]



Positive sine of frequency 1/18 - 1/19 = 1/(18 × 19)

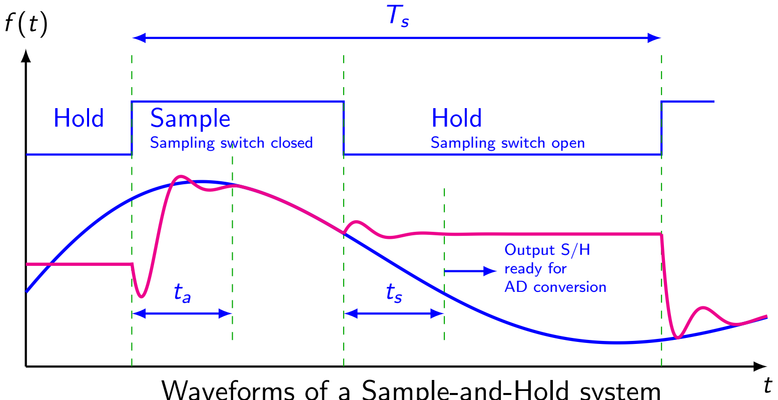
Frequency folding: illustration of the stroboscopic effet



Negative sine of frequency $1/17$ - $1/18 = 1/(17 \times 18)$

Practical aspects of sampling

The **Sample-and-Hold**[76] (S/H) system first **acquires** the signal then takes the sample and **holds** it long enough for **quantisation** and **coding** to be done.



Waveforms of a Sample-and-Hold system

[76] Echantillonnage-blocage

Practical aspects of sampling

Definitions

- **Acquisition time** (t_a): time to acquire the analog signal
- **Settling time** (t_s): time to settle to the final held value within an accuracy tolerance. After that time, the sample is ready for quantification and coding.

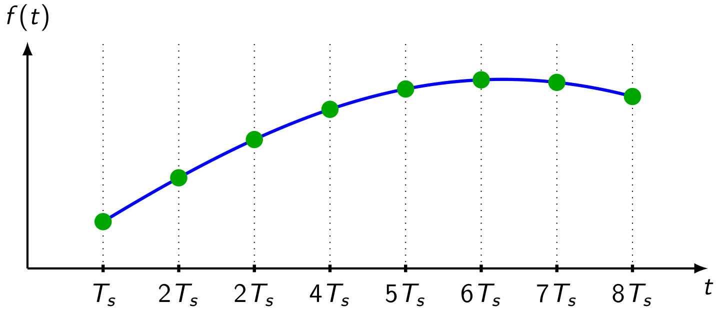
Other considerations

- **Aperture time**: time for the sampling switch to open.
- **Aperture jitter**[77]: variations on the aperture time due to clock variations and noise.

[77] Gigue d'échantillonnage

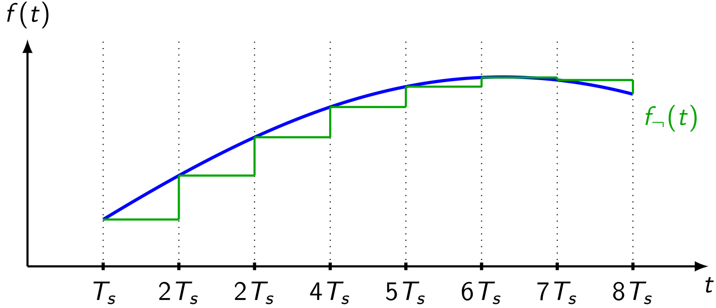
Practical aspects of sampling

Assuming **acquisition** is **instantaneous**, the signal resulting from the Sample-and-Hold has a **staircase** shape of $f_{\neg}(t)$.



Practical aspects of sampling

Assuming **acquisition** is **instantaneous**, the signal resulting from the Sample-and-Hold has a **staircase** shape of $f_\neg(t)$.



Quantisation

- **Quantisation** is the **second step** in the **digitalisation**[78] of signals.
- It allows **storage** and signal processing once the signal is **coded**.
- The role of quantisation is to affect a **finite resolution** value to a sample which amplitude has in **theory** an **infinite resolution**.
- Quantisation is **rounding the analog value** to the **closest digital value** according to a **quantisation grid**.

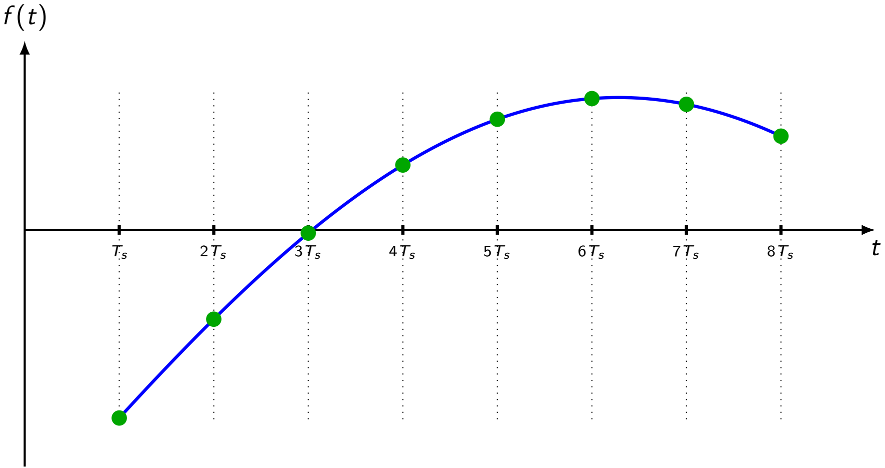
Quantisation

Quantising a sample is to take the one of the **fixed amplitude levels** that **best** represents the sample according to some **approximation scheme**. It is a **nonlinear** process.

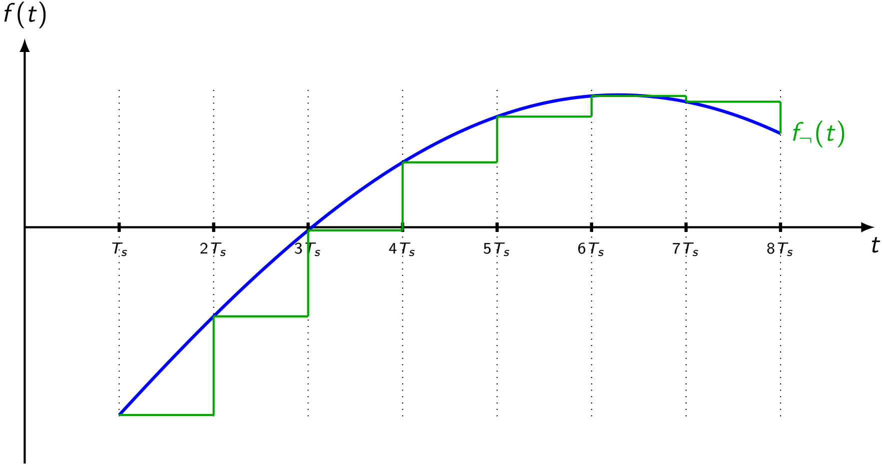
When the amplitude levels are **equidistant**, quantisation is **uniform**.

[78] Also known as digitisation

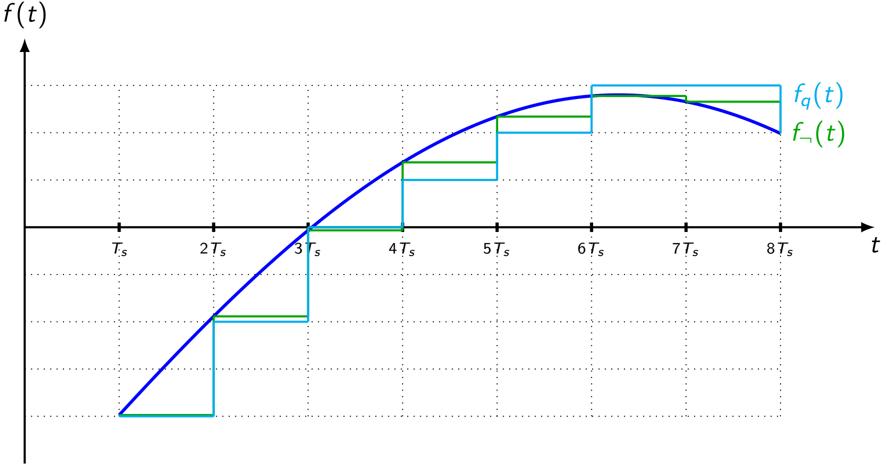
Quantisation



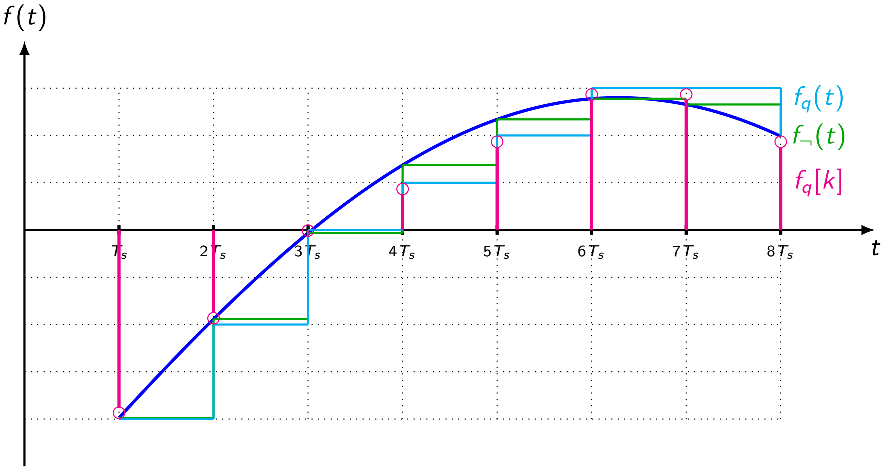
Quantisation



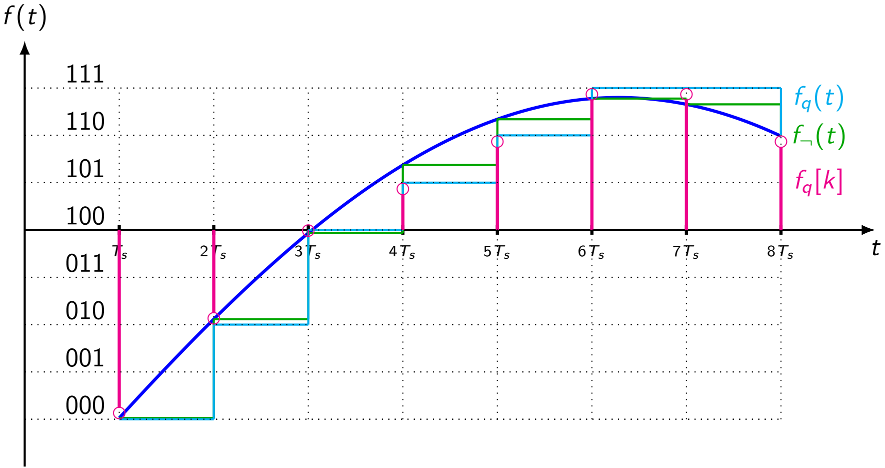
Quantisation



Quantisation



Quantisation and coding



Coding

Coding

By **coding** the quantised signal, we make it accessible to a **processor**.

- The **quantisation grid** must **cover** the **dynamic range** of the signal.
- With N bits, it is possible to code 2^N **levels** of quantisation.
- Quantisation followed by coding results in **loss of information**, and the **quantisation noise** relates to the **rounding** or **truncating error** in the representation of each sample.

Coding: fundamentals

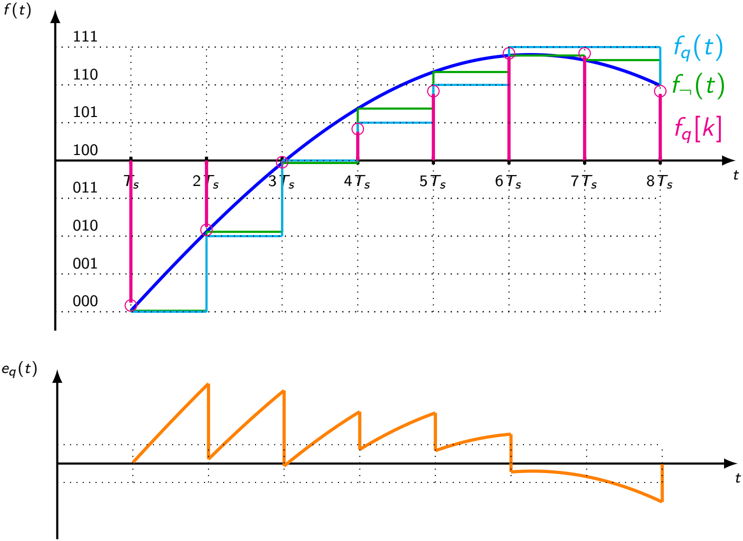
- When all representative bits representing the sample are 0, the associated value of the digital signal is f_{min}
- When all representative bits representing the sample are 1, the associated value of the digital signal is f_{max}
- The **precision** or **resolution** is related to the **Least Significant Bit**[79] (LSB). It is called the **quantum**, the **bit resolution** or the **quantisation step size**. With N bits, there will be 2^N quantisation levels and the quantisation step size is

$$q = (f_{max} - f_{min})/(2^N - 1)$$

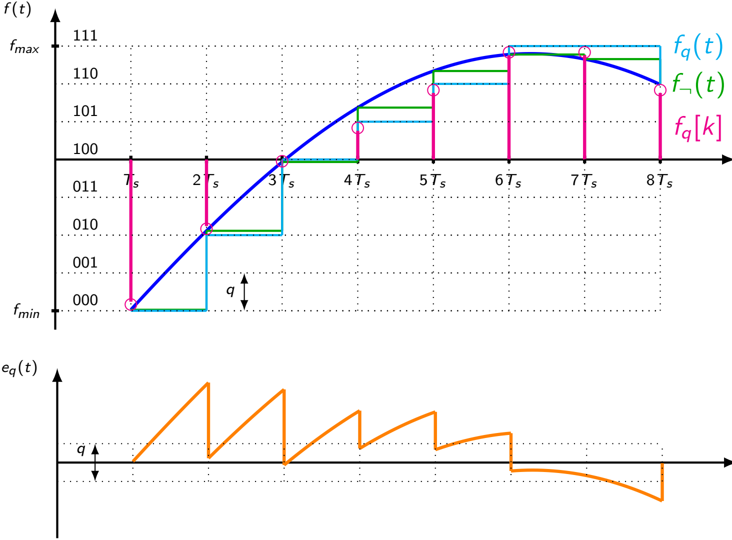
- The **quantisation noise** varies between $\pm\frac{\text{LSB}}{2}$ at the sample instants.

[79] Bit de poids faible

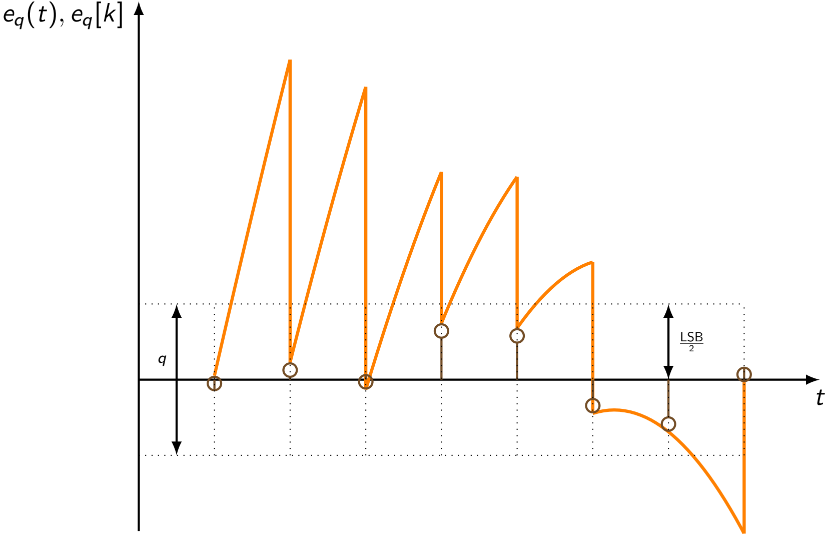
Quantisation and coding



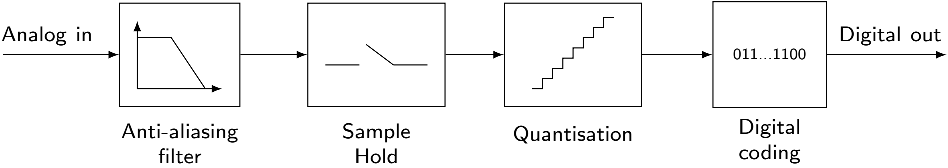
Quantisation and coding



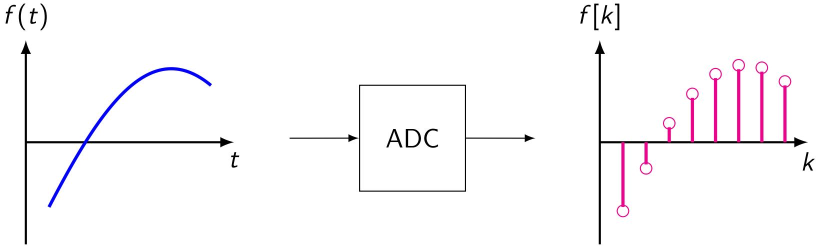
Quantisation error



Analog-to-Digital Conversion (ADC)



With an **infinite resolution**, assuming **ideal impulse sampling**, and **neglecting coding effects**, we have



An example from practice

F.Y.I.

- Let us take a **temperature transmitter** with the following characteristics:
  - **Range**: -18 to 93°C (0 to 200°F)
  - **Output**: Linearized 4 to 20 mA
- Assume that we use an **Analog Input** (AI) **module** with the following technical property:
  - **Input range for current measurement**: 4 to 20 mA, resolution 15 bits
- This means that $[4 \ldots 20]$ mA is the image of $[-18 \ldots 93]$ °C.
- After **quantisation** and **coding** we have that
  - 4mA corresponds to binary number 000000000000000 and the decimal number 0.
  - 20mA corresponds to the binary number 111111111111111 and to the decimal number $2^{15} - 1 = 32767$.

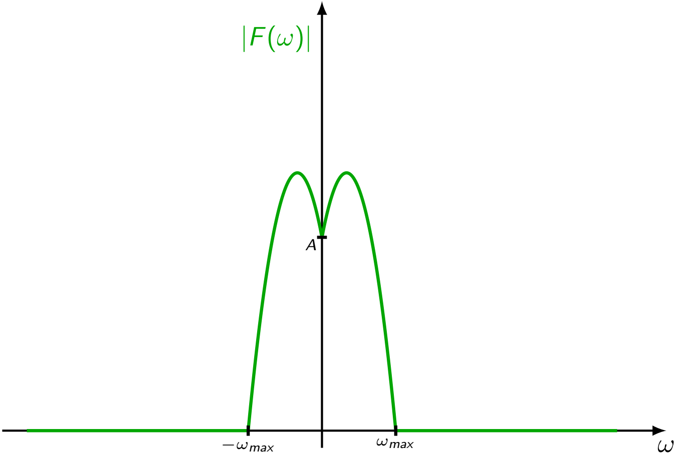
An example from practice

F.Y.I.

- Let us take a **temperature transmitter** with the following characteristics:
  - **Range**: -18 to 93°C (0 to 200°F)
  - **Output**: Linearized 4 to 20 mA
- Assume the we use an **Analog Input** (AI) **module** with the following technical property:
  - **Input range for current measurement**: 4 to 20 mA, resolution 15 bits
- This means that $[4 \ldots 20]$ mA is the image of $[-18 \ldots 93]$ °C.
- The **resolution** is computed as follows:

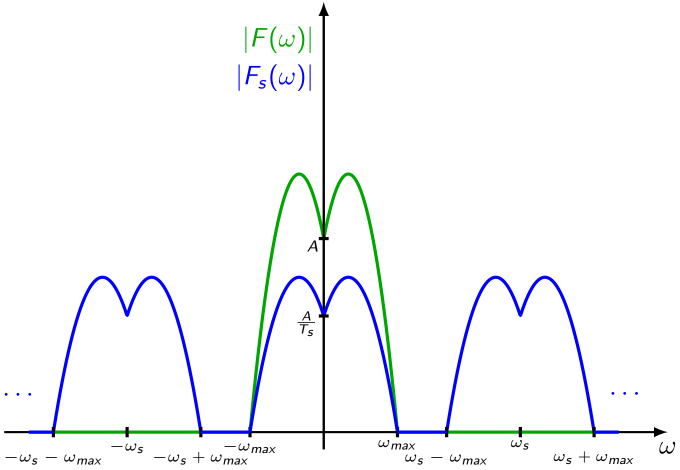
$$\text{resolution} = \frac{93 - (-18)}{(2^{15} - 1)} = 0.0034^{\circ}\text{C per bit.}$$

Low-pass spectrum of finite support: ideal reconstruction



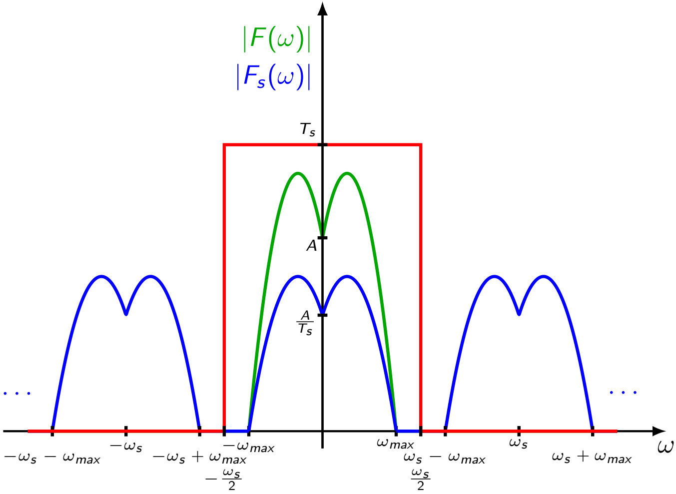
Nyquist condition is respected !

Low-pass spectrum of finite support: ideal reconstruction



Nyquist condition is respected !

Low-pass spectrum of finite support: ideal reconstruction



Nyquist condition is respected !

Low-pass spectrum of finite support: ideal reconstruction

If the initial signal $f(t)$ has a **low-pass spectrum** of **finite support**:, i.e. $F(\omega) = 0$ pour $\omega > \omega_{max}$, it is possible to **reconstruct** $f(t)$ from $f_s(t)$ using an **ideal low-pass filter**

$$H_{lp}(\omega) = \begin{cases} T_s & -\frac{\omega_s}{2} < \omega < \frac{\omega_s}{2} \\ 0 & \text{elsewhere} \end{cases}$$

The Fourier transform of the filtered signal is $F_r(\omega) = H_{lp}(\omega) F_s(\omega)$, i.e.

$$F_r(\omega) = \begin{cases} F(\omega) & -\frac{\omega_s}{2} < \omega < \frac{\omega_s}{2} \\ 0 & \text{elsewhere} \end{cases}$$

This corresponds to the **Fourier transform** of the **original signal**.

Low-pass spectrum of finite support: reconstruction

The ideal low-pass filter $H_{lp}(\omega)$ has a **non-causal** impulse response[80]

$$h_{lp}(t) = \frac{T_s}{2\pi} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} e^{j\omega t} d\omega = \frac{\sin(\pi t/T_s)}{\pi t/T_s} = \mathrm{sinc}(t/T_s)$$

The **reconstructed** signal $f_r(t)$ is the **convolution** of $f_s(t)$ and $h_{lp}(t)$

$$f_r(t) = f_s(t) * h_{lp}(t) = \int_{-\infty}^{\infty} f_s(\tau) h_{lp}(t-\tau)\, d\tau$$

The **ideal reconstruction** filter is **not realisable in practice** as

- it would require $f_s(\tau)$ for $\tau > t$ to reconstruct $f_r(t)$ and
- it requires an **infinite amount of data** !

There is always some level of **approximation**.

[80] The inverse Fourier transform is used here !

Low-pass spectrum of finite support: reconstruction

F.Y.I.

The **reconstructed** signal $f_r(t)$ is the **convolution** of $f_s(t)$ and $h_{lp}(t)$

$$
\begin{aligned}
f_r(t) &= f_s(t) * h_{lp}(t) = \int_{-\infty}^{\infty} f_s(\tau) h_{lp}(t-\tau)\, d\tau \\
&= \int_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} f(nT_s)\, \delta(\tau - n\,T_s) \right) \frac{\sin(\pi(t-\tau)/T_s)}{\pi(t-\tau)/T_s}\, d\tau \\
&= \sum_{n=-\infty}^{\infty} f(nT_s)\, \frac{\sin(\pi(t-nT_s)/T_s)}{\pi(t-nT_s)/T_s}
\end{aligned}
$$

Low-pass spectrum of finite support: reconstruction

F.Y.I.

The **reconstructed** signal $f_r(t)$ is the **convolution** of $f_s(t)$ and $h_{lp}(t)$

$$f_r(t) = \sum_{n=-\infty}^{\infty} f(nT_s)\,\frac{\sin(\pi(t-nT_s)/T_s)}{\pi(t-nT_s)/T_s}.$$

The **reconstructed** signal is thus an **interpolation** of **time-shifted cardinal sine signals** with amplitudes the samples $\{f(nT_s)\}$.

Taking $t = kT_s$ we can see that

$$f_r(kT_s) = \sum_{n=-\infty}^{\infty} f(nT_s)\,\frac{\sin(\pi(k-n))}{\pi(k-n)} = f(kT_s)$$

since

$$\frac{\sin(\pi(k-n))}{\pi(k-n)} = \begin{cases} 1 & k = n \\ 0 & k \neq n \end{cases}$$

The **reconstruction** is **exact** at the **sampling instants** $t = k\,T_s$.

Remember that the **ideal reconstruction** filter is **not realisable in practice** as it is **non-causal** and it requires an **infinite amount of data** !

There is always some level of **approximation**.

F.Y.I.

Nyquist-Shannon sampling theorem

If a continuous-time signal $f(t)$ is a **band-limited low-pass spectrum**, i.e. $F(\omega) = 0$ pour $\omega > \omega_{max}$, the **Nyquist condition** states

$$\omega_s > 2\,\omega_{max} \text{ or } f_s = \frac{1}{T_s} > \frac{\omega_{max}}{\pi}.$$

The original signal $f(t)$ can be reconstructed by passing the sampled signal through an **ideal low-pass filter**

$$H_{lp}(\omega) = \begin{cases} T_s & -\frac{\omega_s}{2} < \omega < \frac{\omega_s}{2} \\ 0 & \text{elsewhere} \end{cases}$$

The **ideal reconstruction** of signal is thus an **interpolation** of **time-shifted cardinal sine signals** from the samples $f(kT_s)$

$$\boxed{f_r(t) = \sum_{k=-\infty}^{\infty} f(kT_s)\,\frac{\sin(\pi(t-kT_s)/T_s)}{\pi(t-kT_s)/T_s}}$$

Ideal reconstruction is impossible in practice

In **practice**, the **exact recovery** of the original signal is **not possible** for several reasons:

- The continuous-time signal is **never really band limited** due to the presence of noise.
- The sampling is **not done exactly at uniform times**. Slight random variations of the sampling times are always present.
- In reality, the values taken by the samples do not correspond to the amplitude at a precise sampling instant, but rather they are the **average** over a small interval around kT_s.
- **Quantisation** and **coding effects** have to be taken into account.
- The filter required for exact recovery is an **ideal low-pass filter**, which **cannot be realized**; only an approximation is possible.

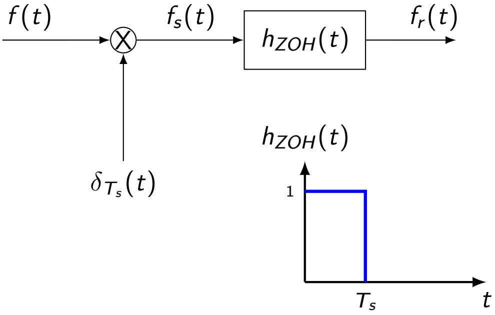
Digital-to-Analog Conversion (DAC)

- The ideal filter operation just described assumes availability of infinitely many samples. **This is not realistic !**
- Practical operation uses only a **finite number of samples**. Many techniques can be used to approximately reconstruct the signal.
- One such technique is to use a **Zero-Order Holder (ZOH)** reconstruction filter. We will only consider ZOH reconstruction.
- A **First-Order-Hold (FOH)** is sometimes used instead of ZOH. For the FOH, the signal is reconstructed as a **piecewise linear approximation** to the original signal that was sampled.

Zero Order Hold (ZOH) reconstruction

The **Zero Order Hold (ZOH)** reconstruction process can be represented by an LTI system having **impulse response** $h_{ZOH}(t)$ of width T_s.

Let us represent **ideal sampling** followed by **ZOH reconstruction**



Zero Order Hold (ZOH) reconstruction

The **output** of the ZOH process is

$$f_r(t) = [f_s * h_{ZOH}](t)$$

The associated **Fourier transform** is

$$\begin{aligned}
F_r(\omega) &= F_s(\omega)\, H_{ZOH}(\omega) = \left[\frac{1}{T_s} \sum_{k=-\infty}^{\infty} F(\omega - k\omega_s) \right] H_{ZOH}(\omega) \\
H_{ZOH}(\omega) &= \mathcal{L}\left[h_{ZOH}(t)\right]_{s=j\omega} = \frac{1}{s}\left(1 - e^{-T_s s}\right)_{s=j\omega} \\
H_{ZOH}(\omega) &= \mathcal{F}\left[h_{ZOH}(t)\right] = \int_{-\infty}^{\infty} h_{ZOH}(t) e^{-j\omega t} dt = \int_0^{T_s} e^{-j\omega t} dt = \frac{1}{j\omega}\left(1 - e^{-j\omega T_s}\right) \\
&= \frac{e^{-j\omega \frac{T_s}{2}}}{j\omega} \left(e^{j\omega \frac{T_s}{2}} - e^{-j\omega \frac{T_s}{2}}\right) = \frac{e^{-j\omega \frac{T_s}{2}}}{j\omega} \left(2j \sin(\omega \frac{T_s}{2})\right) \\
&= T_s\, e^{-j\omega \frac{T_s}{2}} \left[\frac{\sin(\omega \frac{T_s}{2})}{\omega \frac{T_s}{2}} \right] = T_s\, e^{-j\omega \frac{T_s}{2}} \ \mathrm{sinc}(f T_s)
\end{aligned}$$

Zero Order Hold (ZOH) reconstruction

Ideal reconstruction assuming the Nyquist condition is respected:

$$F_r(\omega) = F(\omega)$$

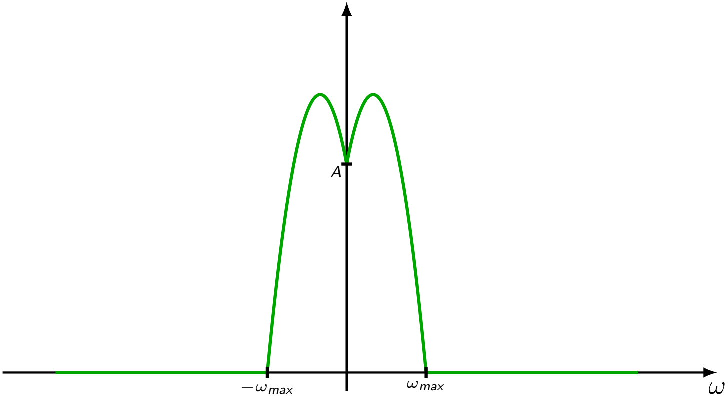
ZOH reconstruction assuming the Nyquist condition is respected:

$$F_r(\omega) = e^{-j\omega \frac{T_s}{2}} \frac{\sin(\omega \frac{T_s}{2})}{\omega \frac{T_s}{2}} \sum_{k=-\infty}^{\infty} F(\omega - k\omega_s)$$

The recovered signal $f_r(t)$ is **distorted** by the **spectrum of** $h_{ZOH}(t)$.
The repetitions of the spectrum are **still present** although **attenuated**.

Low-pass spectrum of finite support: ZOH spectrum

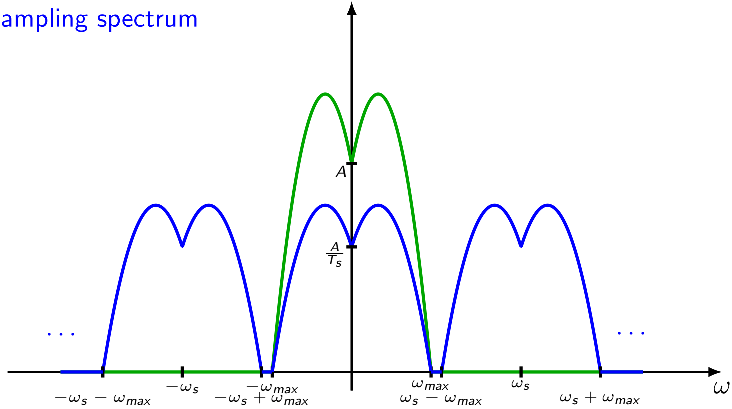
$|F(\omega)|$: Original spectrum



Low-pass spectrum of finite support: ZOH spectrum

$|F(\omega)|$: Original spectrum

$|F_s(\omega)|$: Ideal sampling spectrum

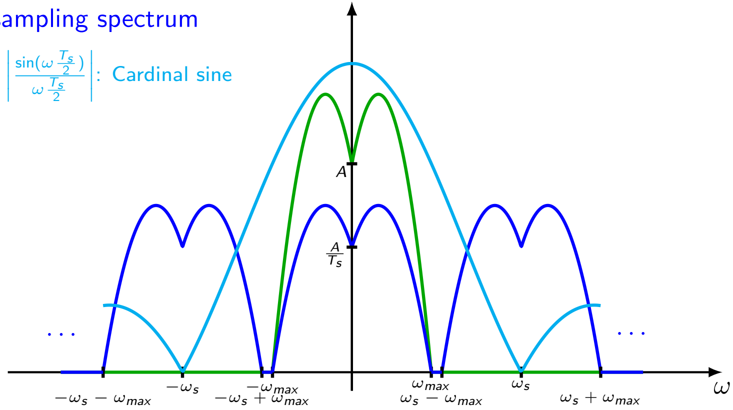


Low-pass spectrum of finite support: ZOH spectrum

$|F(\omega)|$: Original spectrum

$|F_s(\omega)|$: Ideal sampling spectrum

$|H_{ZOH}(j\omega)| = T_s \left|\frac{\sin(\omega \frac{T_s}{2})}{\omega \frac{T_s}{2}}\right|$: Cardinal sine



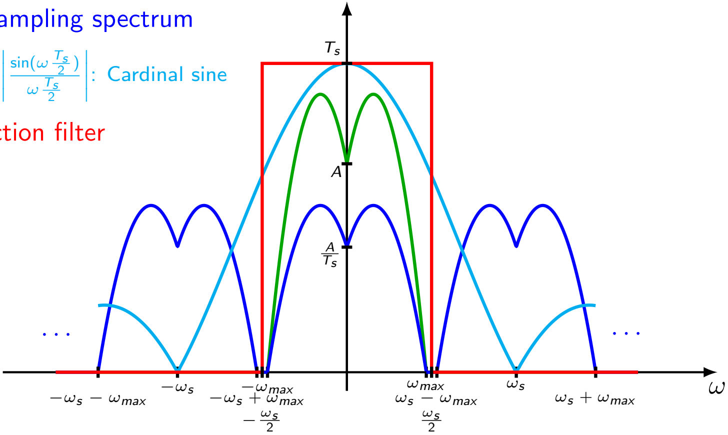
Low-pass spectrum of finite support: ZOH spectrum

$|F(\omega)|$: Original spectrum

$|F_s(\omega)|$: Ideal sampling spectrum

$|H_{ZOH}(j\omega)| = T_s \left| \frac{\sin(\omega \frac{T_s}{2})}{\omega \frac{T_s}{2}} \right|$: Cardinal sine

Ideal reconstruction filter



Low-pass spectrum of finite support: ZOH spectrum

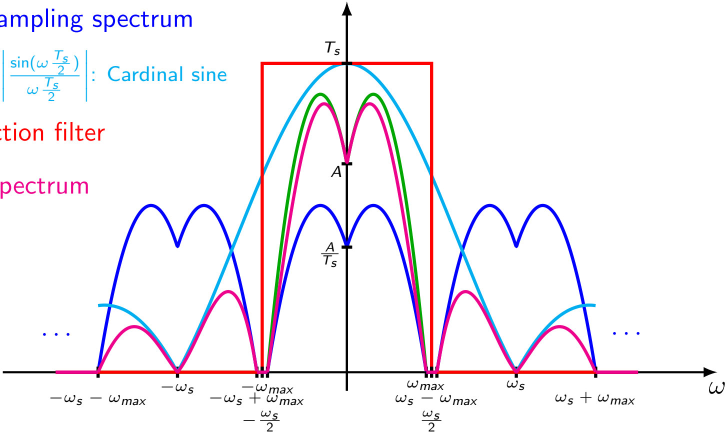
$|F(\omega)|$: Original spectrum

$|F_s(\omega)|$: Ideal sampling spectrum

$|H_{ZOH}(j\omega)| = T_s \left|\frac{\sin(\omega \frac{T_s}{2})}{\omega \frac{T_s}{2}}\right|$: Cardinal sine

Ideal reconstruction filter

$|F_r(\omega)|$: ZOH spectrum



ZOH reconstruction: consequences

The recovered signal $f_r(t)$ is **distorted** by the **spectrum of** $h_{ZOH}(t)$.
The repetitions of the spectrum are **still present** although **attenuated**.

As a consequence, the sampling frequency will have to satisfy the Nyquist condition with

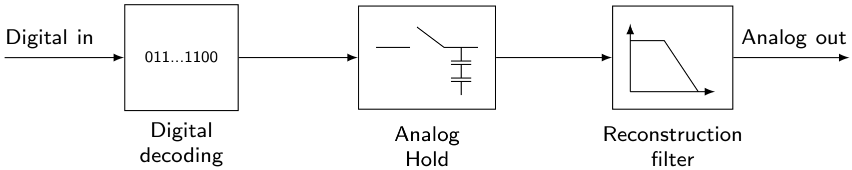
$$\omega_s >> 2\,\omega_{max} \text{ or } f_s = \frac{1}{T_s} >> \frac{\omega_{max}}{\pi}$$

to attenuate the distortion effects of the ZOH filter. This is known as **oversampling**.

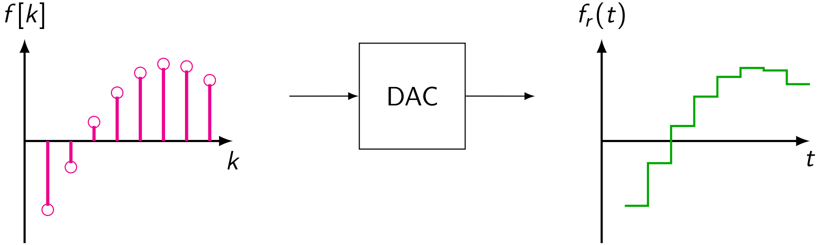
The ZOH reconstruction filter is often followed by a **reconstruction low-pass filter** to attenuate the repetitions in the spectrum of $f_r(t)$.

Sometimes an **additional filter** is included to **attenuate** (invert) the **distortion effects** of the ZOH filter within the frequency band of interest.

Digital-to-Analog Conversion (DAC)



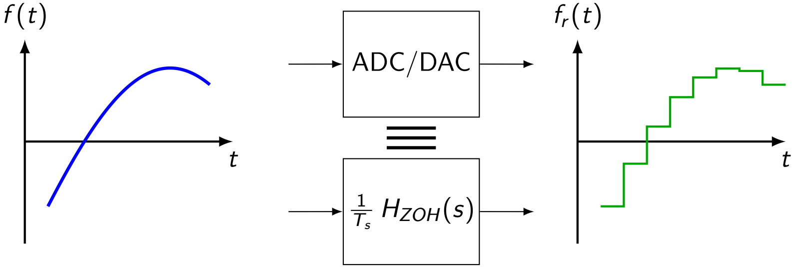
Neglecting decoding effects, we have[81]



[81] Often the Analog Hold is a Zero Order Hold (ZOH) !

ADC followed by DAC

With an **infinite resolution** and assuming **ideal impulse sampling**, we have



where

$$H_{ZOH}(s) = \frac{1}{s}\left(1 - e^{-T_s\, s}\right)$$

7. Discrete-time signals and systems

Discrete-time signals
Discrete-time systems

Discrete-time signals

Discrete-time signals

A **discrete-time signal** $f[k]$ can be tought of as a **real**- or **complex-valued** function of **integer sample index** k:

$$\begin{array}{rccc} f[.]: & \mathbb{Z} & \longrightarrow & \mathbb{R} \quad (\mathbb{C}) \\ & k & \longrightarrow & f[k] \end{array}$$

The signal is **defined** for all **integers** k; it is **not defined** for **non-integer values** of k.

Although in many situations, discrete-time signals are obtained from continuous-time signals by sampling, that is **not always** the case. Many signals are inherently discrete: final values attained daily by company shares.

Fibonacci sequence: $f[n+1] = f[n] + f[n-1], n \geq 2, f[0] = 1, f[1] = 1.$

Periodic signals

A discrete-time signal $f[k]$ is **periodic** if

- it is defined for **all possible** integer values of k, $-\infty < k < \infty$ and
- there is a **positive integer** N_p, the **period** of $f[k]$, such that

$$f[k + N_p] = f[k], \ \ \forall k \in \mathbb{Z}.$$

A **discrete sinusoidal signal** is **not necessarily periodic**.

A sequence $f[k] = A\cos(\Omega_0 k + \phi)$ is **periodic only if** there exist coprime positive integers N_p and m such that

$$\Omega_0 = \frac{2\pi m}{N_p}, \ \text{i.e. } \Omega_0 \text{ is a } \underline{\textbf{rational multiple}} \text{ of } 2\pi.$$

The **period** is N_p. Indeed,

$$\begin{aligned} f[k + N_p] = A\cos\left(\frac{2\pi m}{N_p}(k + N_p) + \phi\right) &= A\cos\left(\frac{2\pi m}{N_p}k + 2\pi m + \phi\right) \\ &= A\cos\left(\frac{2\pi m}{N_p}k + \phi\right) = f[k] \end{aligned}$$

Sampling of periodic signals

When **sampling** an **analog sinusoidal signal** $f(t) = A\cos(\omega_0 t + \phi)$ of period T, the discrete signal

$$f[k] = A\cos(\omega_0 k T_s + \phi) = A\cos\left(\frac{2\pi T_s}{T}\,k + \phi\right)$$

is obtained. This sampled signal is **periodic** if

$$\frac{T_s}{T} = \frac{m}{N_p} \iff \boxed{mT = N_p T_s}$$

for coprime positive integers N_p and m.

To **avoid frequency aliasing** the **sampling period** should also **satisfy**[82]

$$\omega_s \geq 2\omega_0 \iff T_s \leq \frac{\pi}{\omega_0} = \frac{T}{2}.$$

[82] At least two samples per period !

Sampling of periodic signals

The **sampled signal**

$$f[k] = A\cos(\omega_0 k T_s + \phi) = A\cos\left(\frac{2\pi T_s}{T} k + \phi\right)$$

is **periodic** if

$$\frac{T_s}{T} = \frac{m}{N_p} \iff \boxed{mT = N_p T_s}$$

for coprime positive integers N_p and m.

This conditions says that a **period** $(m = 1)$ or **several periods** $(m > 1)$ should be **divided** into $N_p > 0$ segments of duration T_s seconds.

The "equivalent continuous-time" period is $N_p T_s = mT$, i.e. a **multiple** m of the original period T.

If the condition is **not** satisfied, then the discretized sinusoidal signal is **not periodic**.

Sampling of periodic signals: same periodicity

The **sampled signal**

$$f[k] = A\cos(\omega_0 k T_s + \phi) = A\cos\left(\frac{2\pi T_s}{T}\,k + \phi\right)$$

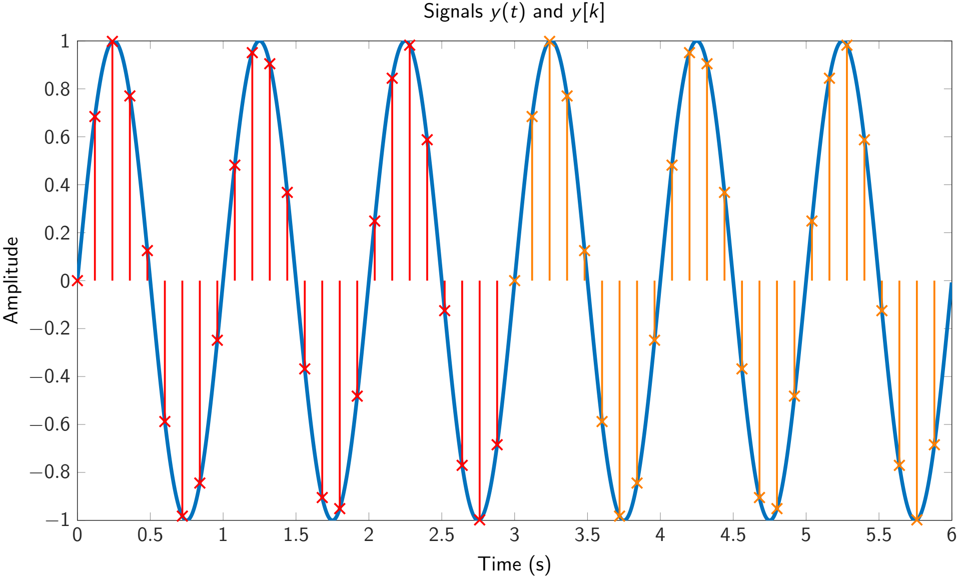
is **periodic** with <u>**same**</u> "equivalent continuous-time" period T if the sampling period T_s **obeys**

$$\boxed{T = N_p T_s}$$

Sampling of periodic signals

```
close all; clear all

% Period of sine
T = 1;

% Sampling period
m = 3; % Test m = 1, m = 3, m = 3.1
Np = 25;
Ts = m*T/Np; fs = 1/Ts; fprintf('fs = %4.2f Hz\n', fs)

% Number of data
K = 2; N = K*Np; % K discrete periods will be considered

tsampled = (0:N-1)*Ts; ysampled = sin((2*pi/T)*(tsampled));

fact = 100;
tsim = (0:N*fact)*Ts/fact; ysim = sin((2*pi/T)*(tsim));

figure
plot(tsim,ysim)
hold on
stem(tsampled,ysampled,'x')
hold off
title('Signals y(t) and y[k]')
xlabel('Time (s)'), xlim([0 tsim(end)])
ylabel('Amplitude')
```

Sampling of periodic signals



Energy

F.Y.I.

Energy

The **energy** of a discrete-time signal is defined as

$$E = \sum_{k=-\infty}^{\infty} |f[k]|^2$$

The signal $f[k]$ is said to have **finite energy** or to be **square summable** if $E < \infty$.

The signal $f[k]$ is called **absolutely summable** if

$$\sum_{k=-\infty}^{\infty} |f[k]| < \infty.$$

Power

F.Y.I.

Power

If a signal has **infinite energy**, the **average power** is defined as

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

The signal $f[k]$ is said to have **finite power** if $P < \infty$.

Example: discrete unit step[83] (infinite energy)

$$P = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{0}^{N} 1 = \lim_{N \to \infty} \frac{N+1}{2N+1} = \frac{1}{2}$$

[83] The discrete unit step is introduced subsequently.

Time shifting and reflection

Time shifting and **reflection** are very similar to the continuous-time cases, the only difference being that the operations are done using **integers**.

Time shifting and reflection

Given a positive integer N and a discrete-time signal $f[k]$.

- The discrete-time signal $g[k]$ is $f[k]$ **delayed** by N samples if $g[k] = f[k-N]$, i.e. $f[k]$ shifted to the **right** N samples.
- The discrete-time signal $g[k]$ is $f[k]$ **advanced** by N samples if $g[k] = f[k+N]$, i.e. $f[k]$ shifted to the **left** N samples.
- The discrete-time signal $g[k]$ is $f[k]$ **reflected** if $g[k] = f[-k]$.

Even and odd signals

Even and odd signals

A discrete signal

$$\begin{aligned} f[k] \text{ is } \textbf{even} \quad &\Longleftrightarrow \quad f[k] = f[-k] \\ f[k] \text{ is } \textbf{odd} \quad &\Longleftrightarrow \quad f[k] = -f[-k] \end{aligned}$$

A discrete signal $f[k]$ can be **decomposed** as the **sum** of an **even** component and an **odd** component, i.e.

$$\begin{aligned} f[k] &= \underbrace{\frac{1}{2}(f[k] + f[-k])}_{f_e[k]} + \underbrace{\frac{1}{2}(f[k] - f[-k])}_{f_o[k]}, \\ &= f_e[k] + f_o[k]. \end{aligned}$$

Discrete-time complex exponential

Discrete-time complex exponential

A **discrete-time complex exponential** is a signal of the form

$$\begin{aligned} f[k] &= A\,\alpha^k = K\,e^{j\phi}\,r_0^k\,e^{j\Omega_0 k} = K\,r_0^k\,e^{j(\Omega_0 k+\phi)} \\ &= K\,r_0^k\,(\cos(\Omega_0 k+\phi)+j\sin(\Omega_0 k+\phi)) \end{aligned}$$

with $A = K\,e^{j\phi}$ and $\alpha = r_0\,e^{j\Omega_0}$.

Sample a **continuous-time exponential**

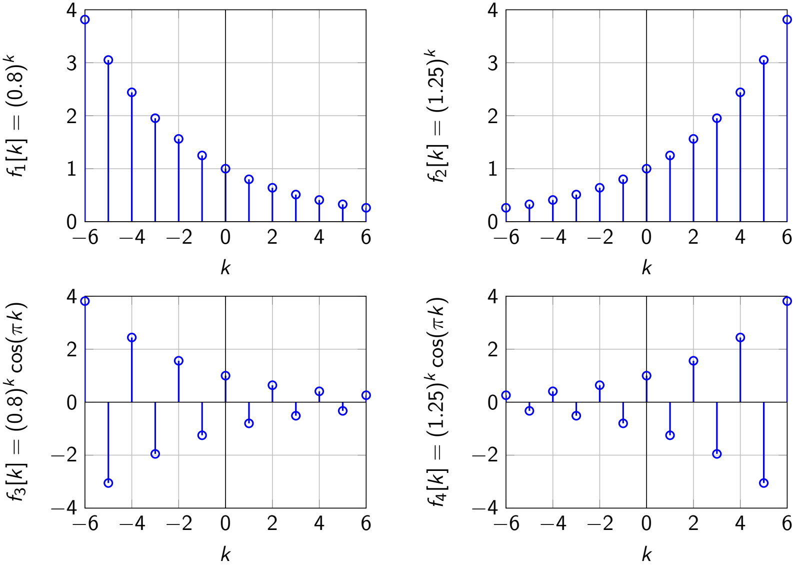
$$f(t) = A\,e^{at}$$

with $A = K\,e^{j\phi}$ and $a = \sigma_0 + j\omega_0$ **both complex**. The **sampled signal** is

$$f[k] = K\,e^{j\phi}\,e^{(\sigma_0+j\omega_0)kT_s} = K\,(e^{\sigma_0 T_s})^k e^{j(\omega_0 T_s k+\phi)} = K\,r_0^k e^{(j\Omega_0 k+\phi)}$$

where $r_0 = e^{\sigma_0 T_s}$ and $\Omega_0 = \omega_0\,T_s$.

Discrete-time complex exponential



Discrete-time unit-step and unit-impulse

Discrete-time unit-step and unit-impulse

The discrete-time **unit-step** $u[k]$ and **unit-impulse** $\delta[k]$ are defined as

$$u[k] = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0 \end{cases}$$

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

These two signals are **related** as follows:

$$\delta[k] = u[k] - u[k-1]$$

$$u[k] = \sum_{n=0}^{\infty} \delta[k-n] = \sum_{m=-\infty}^{k} \delta[m]$$

Generic representation of discrete-time signals

Generic representation of discrete-time signals

Any discrete-time signal $x[k]$ can be **represented** using **shifted unit-impulse signals** as

$$x[k] = \sum_{n=-\infty}^{\infty} x[n]\,\delta[k-n].$$

Properties of discrete-time systems

Linearity

A discrete-time process H is **linear**, if the **superposition principle** is applicable, i.e.

$$y[k] = H(\alpha_1 x_1[k] + \alpha_2 x_2[k]) = \alpha_1 H(x_1[k]) + \alpha_2 H(x_2[k]).$$

Time invariance

A discrete-time process H is **time-invariant**, if for an input $x[k]$ and corresponding output $y[k] = H(x[k])$, the output corresponding to a delayed or advanced version of $x[k]$, $x[k \pm N]$, is

$$y[k \pm N] = H(x[k \pm N]).$$

Properties of discrete-time systems

Causality

A discrete-time **process** H is **causal** if

- for a **causal** input, i.e. $x[k] = 0$ for $k < 0$, and **zero initial conditions** in $k = 0$, this **implies** a **causal** output $y[k]$, i.e. $y[k] = 0$ for $k < 0$.
- its **impulse response** $h[k]$ is **causal**, i.e.

$$h[k] = 0, \quad k < 0.$$

Convolution sum

Convolution sum

Suppose $h[k]$ is the **impulse response** of a **LTI system**, i.e. the **response to a unit-impulse** $\delta[k]$ with **zero initial conditions**.

The **response** of the system to **any input** $x[k]$ is

$$\begin{aligned} y[k] &= x[k] * h[k] = h[k] * x[k] \\ &= \sum_{n=-\infty}^{\infty} x[n]\, h[k-n] = \sum_{m=-\infty}^{\infty} h[m]\, x[k-m]. \end{aligned}$$

Convolution sum

- By **definition**, the response to an impulse $\delta[k]$ is the impulse response $h[k]$.
- By **time invariance**, the response to $\delta[k-n]$ is $h[k-n]$.
- By **linearity**, the response to $x[n]\,\delta[k-n]$ is $x[n]\,h[k-n]$.
- By **superposition** (linearity), the response to

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

is

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

Causal system

Causal system

Suppose $h[k]$ is the impulse response of a **causal LTI systems**, i.e.

$$h[k] = 0, \quad k < 0.$$

The response to a **causal signal**

$$x[k] = 0, \quad k < 0$$

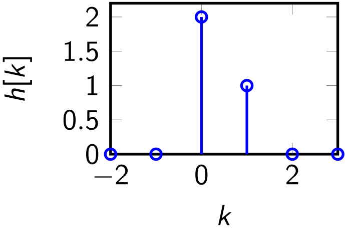
is

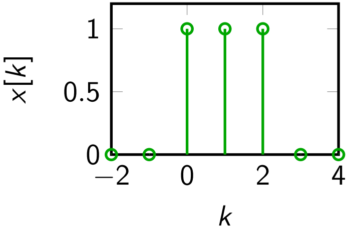
$$y[k] = \sum_{n=0}^{k} x[n]\, h[k-n] = \sum_{m=0}^{k} h[m]\, x[k-m].$$

The **lower limit** of the first sum results from the **causality** of the **input** signal. The **upper limit** of the first sum results from the **causality** of the **impulse response**.

Example

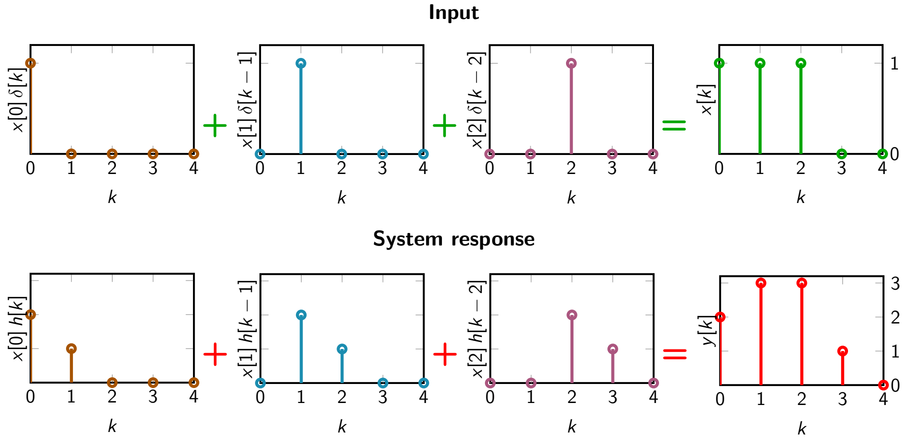
- **Impulse response**: $h[k] = 2\,\delta[k] + \delta[k-1]$
- **Input**: $x[k] = u[k] - u[k-3] = \delta[k] + \delta[k-1] + \delta[k-2]$





Example: interpretation of the convolution sum

- **Impulse response**: $h[k] = 2\,\delta[k] + \delta[k-1]$
- **Input**: $x[k] = u[k] - u[k-3] = \delta[k] + \delta[k-1] + \delta[k-2]$



Graphical convolution

Suppose the **input** signal $x[k]$ and the **impulse response** $h[k]$ are both **causal** signals.

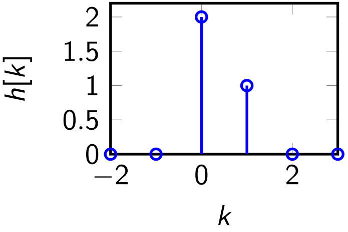
Graphical convolution

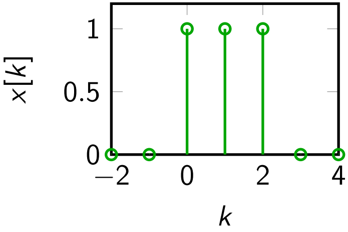
- Multiply $x[n]$ with a version of the impulse response $h[k-n]$ that is **reflected** and **shifted** by k samples to the **right**.
- **Sum** the resulting **product** from 0 to k.

- The **convolution sum** is **computationally** quite **expensive**. A **more efficient method** will be to use the z-**transform**.
- The convolution of signals of length (supports) M and N results in a signal of **length** $M+N-1$.

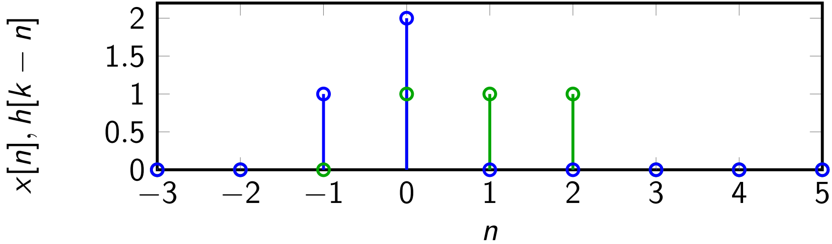
Example

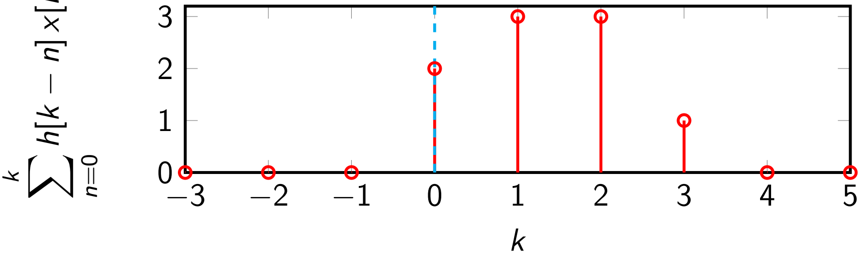
- **Impulse response**: $h[k] = 2\,\delta[k] + \delta[k-1]$
- **Input**: $x[k] = u[k] - u[k-3] = \delta[k] + \delta[k-1] + \delta[k-2]$



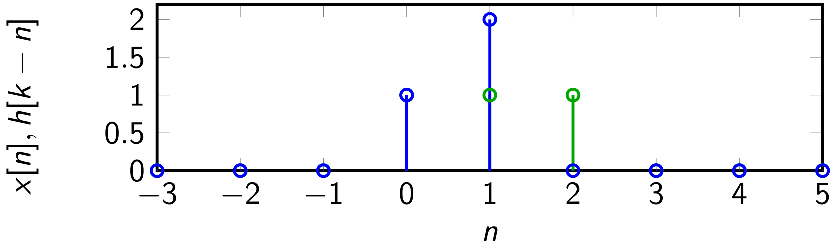


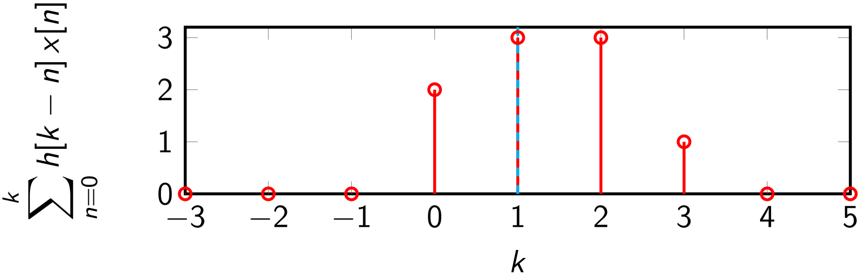
Example: graphical convolution ($k = 0$)



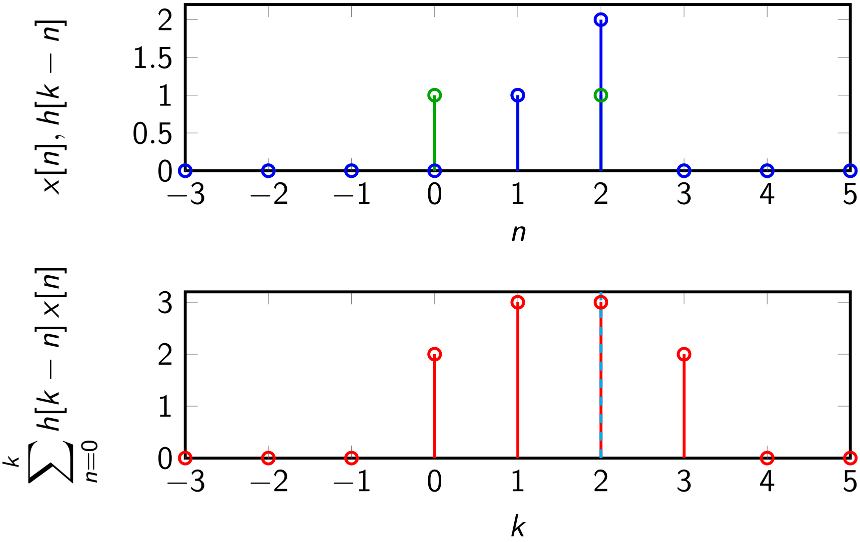


Example: graphical convolution ($k = 1$)

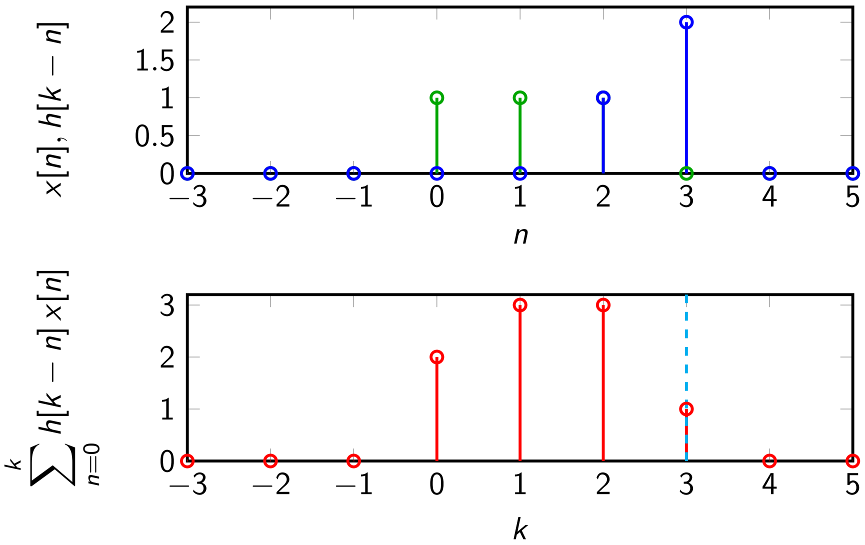




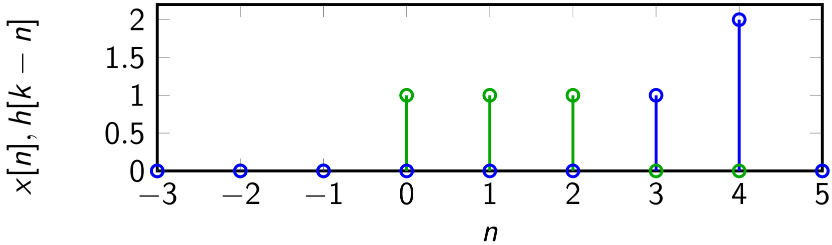
Example: graphical convolution ($k = 2$)

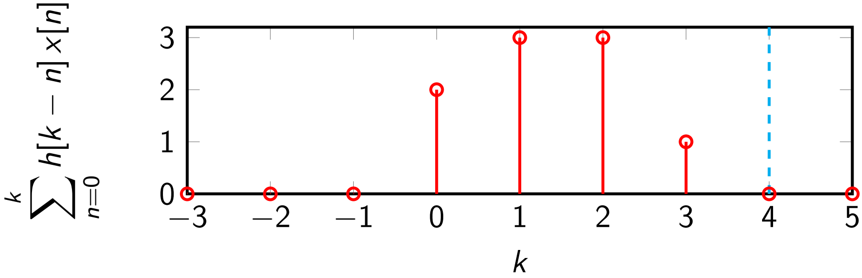


Example: graphical convolution ($k = 3$)

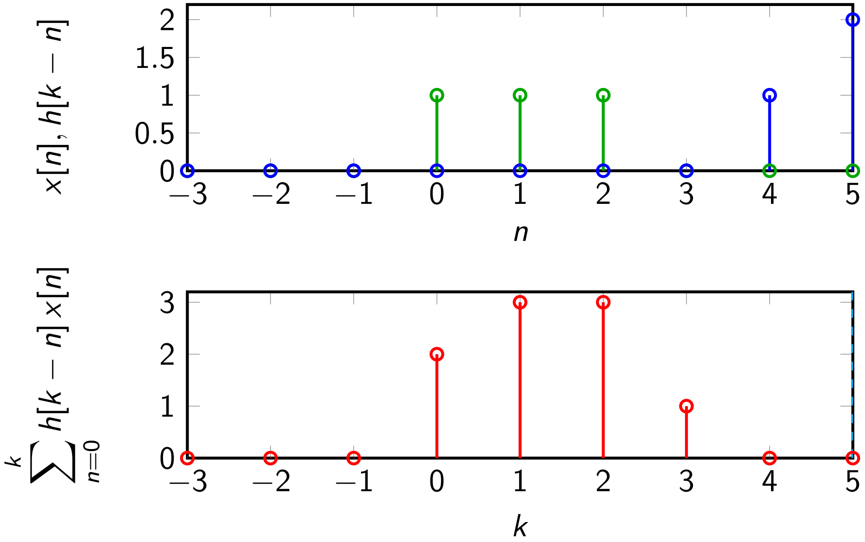


Example: graphical convolution ($k = 4$)





Example: graphical convolution ($k = 5$)



Difference equations

Difference equations in the **discrete-time case** are **analogous** to **differential equations** in the **continuous-time case**.

This type of equations with **constant coefficients** are used to **describe discrete-time systems** using the **addition**, **multiplication** and **time-shifting operations**.

Recursive system

In general, the **relation** between the **input** $x[k]$ and the **output** $y[k]$ can be written

$$\boxed{y[k] = -\sum_{m=1}^{N} a_m \, y[k-m] + \sum_{m=0}^{M} b_m \, x[k-m], \quad k > 0}$$

with initial conditions $y[-k], k = 1, \cdots, N$. This system is **recursive** and is also called an **Infinite Impulse Response** (IIR) system.

Difference equations and IIR: an example (1)

Let us take the simplest **difference equation** with **zero initial conditions**, i.e.

$$y[k] = -a\, y[k-1] + b\, x[k]$$

Note that

$$\begin{aligned}
y[k] &= -a\,(-a\, y[k-2] + b\, x[k-1]) + b\, x[k] \\
&= -a\,(-a\,(-a\, y[k-3] + b\, x[k-2]) + b\, x[k-1]) + b\, x[k] \\
&= -a\,(-a\,(-a\,(-a\, y[k-4] + b\, x[k-3]) + b\, x[k-2]) + b\, x[k-1]) + b\, x[k] \\
&\vdots \qquad\qquad \vdots
\end{aligned}$$

We obtain

$$\begin{aligned}
y[k] &= b\, x[k] - ab\, x[k-1] + a^2 b\, x[k-2] - a^3 b\, x[k-3] + \ldots \\
&= \sum_{m=0}^{\infty} (-a)^m b\, x[k-m] = \sum_{m=0}^{\infty} h[m]\, x[k-m]
\end{aligned}$$

This system has an **infinite impulse reponse** $h[k] = (-a)^k b\, u[k]$!

FIR system

The second type of discrete-time system is a **finite impulse response system**, also called a **non-recursive** system.

FIR system or non-recursive system

The **relation** between the **input** $x[k]$ and the **output** $y[k]$ can be written as

$$\boxed{y[k] = \sum_{m=0}^{M} b_m \, x[k-m]}$$

There are **no initial conditions** as the system is **non-recursive**. This system is also called a **Finite Impulse Response** (FIR) system.

FIR system

FIR system or non-recursive system

The response of a **non recursive** filter is the **convolution** between a **causal input** signal $x[k]$ and the **finite impulse response**

$$h[k] = \begin{cases} b_k, & k = 0, \cdots, M \\ 0 & \text{elsewhere} \end{cases}$$

i.e.

$$y[k] = [h * x][k] = \sum_{m=-\infty}^{\infty} h[m]\, x[k-m] = \sum_{m=0}^{M} b_m\, x[k-m].$$

BIBO stability

BIBO stability

A LTI discrete-time system is called **Bounded-Input-Bounded-Output** (BIBO) stable if and only if for **any bounded input** $|x[k]| \leq M$, the output $y[k]$ is **bounded**.

A LTI discrete-time system is **BIBO stable**, if its **impulse response** $h[k]$ is **absolutely summable**,

$$\boxed{\sum_k |h[k]| < \infty}$$

A much **simpler way** to test the **stability** of an IIR system will be based on the **location of the poles** of the z-transform. Note that a **FIR system** is **always stable**.

BIBO stability: sufficiency

Assume that the **input** $x[k]$ is **bounded**, i.e. that there is a **bounded** M for which $|x[k]| \leq M$ for all k. Then

$$\begin{aligned} |y[k]| = \left| \sum_{n=-\infty}^{\infty} x[n]\, h[k-n] \right| &= \left| \sum_{m=-\infty}^{\infty} h[m]\, x[k-m] \right| \\ &\leq \sum_{m=-\infty}^{\infty} |h[m]|\, |x[k-m]| \\ &\leq M \sum_{m=-\infty}^{\infty} |h[m]| \end{aligned}$$

It follows that

$$\sum_{m=-\infty}^{\infty} |h[m]| < \infty$$

is a **sufficient condition** for **BIBO stability**[84].

[84] It is also possible to show that this condition is **necessary** although this is **more complicated**.

8. Z-transform

Introduction
Definitions
Z-transform and non causal signals
Mapping the s-plane into the z-plan
Computation of z-transforms
Properties of the z-transform
Transfer function
Stability
Dynamic behaviour
Initial and final value theorems
Inverse z-transform
LTI system analysis
Laplace, Fourier and z-transforms
Sampled cosine
Matlab and Octave
One-sided z-transforms
Basic properties of one-sided z-transforms

Geometrical series

Geometrical series with $k_0 \leq k_1, k_0, k_1 \in \mathbb{Z}$

$$S_{k_0,k_1} = \sum_{k=k_0}^{k_1} q^k = \frac{q^{k_0} - q^{k_1+1}}{1-q}$$

Proof by induction: The property is **verified** for $k_1 = k_0$.
Suppose that the property is verified for $k_1 = k$, then

$$\begin{aligned} S_{k_0,k+1} &= S_{k_0,k} + q^{k+1} = \frac{q^{k_0} - q^{k+1}}{1-q} + q^{k+1} \\ &= \frac{q^{k_0} - q^{k+1} + q^{k+1} - q^{k+2}}{1-q} = \frac{q^{k_0} - q^{k+2}}{1-q} \end{aligned}$$

i.e. the property is **verified** for $k_1 = k + 1$ and therefore for **all** k_1.

Geometrical series

Geometrical series with $k_1 > 0$

$$S_{0,k_1} = \sum_{k=0}^{k_1} q^k = \frac{1 - q^{k_1+1}}{1 - q}$$

When $\boxed{|q| < 1}$ and with $k_1 \longrightarrow \infty$, the series **converges** to

$$\boxed{S_{0,\infty} = \sum_{k=0}^{\infty} q^k = \frac{1}{1 - q}}$$

Z-transform: Laplace transform of a sampled signal

Z-transform

The **Laplace transform** of a **sampled signal**

$$f_s(t) = \sum_k f(kT_s)\,\delta(t - kT_s)$$

is given by

$$F_s(s) = \mathcal{L}[f_s(t)] = \sum_k f(kT_s)\,\mathcal{L}[\delta(t - kT_s)] = \sum_k f(kT_s)\,e^{-kT_s s}$$

Defining $\boxed{z = e^{T_s s}}$, the equation can **rewritten** as

$$\begin{aligned} \mathcal{Z}[f(kT_s)] &= \mathcal{L}[f_s(t)]\big|_{z=e^{T_s s}} \\ &= \sum_k f(kT_s) z^{-k} = \sum_k f[k] z^{-k} \end{aligned}$$

which is the **z-transform** of the sampled signal $f(kT_s) = f[k]$.

Definition

Two-sided z-transform

Given a **discrete-time signal** $f[k]$, $-\infty < k < \infty$, the **two-sided** z-transform is

$$\boxed{F(z) = \mathcal{Z}[f[k]] = \sum_{k=-\infty}^{\infty} f[k]\, z^{-k}}$$

This transform is defined in a **Region Of Convergence** (ROC) in the z-plane.

The two-sided z-transform is **not useful** in solving difference equations with initial conditions, **just as** the two-sided Laplace transform is not useful in solving differential equations with initial conditions. It is necessary to define a **one-sided** z-transform.

Definition

One-sided z-transform

The **one-sided** z-transform of a given discrete-time signal $f[k]$, **causal** with $f[k] = 0$ for $k < 0$, or **non-causal** and **made causal** by multiplication with a unit step $u[k]$ is

$$\boxed{F(z) = \mathcal{Z}[f[k]u[k]] = \sum_{k=0}^{\infty} f[k]\, z^{-k}}$$

with **region of convergence** ROC.

Region of convergence

For the **one-sided** z-transform to be defined for a discrete-time **causal** $f[k]$,

$$
\begin{aligned}
|F(z)| = \left|\sum_{k=0}^{\infty} f[k] z^{-k}\right| \quad &\leq \quad \sum_{k=0}^{\infty} |f[k] r^{-k} e^{-jk\Omega}| \\
&= \quad \sum_{k=0}^{\infty} |f[k]\, r^{-k}| < \infty,
\end{aligned}
$$

that is $f[k]\, r^{-k}$ must be **absolutely summable**[85]. This is often the case by an **appropriate choice** of r even if $f[k]$ itself is not absolutely summable. The values of r for which the integral converge define the **ROC**; the discrete frequency Ω does **not** affect the ROC.

$$
\boxed{\text{ROC} \; = \left\{ z = r\, e^{j\Omega} \text{ such that } \sum_{k=0}^{\infty} |f[k]\, r^{-k}| < \infty \right\}}
$$

[85] Here we have only proved the **sufficiency** of this condition. It is also possible to prove its **necessity** although this is **more complicated**.

Z-transforms: fundamentals

This section **hinges** on:

- **Geometrical series**:

$$\boxed{S_{0,\infty} = \sum_{k=0}^{\infty} q^k = \frac{1}{1-q}, \ \ |q| < 1}$$

- The **one-sided** z-transform is

$$\boxed{F(z) = \mathcal{Z}[f[k]] = \sum_{k=0}^{\infty} f[k]\, z^{-k}}$$

- The **relation** between the z-variable and the s-variable, i.e.

$$\boxed{z = e^{T_s s}}$$

Poles and zeros

Poles and zeros

Suppose $F(z) = \mathcal{Z}[f[k]] = \frac{n(z)}{d(z)}$ is a **rational** function with $n(z)$ and $d(z)$ **polynomials** in z.

- A **zero** of $F(z)$ is a value of z for which $F(z) = 0$, i.e. a value of z for which $n(z) = 0$.
- A **pole** of $F(z)$ is a value of z for which $F(z) \to \infty$, i.e. value of z for which $d(z) = 0$.

The poles and zeros of $F(z)$ can be **complex**[86].

Remark: By definition, **no poles are included in the ROC**.

[86] They come in **complex conjugate pairs** as we will consider $n(z)$ and $d(z)$ with **real** coefficients.

Region of convergence of a causal signal

- A **causal signal** $f[k]$ of **finite support** $[k_0, k_1]$ where $0 \leq k_0 \leq k \leq k_1 < \infty$ has z-transform

$$F(z) = \mathcal{Z}[f[k]] = \sum_{k=k_0}^{k_1} f[k] z^{-k}$$

  and has a region of convergence which covers **whole** z-plane.

  It is often necessary to **exclude** $z = 0$ from the **region of convergence** depending on the values of k_0 and k_1.

- A **causal signal** $f[k]$ of **infinite support** has a z-transform $F(z)$ with ROC $|z| > r_{max}$ where $r_{max} = \max\{r_i\}$, is the **largest radius** of all poles of $F(z)$, i.e. the region of convergence is **outside a circle of radius** r_{max}.

Definition

F.Y.I.

Two-sided and one-sided z-transforms

The **two-sided** z-transform of a **non causal** signal

$$f[k] = f[k]u[-k] + f[k]u[k] - f[0]$$

can be expressed using the **one-sided** z-transform

$$F(z) = \mathcal{Z}[f[-k]u[k]]|_{(z)} + \mathcal{Z}[f[k]u[k]] - f[0]$$

The **region of convergence** of $F(z)$ is $\mathrm{ROC} = \mathrm{ROC}_c \cap \mathrm{ROC}_{ac}$.
Here ROC_c is the region of convergence of $\mathcal{Z}[f[k]u[k]]$ and ROC_{ac} the region of convergence of $\mathcal{Z}[f[k]u[-k]]$.

Causal and anti-causal decomposition

F.Y.I.

- A **non-causal signal** can be **decomposed** as

$$f[k] = f[k]u[-k] + f[k]u[k] - f[0]$$

- Remember that in discrete-time, $u[1] = 1$.
- The **two-sided** z-transform of $f[k]$ is

$$\begin{aligned} F(z) &= \sum_{-\infty}^{0} f[k]\, z^{-k} + \sum_{0}^{\infty} f[k]\, z^{-k} - f[0] \\ &= \sum_{0}^{\infty} f[-k]\, z^{k} + \sum_{0}^{\infty} f[k]\, z^{-k} - f[0] \\ &= \mathcal{Z}[f[-k]u[k]]|_{(z)} + \mathcal{Z}[f[k]u[k]] - f[0] \end{aligned}$$

- The **two-sided** Laplace transform can be computed using **only** the **one-sided transform** with ROC the **intersection** of the ROCs of the causal and the anti-causal Laplace transforms.

ROC for signals of infinite support

F.Y.I.

For a **signal** $f[k]$ of **infinite support** which

- **anti-causal**, the z-transform has a ROC $|z| < r_{min}$ where $r_{min} = \min\{r_i\}$ is **smallest radius** of the poles of $F(z)$, i.e. the region of convergence is **inside a circle of radius** r_{min}.
- **non-causal**, the z-transform has a ROC $r_{max} < |z| < r_{min}$ where r_{max} and r_{min} are, respectively, the **largest** and **smallest radius** of the poles of $F_c(z)$ and $F_{ac}(z)$, the z-transform of the **causal** and **anti-causal** parts of $f[k]$, i.e. $\text{ROC} = \text{ROC}_c \cap \text{ROC}_{ac}$.

Mapping the s-plane into the z-plane

F.Y.I.

The relation $z = e^{T_s s}$ give a **connection** between the Laplace plane and the z-plane:

$$z = e^{T_s s} = e^{(\sigma + j\omega)T_s} = e^{\sigma T_s} e^{j\omega T_s}$$

One obtains

$$z = r\, e^{j\Omega}$$

which is a **complex variable** in **polar** coordinates, with a **radius** $0 \leq r < \infty$ and **angles** $-\pi \leq \Omega < \pi$ in radians. The variable r corresponds to a **damping factor** and Ω is a **discrete frequency** expressed in radians, i.e.

$$\boxed{r = e^{\sigma T_s}, \Omega = \omega T_s}$$

Mapping the s-plane into the z-plane

F.Y.I.

The relation $z = e^{T_s s}$ maps

- the **real part** of $s = \sigma + j\omega$, $\mathcal{R}_e[s] = \sigma$ into the **radius** $r = e^{\sigma T_s}$ with
  - $0 \leq r < 1$ when $\sigma < 0$,
  - $r = 1$, the **unit circle** $|z| = 1$ when $\sigma = 0$,
  - $r > 1$ when $\sigma > 0$,
- the **analog frequencies** ω on $\Omega = \omega T_s$, i.e.
  - $-\dfrac{\pi}{T_s} \leq \omega \leq \dfrac{\pi}{T_s}$ **on** $-\pi \leq \Omega \leq \pi$,
  - $\dfrac{-\pi + 2k\pi}{T_s} \leq \omega \leq \dfrac{\pi + 2k\pi}{T_s}$ **on** $-\pi + 2k\pi \leq \Omega \leq \pi + 2k\pi$

Mapping the s-plane into the z-plane

F.Y.I.

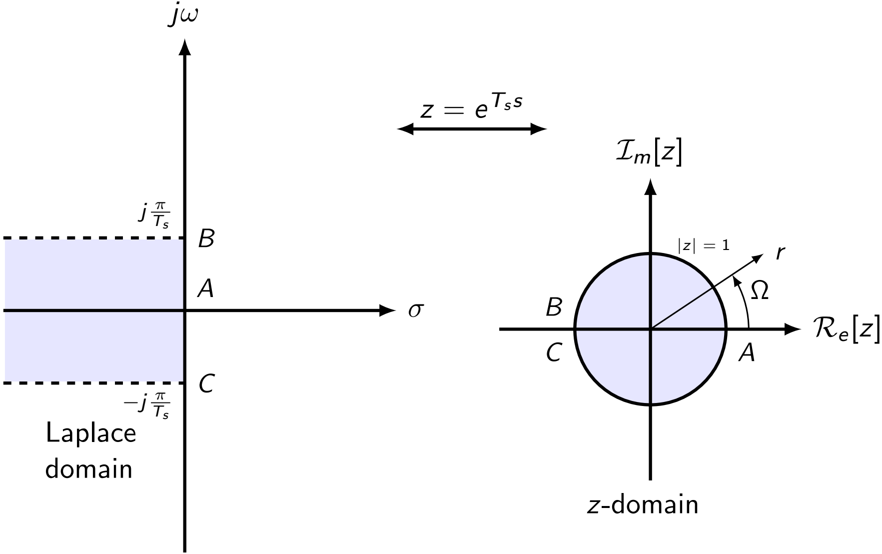
Let us consider **strips** of width $\frac{2\pi}{T_s}$ in the left and right halve planes of the Laplace domain.

The **width** of the strip is of course related to the **Nyquist condition** establishing that the analog signals that we are considering have a **maximum frequency** $\omega_N = \frac{\omega_s}{2} = \frac{\pi}{T_s}$ where T_s is the **sampling period**.

If $T_s \to 0$, we would consider the class of signals with **maximum frequency** approaching ∞, i.e. the entire left and right halve planes.

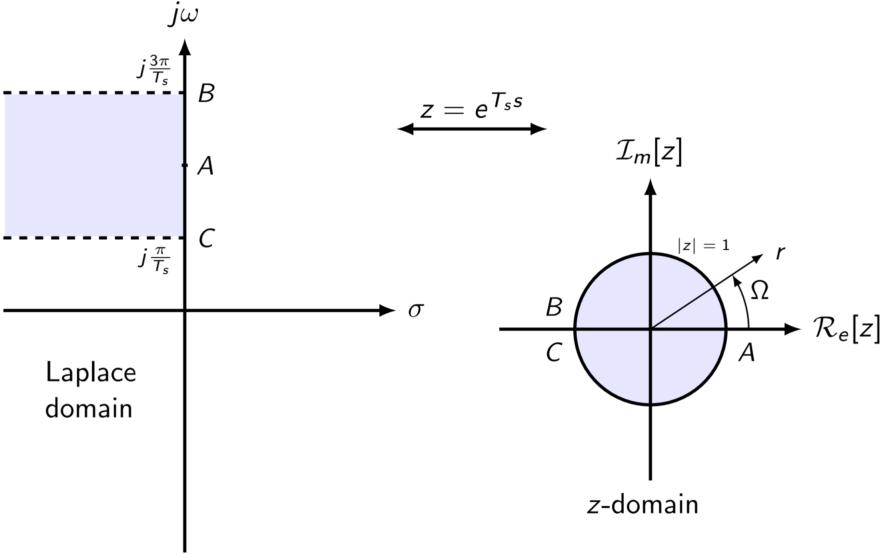
Mapping the s-plane into the z-plane

F.Y.I.



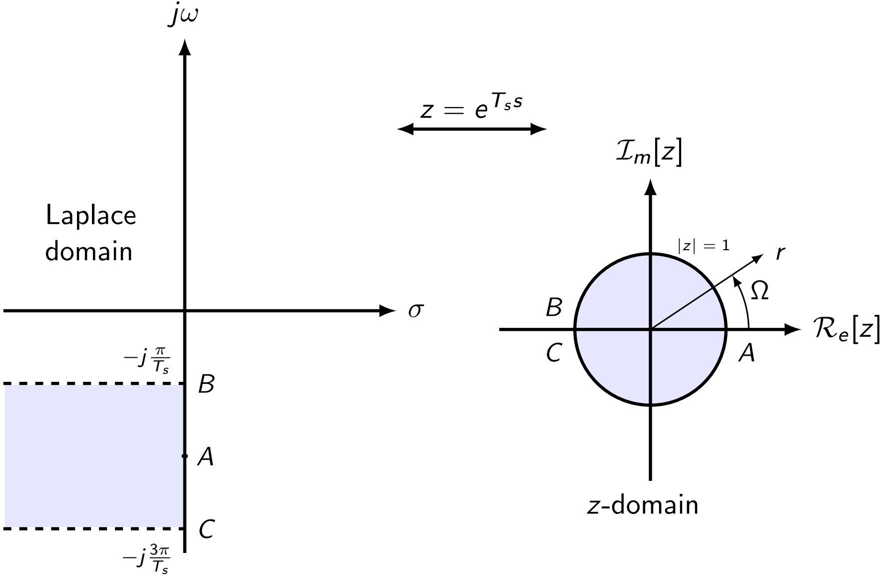
Mapping the s-plane into the z-plane

F.Y.I.



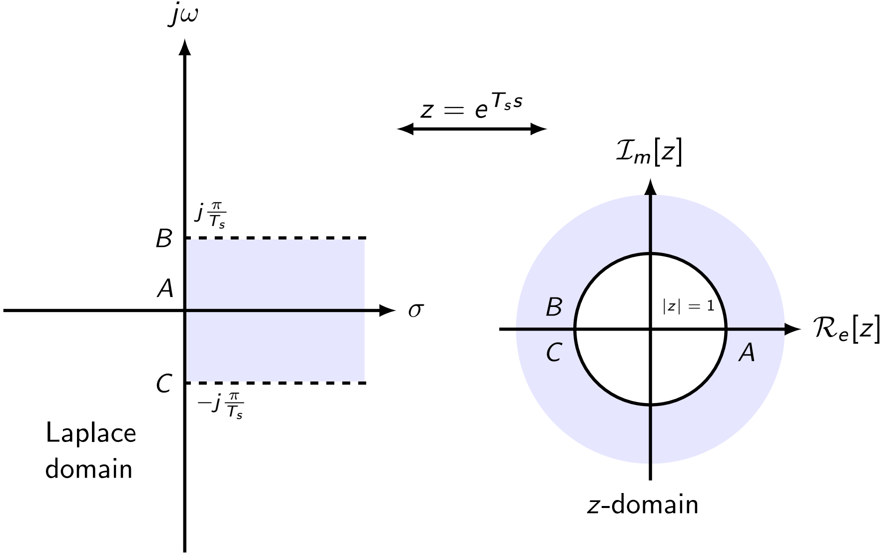
Mapping the s-plane into the z-plane

F.Y.I.



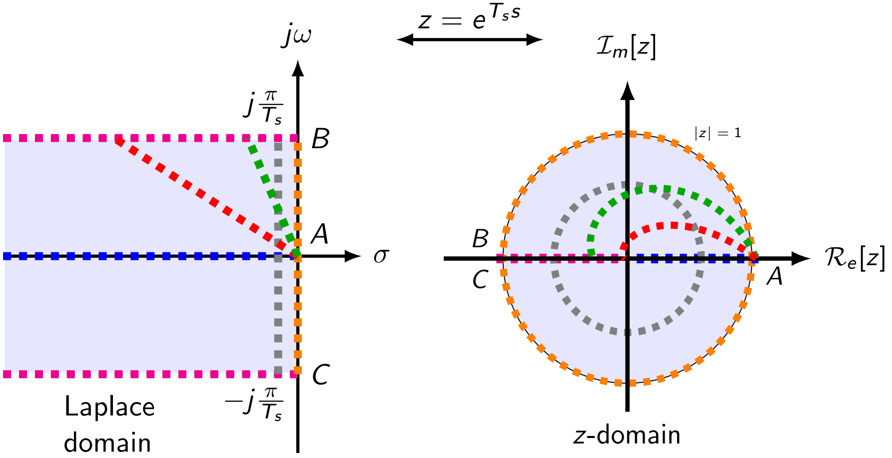
Mapping the s-plane into the z-plane

F.Y.I.



Mapping the s-plane into the z-plane

F.Y.I.



Discrete unit-impulse

The z-transform of a **discrete unit-impulse** $\delta[k]$ is

$$\mathcal{Z}\left[\delta[k]\right] = \sum_{k=0}^{\infty} \delta[k]\, z^{-k} = \sum_{k=0}^{\infty} \delta[k]\, z^{0} = \delta[0] = 1.$$

There are no conditions on z for the sum to converge. The **ROC** is the **whole** z-domain.

Discrete unit-step

The z-transform of a **discrete unit-step** $u[k]$ is

$$\begin{aligned}\mathcal{Z}\left[u[k]\right] &= \sum_{k=0}^{\infty} u[k]\, z^{-k} = \sum_{k=0}^{\infty} z^{-k}, \\ &= \frac{1}{1-z^{-1}}.\end{aligned}$$

The sum converges if $|z| > 1$. The **ROC** is **outside the unit circle** $|z| = 1$.

Discrete complex exponentials

For **discrete complex exponentials** of the form $f[k] = \alpha^k u[k]$

$$\begin{aligned} \mathcal{Z}\left[\alpha^k u[k]\right] &= \sum_{k=0}^{\infty} \alpha^k u[k]\, z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k, \\ &= \frac{1}{1-\alpha z^{-1}}. \end{aligned}$$

The sum converges if $|z| > |\alpha|$. The **ROC** is **outside the circle of radius** $|\alpha|$.

When α is **real** and

- **positive**: the signal is **less and less damped** for α **increasing** in the interval $[0, 1[$. For $\alpha > 1$, the series **diverges**. The value $\alpha = 1$ corresponds a **unit step**.
- **negative**: the signal **alternates sign** at each sample. The envelope of the signal is **less and less damped** for α decreasing in the interval $]-1, 0]$. For $\alpha < -1$, the series **diverges**.

Discrete complex exponentials

F.Y.I.

To compute the z-transform of $f[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$, the **Euler identity** is used to obtain

$$f[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k] = \frac{1}{2}\left[r_0^k\, e^{j(\Omega_0 k + \phi)} + r_0^k\, e^{-j(\Omega_0 k + \phi)}\right] u[k]$$

Using the **linearity** property[87] and the z-transform of $\alpha^k u[k]$ with $\alpha = r_0\, e^{j\Omega_0}$ and its complex conjugate $\alpha^* = r_0\, e^{-j\Omega_0}$, we obtain

$$\begin{aligned} F(z) &= \frac{1}{2}\left[\frac{z\, e^{j\phi}}{z - r_0\, e^{j\Omega_0}} + \frac{z\, e^{-j\phi}}{z - r_0\, e^{-j\Omega_0}}\right] \\ &= \frac{z\,(z\, \cos(\phi) - r_0\, \cos(\Omega_0 - \phi))}{(z - r_0\, e^{j\Omega_0})(z - r_0\, e^{-j\Omega_0})} \end{aligned}$$

with ROC the **outside of the circle** of radius r_0, i.e. $|z| > r_0$.

[87] The linearity of the z-transformed is introduced subsequently

Discrete complex exponentials

F.Y.I.

The z-transform of the **complex exponential signal**

$$\boxed{f[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]}$$

is

$$\boxed{F(z) = \frac{z\,(z\,\cos(\phi) - r_0\,\cos(\Omega_0 - \phi))}{(z - r_0\,e^{j\Omega_0})(z - r_0\,e^{-j\Omega_0})}}$$

with **poles** $\boxed{r_0\,e^{\pm j\Omega_0}}$ and ROC $|z| > r_0$.

In particular,

- $\mathcal{Z}[\cos(\Omega_0 k)u[k]] = \dfrac{z(z - \cos(\Omega_0))}{(z - e^{j\Omega_0})(z - e^{-j\Omega_0})} \quad (\phi = 0,\ r_0 = 1)$
- $\mathcal{Z}[\sin(\Omega_0 k)u[k]] = \dfrac{z\sin(\Omega_0)}{(z - e^{j\Omega_0})(z - e^{-j\Omega_0})} \quad (\phi = -\frac{\pi}{2},\ r_0 = 1)$

with ROC the **outside of the unit circle**, i.e. $|z| > 1$.

Linearity

Linearity

For the signals $f[k]$ and $g[k]$ with z-transform

- $\mathcal{Z}[f[k]] = F(z)$,
- $\mathcal{Z}[g[k]] = G(z)$ and
- constants α and $\beta \in \mathbb{C}$

we have

$$\mathcal{Z}[\alpha f[k] + \beta g[k]] = \alpha\, \mathcal{Z}[f[k]] + \beta\, \mathcal{Z}[g[k]] = \alpha F(z) + \beta G(z)$$

with ROC that is the **intersection** of the regions of convergence of $\mathcal{Z}[f[k]]$ and $\mathcal{Z}[g[k]]$.

Time shifting

Time shifting

For a signal $f[k]$ with z-transform $\mathcal{Z}[f[k]u[k]] = F(z)$ and $k_0 > 0$, we have

$$\begin{aligned}\mathcal{Z}[f[k-k_0]u[k]] &= \sum_{k=0}^{\infty} f[k-k_0]z^{-k} = \sum_{n=-k_0}^{\infty} f[n]z^{-(n+k_0)} \\ &= z^{-k_0}\sum_{n=0}^{\infty} f[n]z^{-n} + \sum_{n=-k_0}^{-1} f[n]z^{-(n+k_0)}\end{aligned}$$

with the same ROC as $\mathcal{Z}[f[k]u[k]]$ and **additional restrictions** in $z = 0$.
With **zero initial conditions**, we have $\mathcal{Z}[f[k-k_0]u[k]] = z^{-k_0}F(z)$.

Time shifting

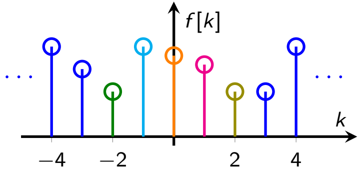
Time shifting

For a signal $f[k]$ with z-transform $\mathcal{Z}[f[k]u[k]] = F(z)$ and $k_0 > 0$, we have a:

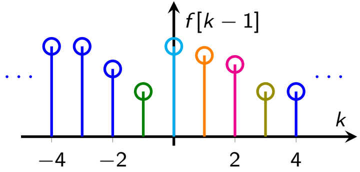
$$\begin{aligned}\mathcal{Z}[f[k+k_0]u[k]] &= \sum_{k=0}^{\infty} f[k+k_0]z^{-k} = \sum_{n=k_0}^{\infty} f[n]z^{-n}z^{k_0} \\ &= z^{k_0}\sum_{n=0}^{\infty} f[n]z^{-n} - \sum_{n=0}^{k_0-1} f[n]z^{-n}z^{k_0}\end{aligned}$$

with the same ROC as $\mathcal{Z}[f[k]u[k]]$ and **additional restrictions** $z = \pm\infty$.
With **zero initial conditions**, we have $\mathcal{Z}[f[k+k_0]u[k]] = z^{k_0}F(z)$.

Time shifting

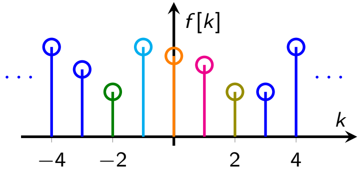


$$F(z) = \mathcal{Z}[f[k]u[k]] = \sum_{k=0}^{\infty} f[k]\, z^{-k} = f[0] + f[1]\, z^{-1} + f[2]\, z^{-2} + \cdots$$

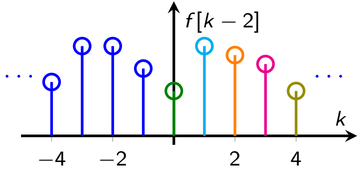


$$\begin{aligned} \mathcal{Z}[f[k-1]u[k]] &= \sum_{k=0}^{\infty} f[k-1]\, z^{-k} = f[-1] + f[0]\, z^{-1} + f[1]\, z^{-2} + f[2]\, z^{-3} + \cdots \\ &= z^{-1} F(z) + f[-1] \end{aligned}$$

Time shifting

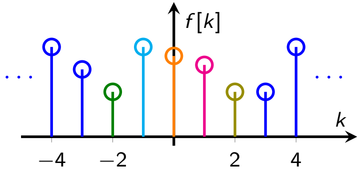


$$F(z) = \mathcal{Z}[f[k]u[k]] = \sum_{k=0}^{\infty} f[k]\, z^{-k} = f[0] + f[1]\, z^{-1} + f[2]\, z^{-2} + \cdots$$

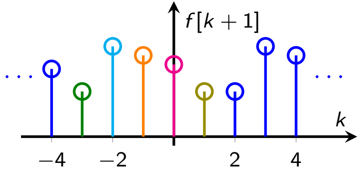


$$\begin{aligned}\mathcal{Z}[f[k-2]u[k]] &= \sum_{k=0}^{\infty} f[k-2]\, z^{-k} = f[-2] + f[-1]\, z^{-1} + f[0]\, z^{-2} + f[1]\, z^{-3} + \cdots \\ &= z^{-2}\, F(z) + f[-1]\, z^{-1} + f[-2]\end{aligned}$$

Time shifting

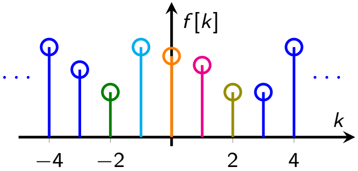


$$F(z) = \mathcal{Z}[f[k]u[k]] = \sum_{k=0}^{\infty} f[k]\,z^{-k} = f[0] + f[1]\,z^{-1} + f[2]\,z^{-2} + \cdots$$

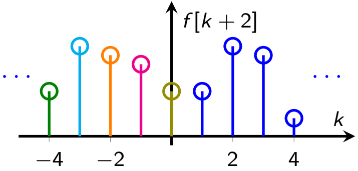


$$\begin{aligned}\mathcal{Z}[f[k+1]u[k]] &= \sum_{k=0}^{\infty} f[k+1]\,z^{-k} = f[1] + f[2]\,z^{-1} + f[3]\,z^{-2} + f[4]\,z^{-3}\cdots \\ &= z\,F(z) - f[0]\,z\end{aligned}$$

Time shifting



$$F(z) = \mathcal{Z}[f[k]u[k]] = \sum_{k=0}^{\infty} f[k]\, z^{-k} = f[0] + f[1]\, z^{-1} + f[2]\, z^{-2} + \cdots$$



$$\begin{aligned} \mathcal{Z}[f[k+2]u[k]] &= \sum_{k=0}^{\infty} f[k+2]\, z^{-k} = f[2] + f[3]\, z^{-1} + f[4]\, z^{-2} + f[5]\, z^{-3} \cdots \\ &= z^2 F(z) - f[0]\, z^2 - f[1]\, z \end{aligned}$$

Time shifting: summary

Suppose $F(z) = \mathcal{Z}[f[k]u[k]]$ then

- $\mathcal{Z}[f[k-1]u[k]] = z^{-1}\,F(z) + f[-1]$
- $\mathcal{Z}[f[k-2]u[k]] = z^{-2}\,F(z) + f[-1]\,z^{-1} + f[-2]$
- $\mathcal{Z}[f[k+1]u[k]] = z\,F(z) - f[0]\,z$
- $\mathcal{Z}[f[k+2]u[k]] = z^{2}\,F(z) - f[0]\,z^{2} - f[1]\,z$

Unit delay operator

From the **time shifting property**, we know that, with **zero initial conditions**, we have

$$\mathcal{Z}[f[k-k_0]u[k]] = z^{-k_0}F(z).$$

Suppose the system with **transfer function** $H(z) = z^{-1}$ and an **input signal** $x[k]$ with associated z-transform $X(z)$.

The **response** is easily obtained in the z-domain, i.e.

$$Y(z) = z^{-1}X(z)$$

with associated **time response** $y[k] = x[k-1]$. The operator z^{-1} is therefore the **unit delay operator**.

Derivative of $F(z)$

Derivative of $F(z)$

For a **signal** $f[k]$ with z-**transform** $F(z) = \mathcal{Z}[f[k]u[k]]$, the **derivative** of $F(z)$ with respect to z is

$$\frac{dF(z)}{dz} = \sum_{k=0}^{\infty} f[k]\frac{dz^{-k}}{dz} = -z^{-1}\sum_{k=0}^{\infty} k\, f[k]z^{-k}.$$

One obtains

$$\boxed{\mathcal{Z}\left[k\, f[k]u[k]\right] = -z\,\frac{dF(z)}{dz}}$$

From the property of the **derivative** of $F(z)$, we can **calculate** the following **transform**

$$\mathcal{Z}\left[k\,\alpha^k u[k]\right] = -z\frac{d}{dz}\left(\frac{1}{1-\alpha z^{-1}}\right) = -z\frac{d}{dz}\left(\frac{z}{z-\alpha}\right) = \frac{\alpha z}{(z-\alpha)^2}$$

Z-scaling / damping

F.Y.I.

Z-scaling / damping

For a **signal** $f[k]$ with z-**transform** $F(z) = \mathcal{Z}[f[k]u[k]]$, we have

$$\begin{aligned} \mathcal{Z}[\alpha^k f[k]u[k]] &= \sum_{k=0}^{\infty} \alpha^k f[k] z^{-k}, \\ &= \sum_{k=0}^{\infty} f[k] \left(\frac{z}{\alpha}\right)^{-k} = \sum_{k=0}^{\infty} f[k] \left(\alpha z^{-1}\right)^k, \\ &= F\left(\frac{z}{\alpha}\right) = F(\alpha z^{-1}). \end{aligned}$$

The **radius** defining the **ROC** is multiplied by $|\alpha|$.

Z-scaling / damping: an example

F.Y.I.

Remember that

$$\mathcal{Z}\left[u[k]\right] = \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1-z^{-1}} = \frac{z}{z-1}$$

with **ROC outside the unit circle** $|z| = 1$.

The z-**scaling property** states that for a **real** α

$$\mathcal{Z}\left[\alpha^k\, u[k]\right] = \frac{1}{1-\alpha\, z^{-1}} = \frac{\frac{z}{\alpha}}{\frac{z}{\alpha}-1} = \frac{z}{z-\alpha}$$

with **ROC outside the unit circle** $|z| = \alpha$. This is **consistent** with previous results.

Convolution sum and z-transform

Convolution sum and z-transform

The z-transform of the **convolution** of the signal $f[k]$ with $\mathcal{Z}[f[k]] = F(z)$ and the signal $g[k]$ with $\mathcal{Z}[g[k]] = G(z)$ is

$$\mathcal{Z}\left[f[k] * g[k]\right] = F(z)G(z).$$

Corollary

The **response** of a LTI system with **impulse response** $h[k]$ to a **causal signal** $x[k]$ is given by

$$y[k] = \mathcal{Z}^{-1}[Y(z)] = \mathcal{Z}^{-1}[H(z)X(z)] = \mathcal{Z}^{-1}\left[\mathcal{Z}\left[h[k] * x[k]\right]\right].$$

where $\mathcal{Z}[h[k]] = H(z)$ and $\mathcal{Z}[x[k]] = X(z)$.
In the z-domain, we have $Y(z) = H(z)X(z)$.

Convolution sum and z-transform

Convolution

Impulse response $h[k]$
Input signal $x[k]$

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

$\mathcal{Z}$

$\mathcal{Z}^{-1}$

Transfer function $H(z)$
Z-transform of input signal $X(z)$

Z-transform of output signal
$Y(z) = H(z)X(z)$

Convolution

Transfer function

Transfer function

The **transfer function** $H(z) = \mathcal{Z}[h[k]]$, the z-transform of the **impulse response** $h[k]$ of a LTI system, can be expressed as the **ratio**

$$H(z) = \frac{\mathcal{Z}[y[k]]}{\mathcal{Z}[x[k]]} = \frac{\mathcal{Z}[\text{output signal}]}{\mathcal{Z}[\text{input signal}]}$$

The **transfer function** characterizes the **system** by its **poles** and **zeros**.

It is an **important tool** in the **analysis** and **design** of systems.

Difference equations and IIR: an example (2)

Let us take the simplest **difference equation** with **zero initial conditions**, i.e.

$$y[k] = -a\,y[k-1] + b\,x[k]$$

Going to the z-domain, we obtain

$$(1 + a\,z^{-1})\,Y(z) = b\,X(z)$$

This means that the **transfer function** is

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response**, the **inverse z-transform** of $H(z)$ is

$$h[k] = (-a)^k\,b\,u[k]$$

The system described by this difference equation has an **infinite impulse reponse**.

Stability in the z-domain

Reminder:

- The **ROC** of the z-**transform** of the **impulse response** of a **causal** system, i.e. its transfer function, is

$$\boxed{\text{ROC} \;=\left\{ z = r\,e^{j\Omega} \text{ such that } \sum_{k=0}^{\infty} |h[k]\,r^{-k}| < \infty \right\}}$$

- A **causal** LTI system is **BIBO stable** if and only if

$$\boxed{\sum_{k=0}^{\infty} |h[k]| < \infty}$$

Note that

- If the **ROC** includes the **unit circle**, i.e. $|z| = r = 1$ in ROC, then the system is **BIBO stable**.
- If the system is **BIBO stable** then the **ROC** includes the **unit circle**, i.e. $|z| = r = 1$ in ROC.

Stability in the z-domain

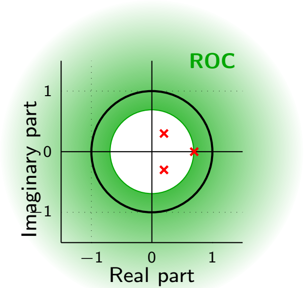
Stability in the z-domain

The discrete-time **causal** LTI system is **stable** if it can be **described** by a **transfer function** $H(z)$ with **ROC** that **includes the unit circle**.

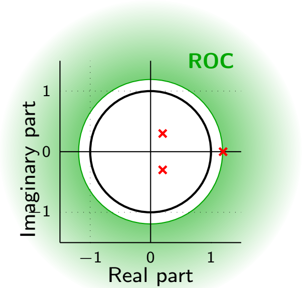
For a causal system, the condition is equivalent to requesting that the poles $H(z)$ are located strictly inside the unit circle.

Stability in the z-domain: illustration for a causal system

Assuming a **causal** system, the **ROC** of the transfer function $H(z)$ is $|z| > r_{max}$ where r_{max} is the **largest radius** of the poles of $H(z)$.
The ROC extends **outward** from the pole with the **largest magnitude**.



Stable system



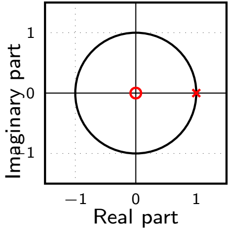
Unstable system

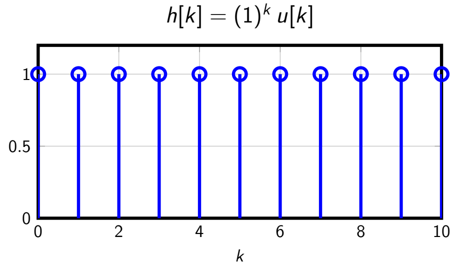
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\begin{aligned}\mathcal{Z}[\alpha^k u[k]] &= \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k \\ &= \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}\end{aligned}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.



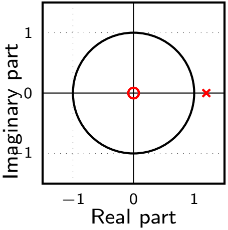


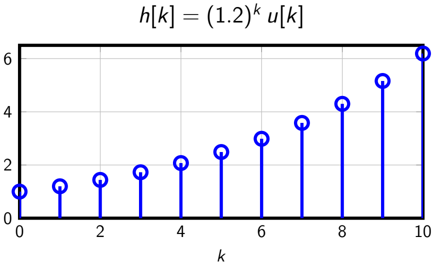
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\mathcal{Z}[\alpha^k u[k]] = \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k$$
$$= \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.



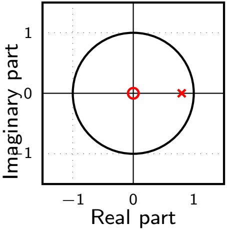


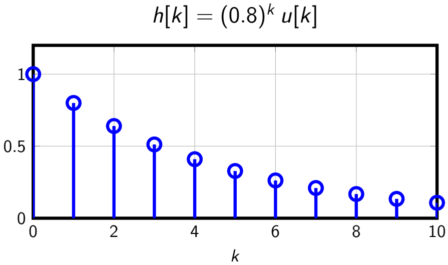
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\mathcal{Z}[\alpha^k u[k]] = \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k = \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.



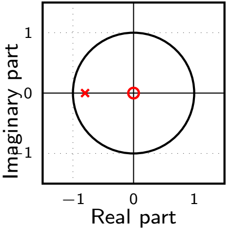


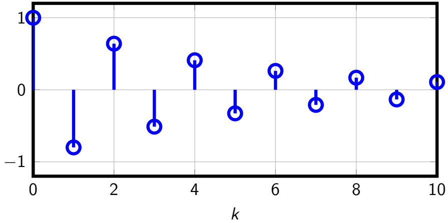
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\mathcal{Z}[\alpha^k u[k]] = \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k$$
$$= \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.



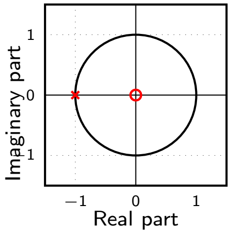


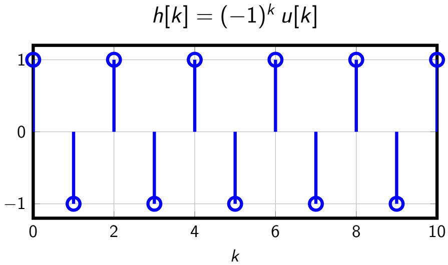
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\begin{aligned}\mathcal{Z}[\alpha^k u[k]] &= \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k \\ &= \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}\end{aligned}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.



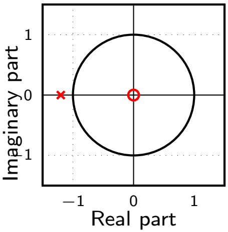


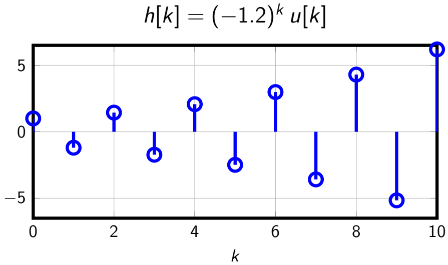
Dynamic behaviour: real exponentials

Discrete real exponentials $h[k] = \alpha^k u[k]$ with α **real**

$$\mathcal{Z}[\alpha^k u[k]] = \sum_{k=0}^{\infty} \alpha^k u[k] z^{-k} = \sum_{k=0}^{\infty} (\alpha z^{-1})^k$$
$$= \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}$$

which has a single real pole at $z = \alpha$, plus a zero at $z = 0$. We have the following **correspondence** between **pole location** and **impulse response**.





Dynamic behaviour: complex exponentials

F.Y.I.

To compute the z-transform of $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$, the **Euler identity** is used to obtain

$$h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k] = \frac{1}{2}\left[r_0^k\, e^{j(\Omega_0 k+\phi)} + r_0^k\, e^{-j(\Omega_0 k+\phi)}\right] u[k]$$

Using the **linearity** property and the z-transform of $\alpha^k u[k]$ with $\alpha = r_0\, e^{j\Omega_0}$ and its complex conjugate $\alpha^* = r_0\, e^{-j\Omega_0}$, we obtain

$$\begin{aligned} H(z) &= \frac{1}{2}\left[\frac{z\, e^{j\phi}}{z - r_0\, e^{j\Omega_0}} + \frac{z\, e^{-j\phi}}{z - r_0\, e^{-j\Omega_0}}\right] \\ &= \frac{z\,(z\, \cos(\phi) - r_0\, \cos(\Omega_0 - \phi))}{(z - r_0\, e^{j\Omega_0})(z - r_0\, e^{-j\Omega_0})} \end{aligned}$$

The **poles** are **complex conjugate** $p_{12} = r_0\, e^{j\pm\Omega_0}$ and are **unaffected** by **phase changes** ϕ.

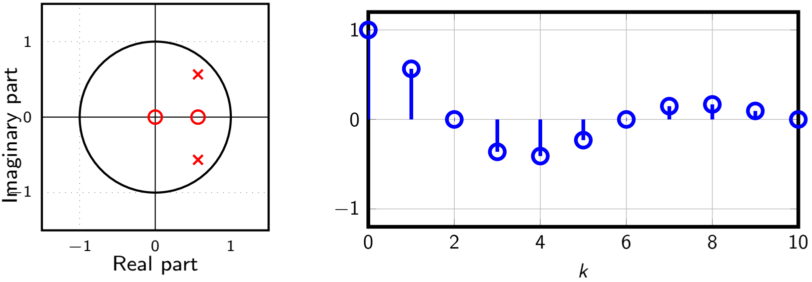
The **zeros** are **real** $z_1 = 0$ and $z_2 = \frac{r_0\, \cos(\Omega_0 - \phi)}{\cos(\phi)}$.

Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$

$h[k] = (0.8)^k \cos(\frac{\pi}{4} k)\, u[k]$

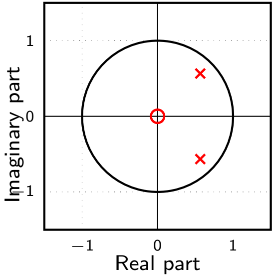


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.8.$

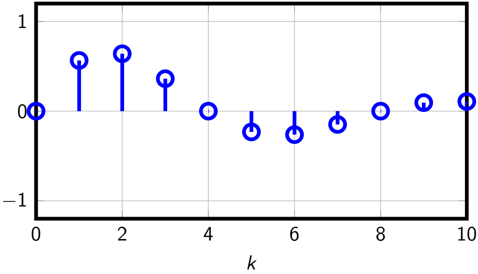
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \sin(\frac{\pi}{4}k)\, u[k]$

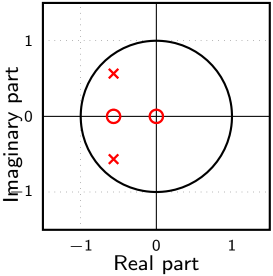


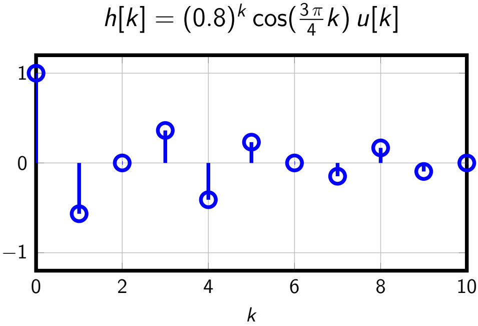
with $\Omega_0 = \frac{\pi}{4}, \phi = -\frac{\pi}{2}, r_0 = 0.8.$

Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



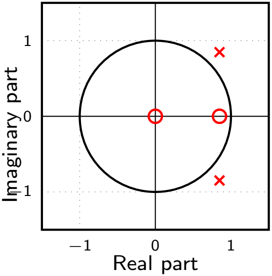


with $\Omega_0 = \frac{3\pi}{4}, \phi = 0, r_0 = 0.8.$

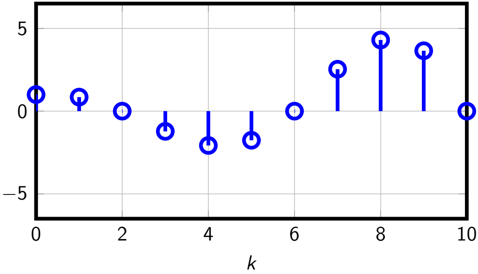
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (1.2)^k \cos(\frac{\pi}{4}k)\, u[k]$

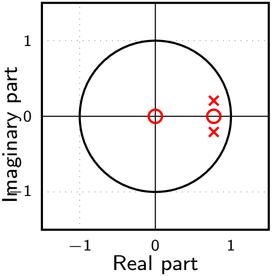


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 1.2.$

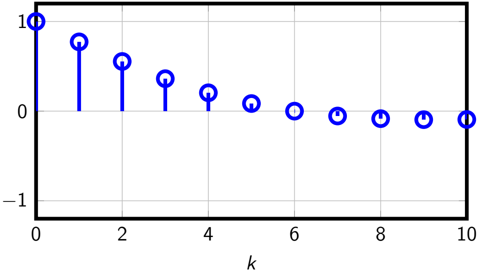
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \cos(\frac{\pi}{12}k)\, u[k]$

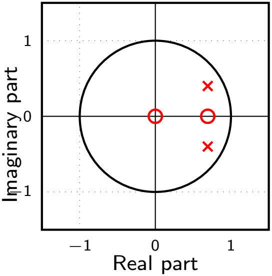


with $\Omega_0 = \frac{\pi}{12}, \phi = 0, r_0 = 0.8.$

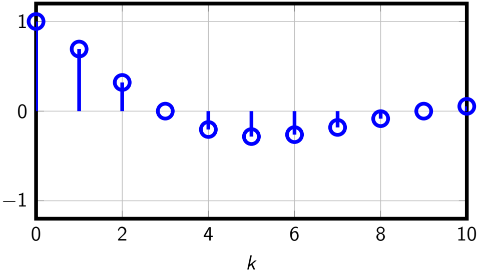
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \cos(\frac{\pi}{6} k)\, u[k]$

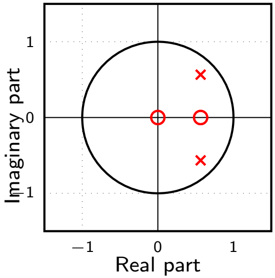


with $\Omega_0 = \frac{\pi}{6}, \phi = 0, r_0 = 0.8$.

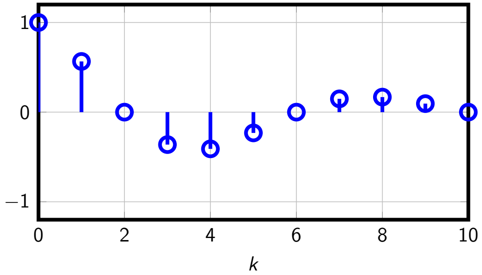
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \cos(\frac{\pi}{4} k)\, u[k]$

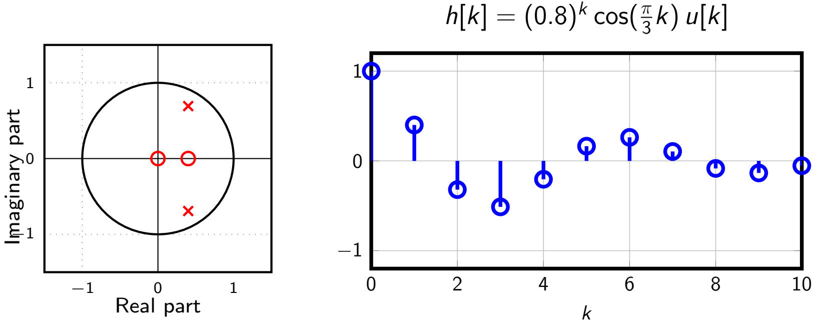


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.8.$

Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$

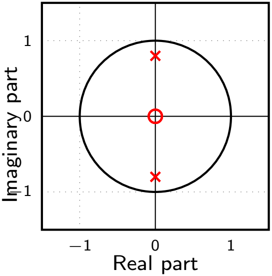


with $\Omega_0 = \frac{\pi}{3}, \phi = 0, r_0 = 0.8.$

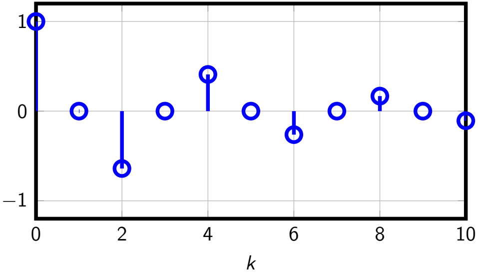
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \cos(\frac{\pi}{2} k)\, u[k]$

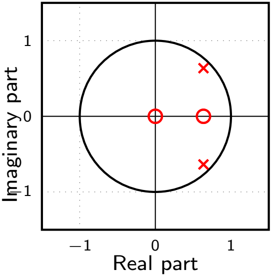


with $\Omega_0 = \frac{\pi}{2}, \phi = 0, r_0 = 0.8.$

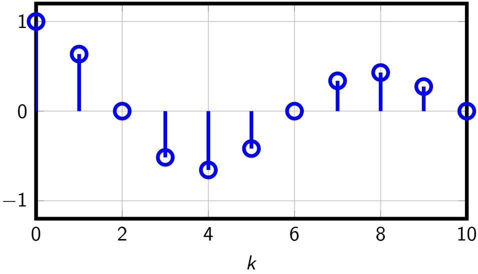
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.9)^k \cos(\frac{\pi}{4} k)\, u[k]$

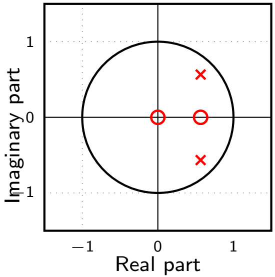


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.9$.

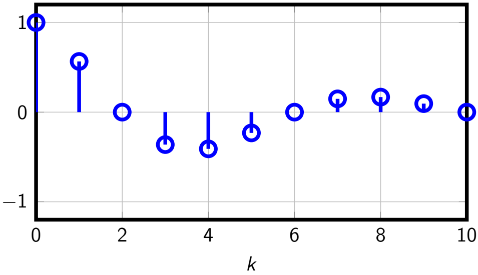
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.8)^k \cos(\frac{\pi}{4} k)\, u[k]$

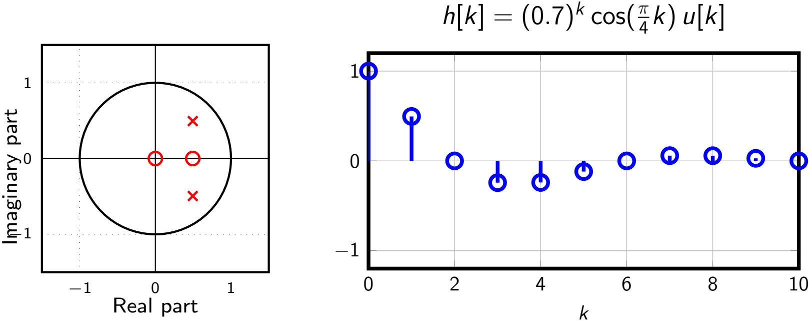


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.8.$

Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$

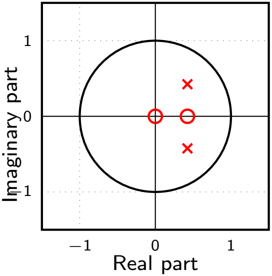


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.7$.

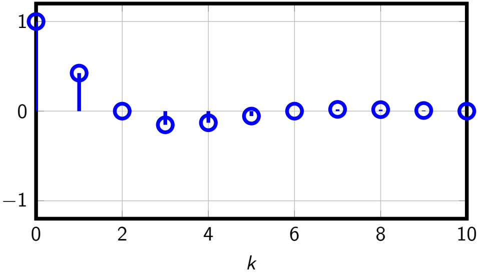
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$$h[k] = (0.6)^k \cos(\tfrac{\pi}{4}k)\, u[k]$$

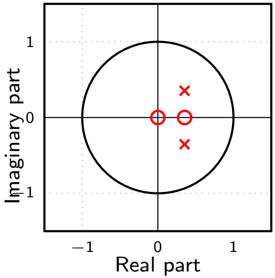


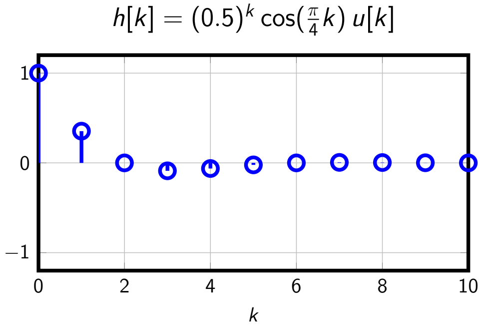
with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.6.$

Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



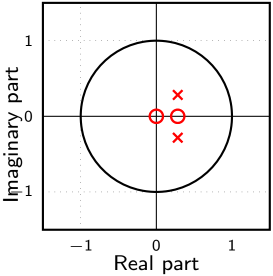


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.5.$

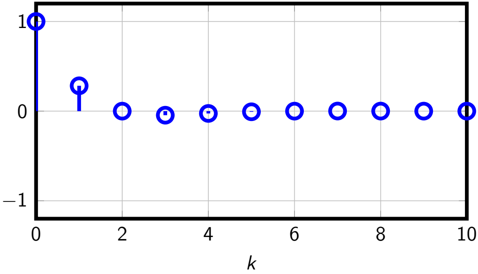
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.4)^k \cos(\frac{\pi}{4} k)\, u[k]$

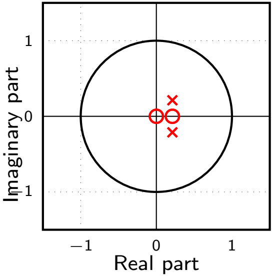


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.4$.

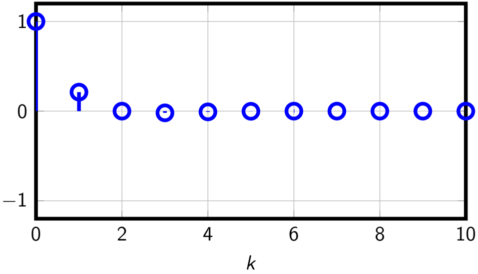
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.3)^k \cos(\frac{\pi}{4} k)\, u[k]$

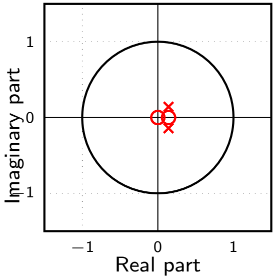


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.3.$

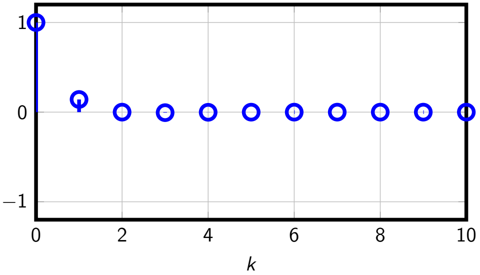
Dynamic behaviour: complex exponentials

F.Y.I.

Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



$h[k] = (0.2)^k \cos(\frac{\pi}{4} k)\, u[k]$

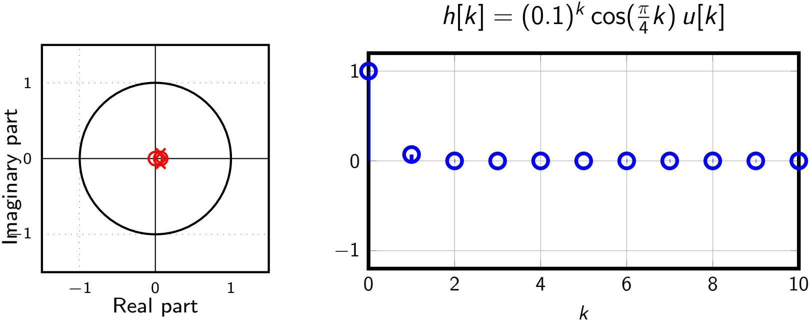


with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.2.$

Dynamic behaviour: complex exponentials

F.Y.I.

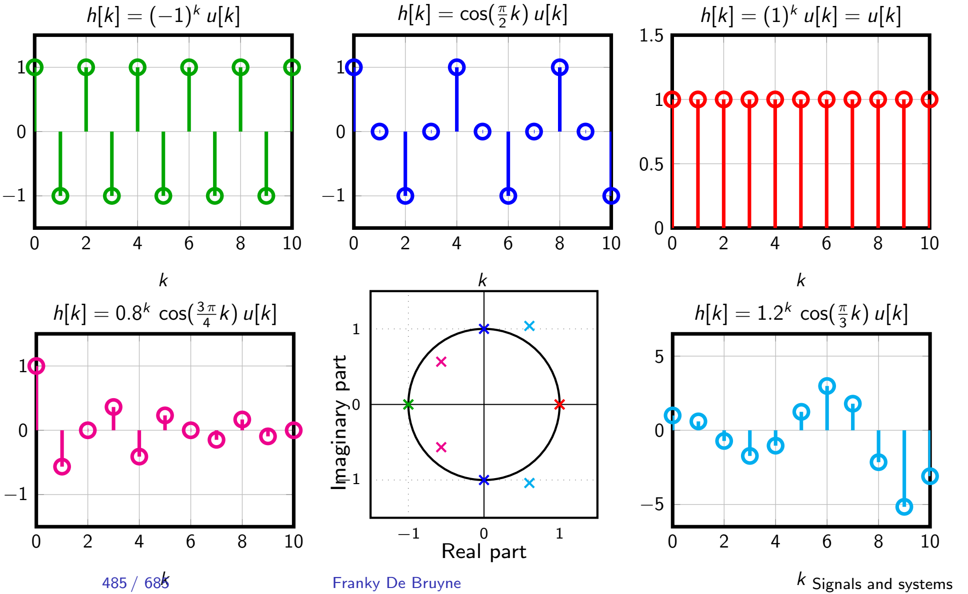
Discrete complex exponential $h[k] = r_0^k \cos(\Omega_0 k + \phi)\, u[k]$



with $\Omega_0 = \frac{\pi}{4}, \phi = 0, r_0 = 0.1$.

Dynamic behaviour

F.Y.I.



Initial and final value theorems

In some **control applications** and to **check** a **partial fraction expansion**, it is useful to find the **initial** and **final** value of a discrete-time signal $y[k]$ from its z-transform $Y(z)$.

Initial and final value theorem

From $Y(z)$, it is possible to determine the **initial** and **final** value of the signal $y[k]$, in the vicinity of $k = 0$ and $k \to \infty$.

- **Initial value**: providing the limit **exists**,

$$y[0] = \lim_{z \to \infty} Y(z).$$

- **Final value**: if all the poles of $(z-1)\,Y(z)$ are **inside** the **unit circle** then

$$\lim_{k \to \infty} y[k] = \lim_{z \to 1} (z-1)\,Y(z).$$

Static gain

The **static gain** of a system described by the **transfer function** $H(z)$ is obtained by evaluating the transfer function at $z = 1$, i.e.

$$\boxed{\textbf{Static gain} \text{ of the system } H(z) \text{ is } H(z)|_{z=1} = H(1)}$$

The **static gain** tells the **ratio** of the **output** and the **input** under **steady state condition**.

If the input $x[k] = x_0$ is **constant** and the system is **stable** then the output will reach the **steady state value**

$$y_0 = H(z)|_{z=1}\, x_0$$

when **all transients have disappeared**.

The transfer function can thus be viewed as a **generalization** of the concept of gain.

Inverse z-transform: long division method

F.Y.I.

Long division

A **rational transfer function** $H(z)$ of a **causal** system can be written

$$H(z) = h[0] + h[1]\, z^{-1} + h[2]\, z^{-2} + \cdots$$

This description can be obtained by the **long division method**.

The **inverse** z-transform is given by

$$h[k] = h[0]\, \delta[k] + h[1]\, \delta[k-1] + h[2]\, \delta[k-2] + \cdots$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response** can also be found by **long division**, i.e.

$$b \quad \Big| \quad \underline{1 + a\,z^{-1}}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrcl|l}
 & b & & & \;1 + a\,z^{-1} \\ \cline{5-5}
- & b & + & ab\,z^{-1} & \;b \\ \cline{2-4}
 & & - & ab\,z^{-1} &
\end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrrrr|l}
 & b & & & & 1 + a\,z^{-1} \\ \cline{6-6}
- & b & + \; ab\,z^{-1} & & & b - ab\,z^{-1} \\ \cline{2-3}
 & & - \; ab\,z^{-1} & & & \\
- & & - \; ab\,z^{-1} & - \; a^2 b\,z^{-2} & & \\ \cline{3-4}
 & & & + \; a^2 b\,z^{-2} & &
\end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response** can also be found by **long division**, i.e.

$$
\begin{array}{rrrrr|l}
 & b & & & & 1 + a\,z^{-1} \\
\cline{6-6}
- & b & +\;ab\,z^{-1} & & & b - ab\,z^{-1} + a^2b\,z^{-2} - \dots \\
\cline{2-3}
 & & -\;ab\,z^{-1} & & & \\
- & & -\;ab\,z^{-1} & -\;a^2b\,z^{-2} & & \\
\cline{3-4}
 & & & +\;a^2b\,z^{-2} & & \\
 & & & \vdots & &
\end{array}
$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b}{1 + a\,z^{-1}}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrrr|l}
 & b & & & 1 + a\,z^{-1} \\ \cline{5-5}
- & b + & ab\,z^{-1} & & b - ab\,z^{-1} + a^2 b\,z^{-2} - \dots \\ \cline{2-3}
 & & - \; ab\,z^{-1} & & \\
- & & - \; ab\,z^{-1} & - \; a^2 b\,z^{-2} & \\ \cline{3-4}
 & & & + \; a^2 b\,z^{-2} & \\
 & & & \vdots &
\end{array}$$

The system has **impulse response**

$$h[k] = b\,\delta[k] - ab\,\delta[k-1] + a^2 b\,\delta[k-2] - \cdots = (-a)^k\,b\,u[k].$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b\,z}{z+a}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{l|l} b\,z & z+a \\ \hline & \end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b\,z}{z+a}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrcr|l}
 & b\,z & & & \quad z+a \\ \cline{5-5}
- & b\,z & + & ab & \quad b \\ \cline{2-4}
 & & - & ab &
\end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b\,z}{z+a}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrrrr|l}
 & b\,z & & & & z+a \\
\cline{6-6}
- & b\,z & + & ab & & b - ab\,z^{-1} \\
\cline{2-4}
 & & - & ab & & \\
 & - & - & ab & - \; a^2 b\,z^{-1} & \\
\cline{3-5}
 & & & & + \; a^2 b\,z^{-1} &
\end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b\,z}{z+a}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rllll|l}
 & b\,z & & & & z+a \\ \cline{6-6}
- & b\,z & + & ab & & b - ab\,z^{-1} + a^2b\,z^{-2} - \ldots \\ \cline{2-4}
 & & - & ab & & \\
 & & - & ab & - \; a^2b\,z^{-1} & \\ \cline{3-5}
 & & & & + \; a^2b\,z^{-1} & \\
 & & & & \vdots &
\end{array}$$

Difference equations and IIR: an example (3)

F.Y.I.

The system described by the **difference equation**

$$y[k] = -a\,y[k-1] + b\,x[k]$$

has **transfer function**

$$H(z) = \frac{b\,z}{z+a}$$

The **impulse response** can also be found by **long division**, i.e.

$$\begin{array}{rrrrr|l}
 & b\,z & & & & z + a \\ \cline{6-6}
- & b\,z & + & ab & & b - ab\,z^{-1} + a^2 b\,z^{-2} - \dots \\ \cline{2-4}
 & & - & ab & & \\
 & - & & ab & - \; a^2 b\,z^{-1} & \\ \cline{2-5}
 & & & & + \; a^2 b\,z^{-1} & \\
 & & & & \vdots &
\end{array}$$

The system has **impulse response**

$$h[k] = b\,\delta[k] - ab\,\delta[k-1] + a^2 b\,\delta[k-2] - \dots = (-a)^k b\,u[k].$$

Inverse z-transform: partial fraction expansion

Partial fraction expansions

The **basics** of partial fraction expansion remain the **same** for the z-transform as for the Laplace transform:

- The transfer function must be **strictly proper**, i.e. the degree of its numerator must be smaller than the degree of its denominator. If **needed**, perform a **Euclidean division** until the condition is satisfied.
- Expand in partial fractions after computing the **poles** of the transfer function.
- The expansion can be performed in z or z^{-1}.
- The resulting expansion is composed of terms of which the inverse z-transform can **easily** be obtained from z-transform tables.

Partial fraction expansion in z^{-1} of $Y(z^{-1})$: example

Find the **inverse** z-transform of

$$Y(z) = \frac{1+z^{-1}}{(1+0.5z^{-1})(1-0.5z^{-1})} = \frac{z(z+1)}{(z+0.5)(z-0.5)} \quad |z| > 0.5$$

Expanding in partial fractions using **negative powers** z^{-1}, we obtain

$$Y(z) = \frac{A}{(1+0.5z^{-1})} + \frac{B}{(1-0.5z^{-1})}$$

$$A = \left.\frac{1+z^{-1}}{1-0.5z^{-1}}\right|_{z^{-1}=-2} = -\frac{1}{2}, \quad B = \left.\frac{1+z^{-1}}{1+0.5z^{-1}}\right|_{z^{-1}=2} = \frac{3}{2}$$

$$y[k] = [1.5\,(0.5)^k - 0.5\,(-0.5)^k]\,u[k].$$

Partial fraction expansion in z of $Y(z)/z$: example

Find the **inverse** z-transform of

$$Y(z) = \frac{1+z^{-1}}{(1+0.5z^{-1})(1-0.5z^{-1})} = \frac{z(z+1)}{(z+0.5)(z-0.5)} \quad |z| > 0.5$$

Expanding in partial fractions using **positive powers** z of $Y(z)/z$, we obtain

$$\frac{Y(z)}{z} = \frac{A}{(z+0.5)} + \frac{B}{(z-0.5)} \Rightarrow Y(z) = \frac{A\,z}{(z+0.5)} + \frac{B\,z}{(z-0.5)}$$

$$A = \left.\frac{z+1}{z-0.5}\right|_{z=-0.5} = -\frac{1}{2}, \quad B = \left.\frac{z+1}{z+0.5}\right|_{z=0.5} = \frac{3}{2}$$

$$y[k] = [1.5\,(0.5)^k - 0.5\,(-0.5)^k]\,u[k].$$

Partial fraction expansion in z of $Y(z)$: example

F.Y.I.

Find the **inverse** z-transform of

$$Y(z) = \frac{1+z^{-1}}{(1+0.5z^{-1})(1-0.5z^{-1})} = \frac{z(z+1)}{(z+0.5)(z-0.5)} \quad |z| > 0.5$$

Expanding in partial fractions using **positive powers** z, **after Euclidean division**, we obtain

$$\begin{aligned} Y(z) &= 1 + \frac{z+0.25}{(z+0.5)(z-0.5)}, \\ &= 1 + \frac{0.75}{z-0.5} + \frac{0.25}{z+0.5}, \\ &= 1 + z^{-1}\,\frac{0.75\,z}{z-0.5} + z^{-1}\,\frac{0.25\,z}{z+0.5}. \end{aligned}$$

$$y[k] = \delta[k] + \left[0.75\,(0.5)^{(k-1)} + 0.25\,(-0.5)^{(k-1)}\right] u[k-1].$$

Partial fraction expansion in z: example

F.Y.I.

$$
\begin{aligned}
y[k] &= \delta[k] + \left[0.75\,(0.5)^{(k-1)} + 0.25\,(-0.5)^{(k-1)}\right] u[k-1], \\
&= \delta[k] + \left[\frac{0.75}{0.5}\,(0.5)^{k} + \frac{0.25}{-0.5}\,(-0.5)^{k}\right] u[k-1], \\
&= \delta[k] + \left[1.5\,(0.5)^{k} - 0.5\,(-0.5)^{k}\right] u[k-1], \\
&= \delta[k] + \left[1.5\,(0.5)^{k} - 0.5\,(-0.5)^{k}\right] u[k] - \delta[k], \\
&= [1.5\,(0.5)^{k} - 0.5\,(-0.5)^{k}]\, u[k].
\end{aligned}
$$

LTI system analysis

The **response** $y[k]$ of a system described by a **difference equation** of order N with **real** constant coefficients

$$y[k] + \sum_{m=1}^{N} a_m\, y[k-m] = \sum_{m=0}^{M} b_m\, x[k-m]$$

with $N \geq M$ and **initial conditions** $y[-k], k = 1, \cdots, N$ is obtained by inverting the z-transform

$$Y(z) = \frac{b(z)}{a(z)}\, X(z) + \frac{1}{a(z)}\, I(z)$$

where $Y(z) = \mathcal{Z}[y[k]]$, $X(z) = \mathcal{Z}[x[k]]$ and $I(z)$ depends on the **initial conditions** and

$$a(z) = 1 + \sum_{m=1}^{N} a_m\, z^{-m}, \;\; b(z) = \sum_{m=0}^{M} b_m\, z^{-m}.$$

Analysis of LTI systems

The **complete response** $y[k]$ is obtained by **inverting** the z-transform

$$Y(z) = \frac{b(z)}{a(z)} X(z) + \frac{1}{a(z)} I(z)$$

which gives

$$y[k] = y_{zs}[k] + y_{zi}[k]$$

where

- $y_{zs}[k] = \mathcal{Z}^{-1}[H(z)X(z)]$ is the **zero-state response**,
- $y_{zi}[k] = \mathcal{Z}^{-1}[H_i(z)I(z)]$ is the **zero-input response**, with

$$H(z) = \frac{b(z)}{a(z)} \text{ et } H_i(z) = \frac{1}{a(z)}.$$

The associated **transfer function** is $H(z)$.

Example 1

Consider the **second order recursive equation**

$$6\,y[n-2] + y[n-1] - y[n] = 0 \quad \text{with} \quad y[-1] = -1/6,\; y[-2] = 1/36.$$

Going to the z-**domain**, we obtain

$$6\left(z^{-2}Y(z) + z^{-1}y[-1] + y[-2]\right) + \left(z^{-1}Y(z) + y[-1]\right) - Y(z) = 0,$$

$$\left(6\,z^{-2} + z^{-1} - 1\right) Y(z) = -6\,z^{-1}y[-1] - 6\,y[-2] - y[-1] = z^{-1}.$$

This yields

$$\begin{aligned} Y(z) &= \frac{z^{-1}}{(3\,z^{-1}-1)(2\,z^{-1}+1)} \\ &= \frac{-z^{-1}}{(1-3\,z^{-1})(1+2\,z^{-1})} \\ &= \frac{-1/5}{(1-3\,z^{-1})} + \frac{1/5}{(1+2\,z^{-1})} \end{aligned}$$

Example 1

This yields

$$Y(z) \quad = \quad \frac{-1/5}{(1 - 3\,z^{-1})} + \frac{1/5}{(1 + 2\,z^{-1})}$$

The **inverse** z**-transform** yields

$$y[n] = \frac{1}{5}\,((-2)^n - 3^n)\,u[n].$$

<u>**Check**</u>:

From the **recursive equation** and the **initial conditions**, we obtain

$$y[0] = 6\,y[-2] + y[-1] = 0,$$

$$y[1] = 6\,y[-1] + y[0] = -1,$$

This is **compatible** with the **solution** above.

Example 2

Consider the **first order recursive equation**

$$y[n+1] - 2y[n] = (n+1)\,u[n] \quad \text{with} \quad y[0] = 2.$$

Going to the z-**domain**, we obtain

$$\begin{aligned}
(zY(z) - zy[0]) - 2Y(z) &= \frac{z}{(z-1)^2} + \frac{z}{z-1} \\
(z-2)Y(z) &= 2z + \frac{z^2}{(z-1)^2} \\
Y(z) &= \underbrace{\frac{2z}{z-2}}_{Y_1(z)} + \underbrace{\frac{z^2}{(z-2)(z-1)^2}}_{Y_2(z)}
\end{aligned}$$

$$\begin{aligned}
\frac{Y_2(z)}{z} &= \frac{z}{(z-2)(z-1)^2} = \frac{-2}{(z-1)} + \frac{-1}{(z-1)^2} + \frac{2}{(z-2)} \\
Y(z) &= \frac{2z}{z-2} + \frac{-2z}{(z-1)} + \frac{-z}{(z-1)^2} + \frac{2z}{(z-2)}
\end{aligned}$$

Example 2

We obtain

$$Y(z) = \frac{4z}{z-2} - \frac{2z}{(z-1)} - \frac{z}{(z-1)^2}.$$

The **inverse z-transform** yields

$$y[n] = (4\,2^n - (n+2))\,u[n].$$

Check:

The **solution** yields $y[0] = 2$ which is **compatible** with the **initial conditions**.

Laplace, Fourier and *z*-transforms

Laplace transform

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

$s = j\omega$

$\boxed{\sigma = 0 \text{ in ROC}}$

Fourier transform

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

$t = kT_s$

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

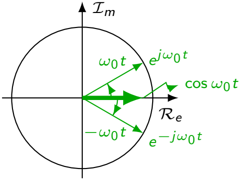
$$z = e^{T_s s}$$

***Z*-transform**

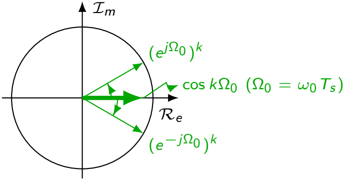
Sampled cosine: Nyquist OK $(\Omega_0 = \omega_0 T_s < \pi \iff \omega_0 < \omega_N)$

F.Y.I.

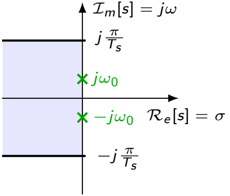
Continuous-time



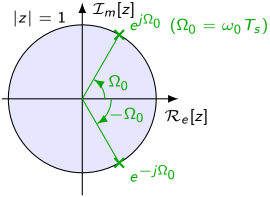
Discrete time $(t = kT_s)$



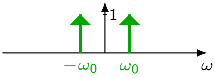
Laplace domain $(s = \sigma + j\omega)$



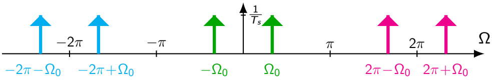
Z-domain $(z = r\,e^{j\Omega})$



Fourier transform



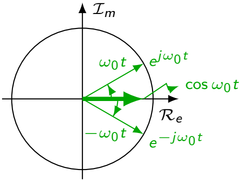
Fourier transform



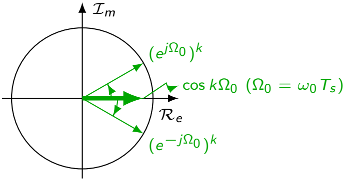
Sampled cosine: Nyquist OK ($\Omega_0 = \omega_0 T_s < \pi \iff \omega_0 < \omega_N$)

F.Y.I.

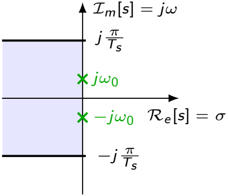
Continuous-time



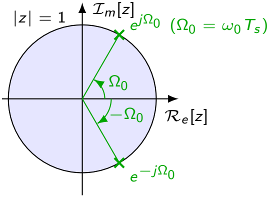
Discrete time ($t = kT_s$)



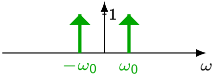
Laplace domain ($s = \sigma + j\omega$)



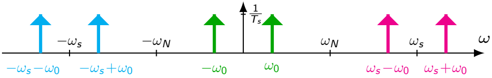
Z-domain ($z = r\,e^{j\Omega}$)



Fourier transform



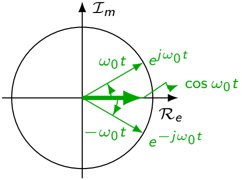
Fourier transform



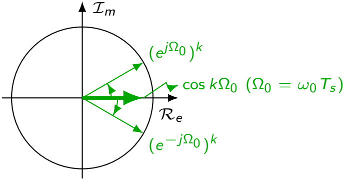
Sampled cosine: Nyquist NOK ($\Omega_0 = \omega_0 T_s > \pi \iff \omega_0 > \omega_N$)

F.Y.I.

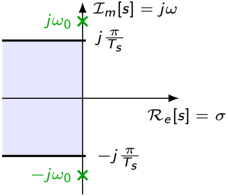
Continuous-time



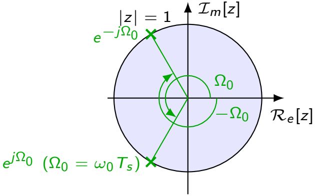
Discrete time ($t = kT_s$)



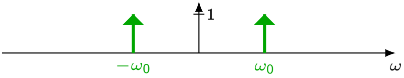
Laplace domain ($s = \sigma + j\omega$)



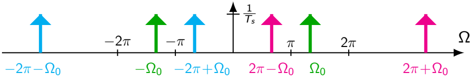
Z-domain ($z = r\,e^{j\Omega}$)



Fourier transform



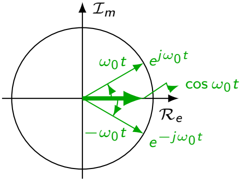
Fourier transform



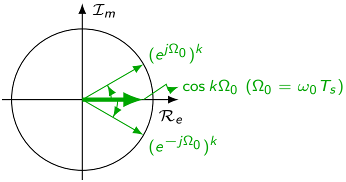
Sampled cosine: Nyquist NOK $(\Omega_0 = \omega_0 T_s > \pi \Longleftrightarrow \omega_0 > \omega_N)$

F.Y.I.

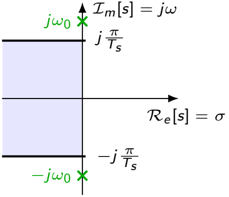
Continuous-time



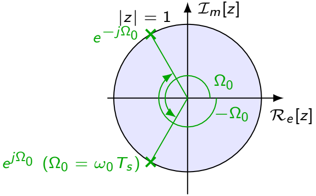
Discrete time $(t = kT_s)$



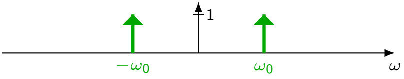
Laplace domain $(s = \sigma + j\omega)$



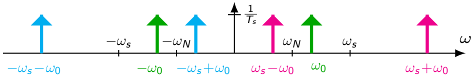
Z-domain $(z = r\, e^{j\Omega})$



Fourier transform



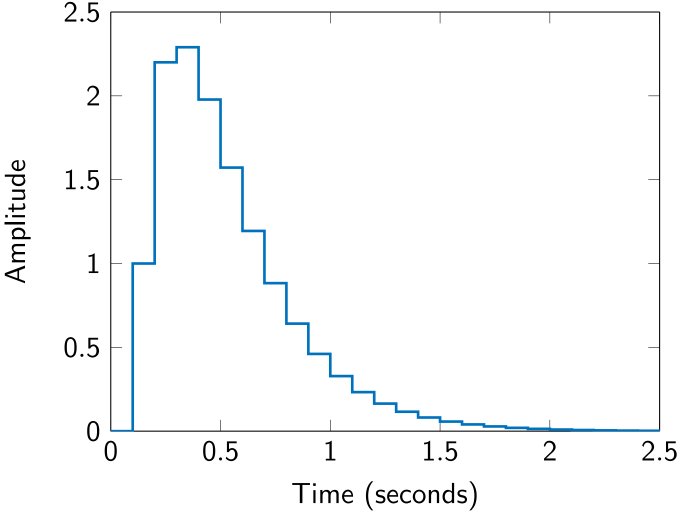
Fourier transform



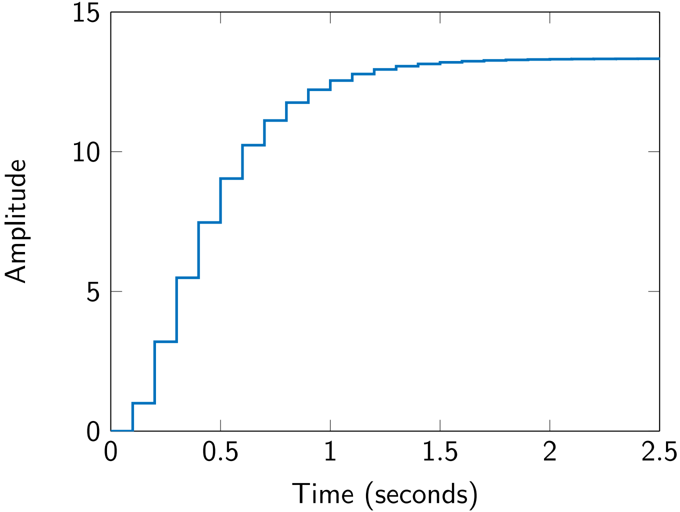
Responses of a discrete system

```
Ts = 0.1;
sysd = tf([1 1],conv([1 -0.5],[1 -0.7]),Ts);

figure
impulse(sysd*Ts) % To be consistent with discrete-time impulse seen in course

figure
step(sysd)

t = 0:Ts:10; x = sin(t);
y = lsim(sysd,x,t);
figure
subplot(211); stairs(t,x);
ylabel('Input')
subplot(212); stairs(t,y)
ylabel('Output'); xlabel('Time [s]')

figure
h = bodeplot(sysd);
p = getoptions(h);
p.Ylim{1}= [-50 30];
p.Ylim{2}= [-300 0];
setoptions(h,p);
```

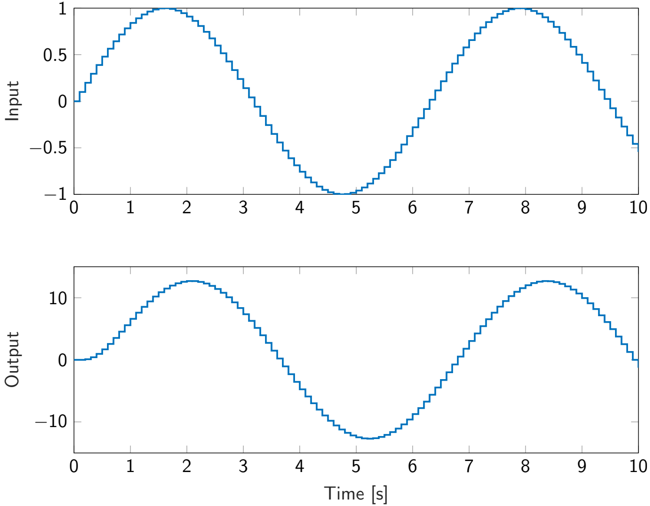
Impulse response



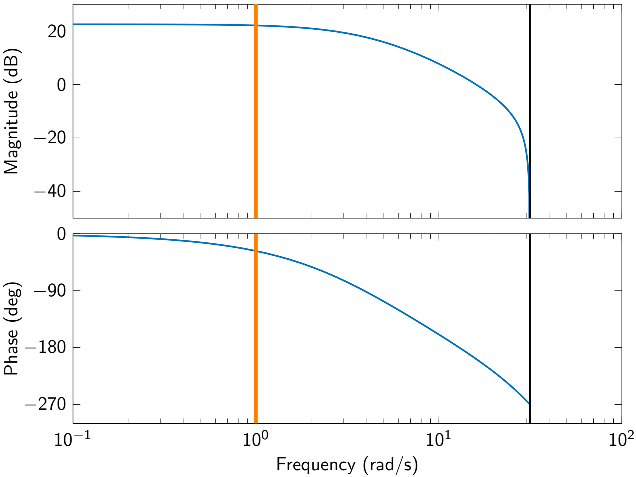
Step response response



Sinusoidal response



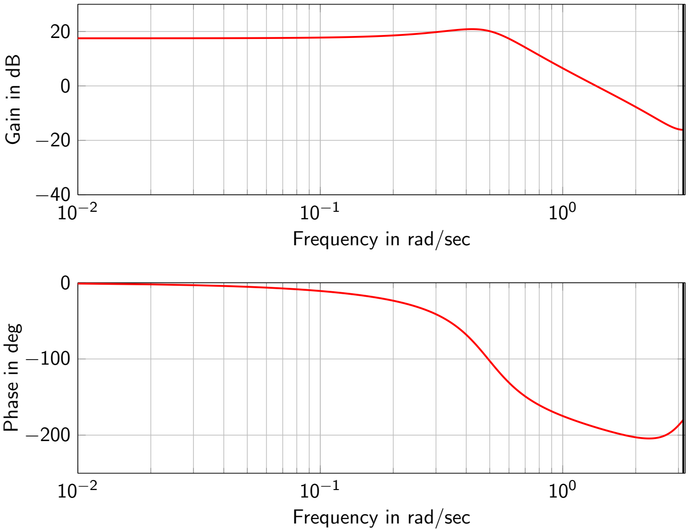
Bode plot


20
0
−20
−40
Magnitude (dB)
0
−90
−180
−270
Phase (deg)
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

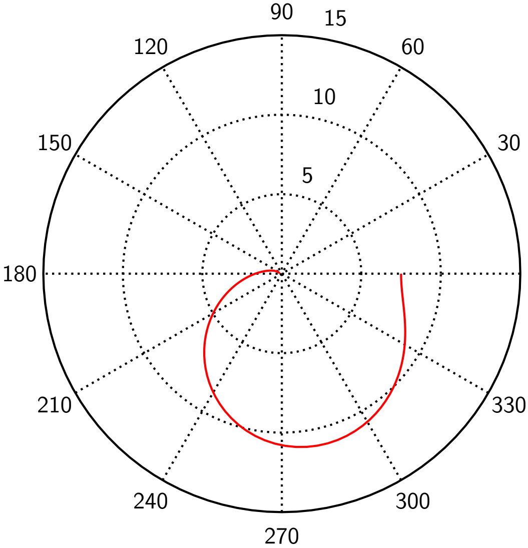
Bode and Nyquist plots

```
bp = [0 1 0.5];
ap = [1 -1.5 0.7];

w = 0:0.01:pi;
[mag,phase] = dbode(bp,ap,1,w);

% Bode diagram
figure(1)
subplot(211)
semilogx(w,20*log10(mag),'r','Linewidth',2);
grid on; axis([0.01 pi -40 30]);
ylabel('Gain in dB');
xlabel('Frequency in rad/sec');

subplot(212)
semilogx(w,phase,'r','Linewidth',2);
grid on; axis([0.01 pi -250 0]);
ylabel('Phase in deg');
xlabel('Frequency in rad/sec');

% Nyquist diagram
figure(2)
h1 = polar(phase*(pi/180),mag);
set(h1,'color','r','linewidth',2);
grid
```

Bode plot


20
0
−20
−40
Gain in dB
10^{-2}
10^{-1}
10^{0}
Frequency in rad/sec
0
−100
−200
Phase in deg
10^{-2}
10^{-1}
10^{0}
Frequency in rad/sec

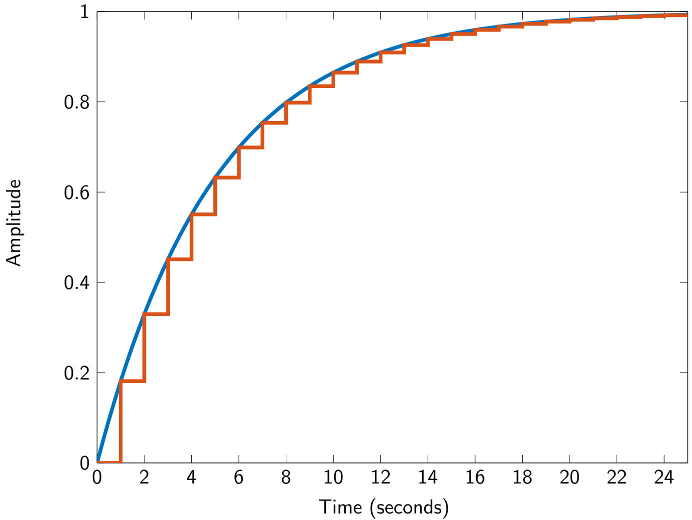
Nyquist plot



Discretisation

```
clear all
num = [0 1]; % H(s) = 1/(5s + 1)
den = [5 1];
sys = tf(num,den);
Ts = 1;
sysd = c2d(sys,Ts); % Discretisation of continuous system
step(sys,sysd,25)
```

Discretisation



Z-transforms

```
% pkg load symbolic % uncomment for use with octave with symbolic toolbox
syms a n z
f1 = a^n;
f2 = n*a^n;
ztrans(f1, z)
ztrans(f2, z)

syms n z
F = z/(z-3)^2 + z/(z-2);
f = iztrans(F, n)
simplify(f)
```

```
ans =

-z/(a - z)

ans =

(a*z)/(a - z)^2

f =

2^n + 3^n/3 + (3^n*(n - 1))/3

ans =

(3^n*n)/3 + 2^n
```

One-sided z-transforms		
$f[k]$	$\mathcal{Z}[f[k]u[k]]$	ROC
$\delta[k]$	1	z-domain
1	$\dfrac{1}{1-z^{-1}}$	$\|z\| > 1$
k	$\dfrac{z^{-1}}{(1-z^{-1})^2}$	$\|z\| > 1$
α^k	$\dfrac{1}{1-\alpha z^{-1}}$	$\|z\| > \|\alpha\|$
$k\,\alpha^k$	$\dfrac{\alpha z^{-1}}{(1-\alpha z^{-1})^2}$	$\|z\| > \|\alpha\|$

One-sided z-transforms		
$f[k]$	$\mathcal{Z}[f[k]u[k]]$	ROC
$\cos(\Omega_0 k)$	$\frac{1-\cos(\Omega_0)z^{-1}}{1-2\cos(\Omega_0)z^{-1}+z^{-2}}$	$\|z\| > 1$
$\sin(\Omega_0 k)$	$\frac{\sin(\Omega_0)z^{-1}}{1-2\cos(\Omega_0)z^{-1}+z^{-2}}$	$\|z\| > 1$
$r_0^k \cos(\Omega_0 k)$	$\frac{1-r_0\cos(\Omega_0)z^{-1}}{1-2r_0\cos(\Omega_0)z^{-1}+r_0^2 z^{-2}}$	$\|z\| > r_0$
$r_0^k \sin(\Omega_0 k)$	$\frac{\sin(\Omega_0)r_0 z^{-1}}{1-2r_0\cos(\Omega_0)z^{-1}+r_0^2 z^{-2}}$	$\|z\| > r_0$

Basic properties of one-sided z-transforms		
Properties	$f[k]$	$F(z)$
Causal signals and constants	$\alpha f[k]$, $\beta g[k]$	$\alpha F(z)$, $\beta G(z)$
Linearity	$\alpha f[k] + \beta g[k]$	$\alpha F(z) + \beta G(z)$
Convolution	$[h * x][k]$ $= \sum_n x[k] h[k-n]$	$H(z)X(z)$
Time shifting (zero I.C.)	$f[k - k_0]$	$z^{-k_0} F(z)$
Time shifting	$f[k - k_0]$	$z^{-k_0} F(z) + f[-1] z^{-k_0+1}$ $+ f[-2] z^{-k_0+2} + \cdots + f[-k_0]$

Basic properties of one-sided z-transforms		
Properties	$f[k]$	$F(z)$
Time shifting (zero I.C.)	$f[k+k_0]$	$z^{k_0}F(z)$
Time shifting	$f[k+k_0]$	$z^{k_0}F(z) - f[0]z^{k_0} - f[1]z^{k_0-1} - \cdots - f[k_0-1]z$

Basic properties of one-sided z-transforms		
Properties	$f[k]$	$F(z)$
Multiplication by k	$k\,f[k]$	$-z\,\dfrac{dF(z)}{dz}$
Finite difference	$f[k]-f[k-1]$	$(1-z^{-1})\,F(z)-f[-1]$
Accumulation	$\sum_{n=0}^{k} f[k]$	$\frac{F(z)}{1-z^{-1}}$
Initial value	$f[0]$	$\lim_{z\to\infty} F(z)$
Final value	$\lim\limits_{k\to\infty} f[k]$	$\lim_{z\to 1}\,(z-1)\,F(z)$

9. Discrete-time Fourier frequency analysis

Discrete-time Fourier transform

Discrete Fourier transform

- Time aliasing
- Zero padding
- Spectral leakage
- Practical aspects
- Illustrating example

Fourier series and transforms

Fast Fourier transform

Fast Fourier transform algorithm

Matlab and Octave

Laplace, Fourier and z-transforms and DTFT

Laplace transform

$$F(s) = \int_{-\infty}^{\infty} f(t)\, e^{-st} dt$$

$s = j\omega$

$\boxed{\sigma = 0 \text{ in ROC}}$

Fourier transform

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

$t = kT_s$

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

Z-transform

Laplace, Fourier and z-transforms and DTFT

Laplace transform

$$F(s) = \int_{-\infty}^{\infty} f(t)\, e^{-st} dt$$

$s = j\omega$

$\boxed{\sigma = 0 \text{ in ROC}}$

Fourier transform

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

$t = kT_s$

$t = kT_s$

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

Z-transform

$z = e^{j\Omega}$

$\boxed{|z| = 1 \text{ in ROC}}$

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k]\, e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

DTFT

Discrete-time Fourier transform

Discrete-time Fourier transform

The **Discrete-Time Fourier Transform** (DTFT) of a discrete-time signal $f[k]$,

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k]e^{-j\Omega k}, \ \ -\pi \leq \Omega < \pi$$

converts $f[k]$ into $F(\Omega)$ of the **discrete frequency**[88] Ω (rad).

The inverse transform gives back $f[k]$ from $F(\Omega)$ according to

$$f[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(\Omega)e^{j\Omega k} d\Omega.$$

[88] The discrete frequency Ω is a continuous variable over $[-\pi, \pi)$.

Discrete-time Fourier transform: link with z-transform

If in the above, we ignore T_s and consider $f(kT_s)$ a function of k, $f[k]$, we can see that

$$\boxed{F(\Omega) = F(z)|_{z=e^{j\Omega}}}$$

That is, it is the z-transform computed on the **unit circle**. For the above to happen, $F(z)$ **must** have a **region of convergence** (ROC) that **includes** the **unit circle**.

However, any discrete-time signal $f[k]$, of **finite support** in time, has a z-transform $F(z)$ with a region of convergence the **whole** z-plane, excluding either the origin or infinity, and as such its DTFT $F(\Omega)$ is **computed** from $F(z)$ by letting $z = e^{j\Omega}$.

Discrete frequency Ω

- The DTFT spectrum is continuous **periodic** of period 2π, only the frequencies $\Omega \in [-\pi, \pi)$ need to be considered.
- When plotting or displaying the spectrum of a **real-valued** discrete-time signal it is important to know that it is only necessary to show the **magnitude** and the **phase spectra** for **frequencies** $\Omega \in [0 \;\; \pi]$ since
  - the **magnitude spectrum** is an **even** function of Ω and
  - the **phase spectrum** is an **odd** function of Ω.
- **Interpretation** of Ω:

$$\Omega = \omega T_s = \omega T_s \frac{\pi}{\pi} = \pi \frac{\omega}{\omega_N} = 2\pi \frac{\omega}{\omega_s}$$

Discrete-time Fourier transform: example

F.Y.I.

Consider the DTFT of $\Pi[k] = u[k] - u[k-N]$. Since $\Pi[k]$ has **finite support**, its z-transform converges everywhere **except** at the **origin**, i.e.

$$\Pi(z) = \sum_{k=0}^{N-1} z^{-k} = \mathcal{Z}[u[k]]\,(1 - z^{-N}) = \frac{1 - z^{-N}}{1 - z^{-1}}.$$

The **DTFT** is given by

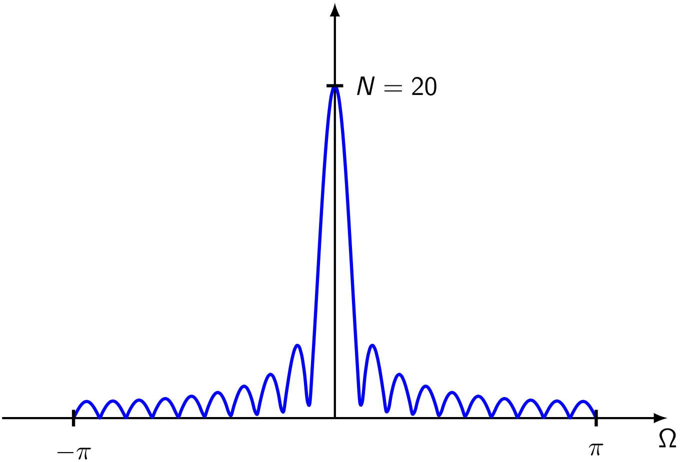
$$\begin{aligned}\Pi(e^{j\Omega}) &= \frac{1 - e^{-j\Omega N}}{1 - e^{-j\Omega}} = \frac{e^{-j\frac{\Omega N}{2}}}{e^{-j\frac{\Omega}{2}}} \frac{e^{j\frac{\Omega N}{2}} - e^{-j\frac{\Omega N}{2}}}{e^{j\frac{\Omega}{2}} - e^{-j\frac{\Omega}{2}}} \\ &= e^{-j\frac{\Omega(N-1)}{2}} \frac{\sin(\frac{\Omega N}{2})}{\sin(\frac{\Omega}{2})}\end{aligned}$$

The function $\sin(\frac{\Omega N}{2})/\sin(\frac{\Omega}{2})$ plays the **same role** as the **sinc function** for the **continuous-time Fourier transform**.

Discrete-time Fourier transform: example

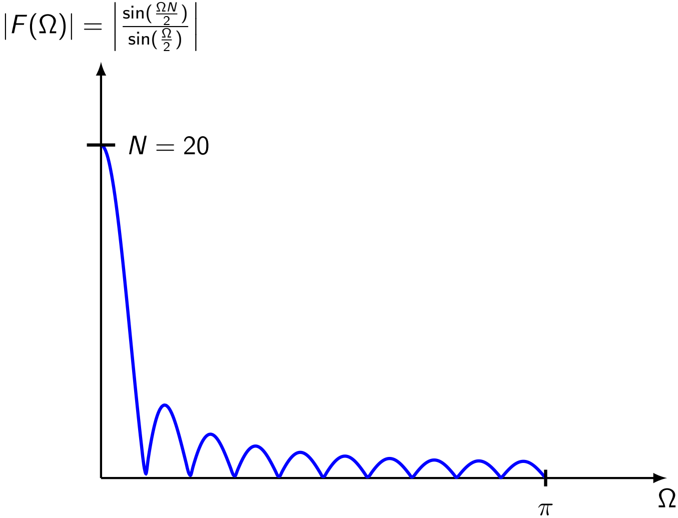
F.Y.I.

$$|F(\Omega)| = \left| \frac{\sin(\frac{\Omega N}{2})}{\sin(\frac{\Omega}{2})} \right|$$



Discrete-time Fourier transform: example

F.Y.I.



Laplace, Fourier and z-transforms and DTFT

Laplace transform

$$F(s) = \int_{-\infty}^{\infty} f(t)\, e^{-st} dt$$

$s = j\omega$

$\boxed{\sigma = 0 \text{ in ROC}}$

Fourier transform

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

$t = kT_s$

$t = kT_s$

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

Z-transform

$z = e^{j\Omega}$

$\boxed{|z| = 1 \text{ in ROC}}$

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k]\, e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

DTFT

Z-transform, DTFT and DFT

Z-transform

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

$$\xrightarrow{z = e^{j\Omega}}$$

$\boxed{|z| = 1 \text{ in ROC}}$

DTFT

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k]\, e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

Z-transform, DTFT and DFT

Z-transform

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

$$z = e^{j\Omega}$$

$$\boxed{|z| = 1 \text{ in ROC}}$$

DTFT

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k]\, e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

$$\Omega = \Omega_n = \frac{2\pi n}{L},$$
$$\omega = 2\omega_N \frac{n}{L},$$
$$n = 0, \cdots, L-1,$$
$$k = 0, \cdots, N-1.$$

$$F[n] = \sum_{k=0}^{N-1} f[k]\, e^{-j\frac{2\pi kn}{L}}$$

DFT

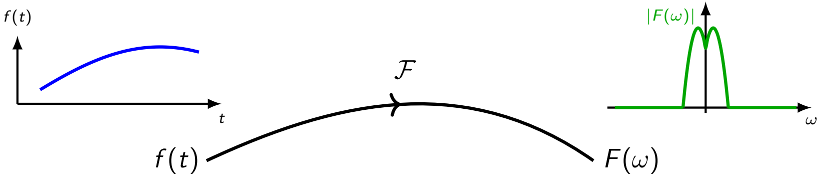
Discrete Fourier transform

- The **DFT** of a signal $f[k]$, $k = 0, \cdots, N-1$ is obtained by **sampling** its **DTFT** $X(\Omega)$ in **frequency**, i.e.

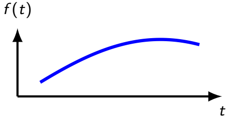
$$\boxed{\Omega_n = \frac{2\pi n}{L}, n = 0, \cdots, L-1}$$

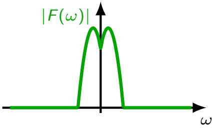
- We know that **sampling a continuous signal** over **time** yields a **periodic spectrum over frequency**.
- By **time-frequency duality**, sampling a **spectrum** over **frequency** yields a **periodic time sequence** !

Time sampling and frequency-domain periodicity



Time sampling and frequency-domain periodicity

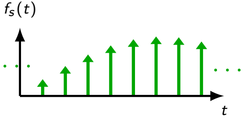




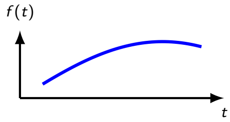
$$f(t) \xrightarrow{\mathcal{F}} F(\omega)$$

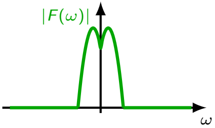
$$t = kT_s$$

$$f_s(t) = \sum_{k=-\infty}^{\infty} f(kT_s)\,\delta(t - kT_s)$$



Time sampling and frequency-domain periodicity

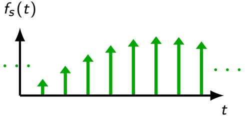


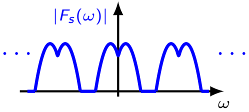


$$f(t) \xrightarrow{\mathcal{F}} F(\omega)$$

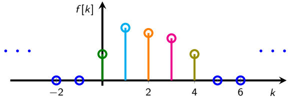
$$t = kT_s$$

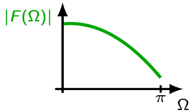
$$f_s(t) = \sum_{k=-\infty}^{\infty} f(kT_s)\,\delta(t - kT_s) \xrightarrow{\mathcal{F}} F_s(\omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} F(\omega - k\omega_s)$$





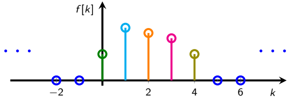
Frequency sampling and time-domain periodicity

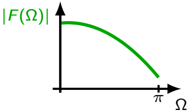




$$f[k], k = 0, \cdots, N-1 \xrightarrow{\text{DTFT}} F(\Omega) = \sum_{k=0}^{N-1} f[k]\, e^{-j\Omega k}$$

Frequency sampling and time-domain periodicity



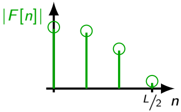


DTFT

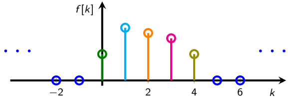
$$f[k],\ k = 0, \cdots, N-1 \quad \xrightarrow{\text{DTFT}} \quad F(\Omega) = \sum_{k=0}^{N-1} f[k]\, e^{-j\Omega k}$$

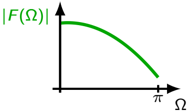
$$\Omega = \Omega_n = \frac{2\pi n}{L} \qquad n = 0, \cdots, L-1$$

$$F[n] = \sum_{k=0}^{N-1} f[k]\, e^{-j\frac{2\pi kn}{L}}$$



Frequency sampling and time-domain periodicity





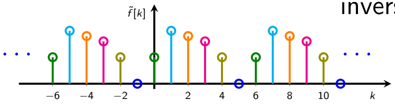
DTFT

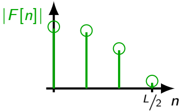
$$f[k],\, k = 0, \cdots, N-1 \qquad F(\Omega) = \sum_{k=0}^{N-1} f[k]\, e^{-j\Omega k}$$

$$\Omega = \Omega_n = \frac{2\pi n}{L} \qquad n = 0, \cdots, L-1$$

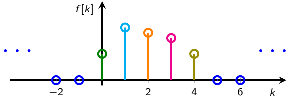
$$\tilde{f}[k] = \sum_{m=-\infty}^{\infty} f[k - mL] \qquad F[n] = \sum_{k=0}^{N-1} f[k]\, e^{-j\frac{2\pi kn}{L}}$$

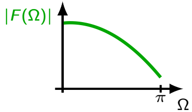
inverse DFT





Frequency sampling and time-domain periodicity





DTFT

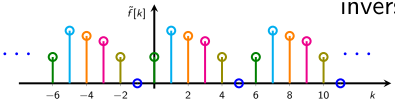
$$f[k], k = 0, \cdots, N-1 \qquad F(\Omega) = \sum_{k=0}^{N-1} f[k]\, e^{-j\Omega k}$$

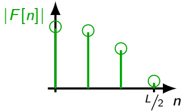
$$N = 5, L = 6$$

$$\Omega = \Omega_n = \frac{2\pi n}{L} \qquad n = 0, \cdots, L-1$$

$$\tilde{f}[k] = \sum_{m=-\infty}^{\infty} f[k - mL] \qquad F[n] = \sum_{k=0}^{N-1} f[k]\, e^{-j\frac{2\pi kn}{L}}$$

inverse DFT





Discrete Fourier transform

- The **DFT** of a signal $f[k]$, $k = 0, \cdots, N-1$ is obtained by **sampling** its **DTFT** $X(\Omega)$ in **frequency**. Suppose

$$\boxed{\Omega_n = \frac{2\pi n}{L}, n = 0, \cdots, L-1}$$

- We know that **sampling a continuous signal** over **time** yields a **periodic spectrum over frequency**.
- By **time-frequency duality**, sampling a **spectrum** over **frequency** yields a **periodic time sequence**, i.e.

$$\tilde{f}[k] = \sum_{m=-\infty}^{\infty} f[k - mL].$$

Discrete Fourier transform: time aliasing

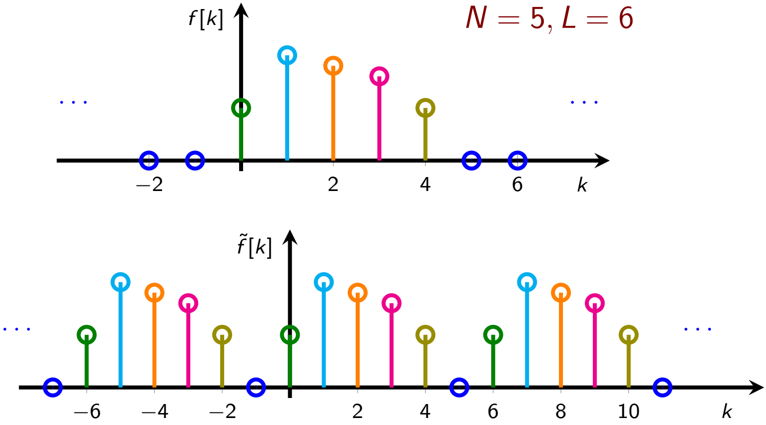
- By **time-frequency duality**, sampling a **spectrum** over **frequency** yields a **periodic time sequence**, i.e.

$$\tilde{f}[k] = \sum_{m=-\infty}^{\infty} f[k - mL].$$

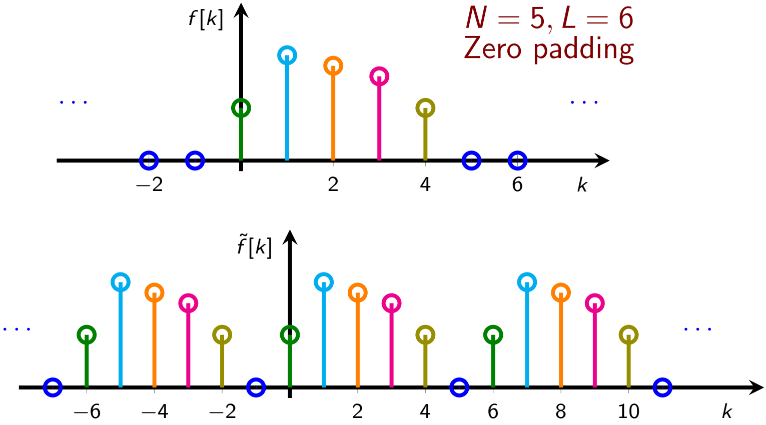
- If $L < N$, the first period of $\tilde{f}[k]$ does **not coincide** with $f[k]$ because of **superposition of shifted versions** of $f[k]$.

  This corresponds to **time aliasing**, the **dual** of **frequency aliasing**, which occurs in time sampling.

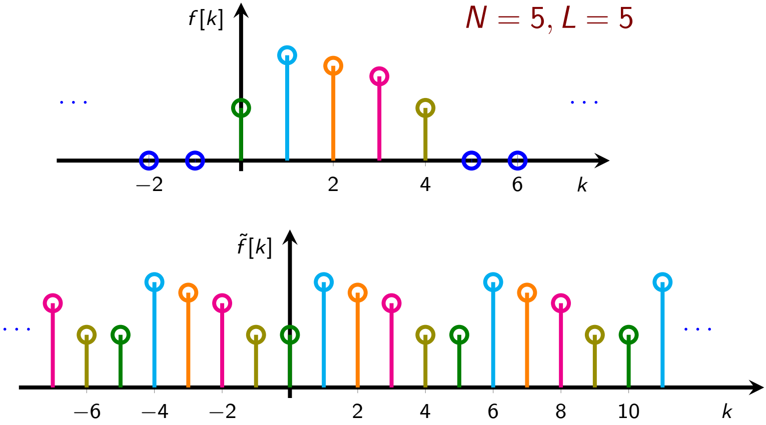
Discrete Fourier transform: no time aliasing with $L > N$



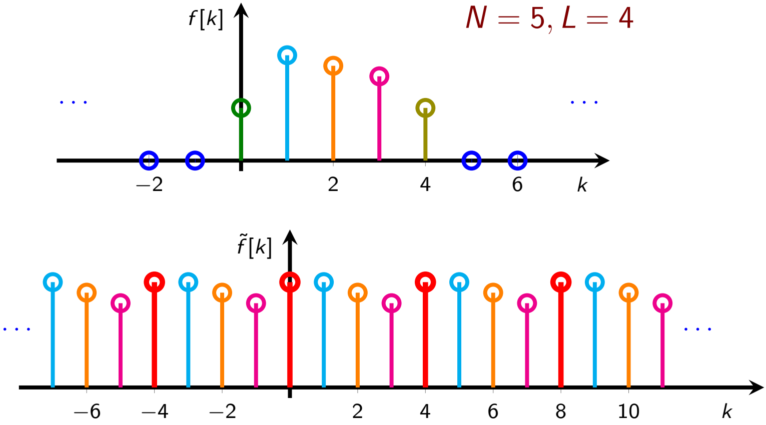
Discrete Fourier transform: no time aliasing with $L > N$



Discrete Fourier transform: no time aliasing with $L = N$



Discrete Fourier transform: **time aliasing** with $L < N$



Time aliasing: the mathematical way

F.Y.I.

- The **DTFT** of a sequence $y[k], k = 0, \cdots, N$ is

$$F(\Omega) = \sum_{k=0}^{N-1} f[k] e^{-j\Omega k}.$$

- After **sampling over frequency**, we obtain the **DFT**, i.e.

$$\begin{aligned} F[n] &= \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi kn}{L}} = \sum_{m=-\infty}^{\infty} \sum_{k=0}^{L-1} f[k-mL]\, e^{-j\frac{2\pi kn}{L}} \\ &= \sum_{k=0}^{L-1} \sum_{m=-\infty}^{\infty} f[k-mL]\, e^{-j\frac{2\pi kn}{L}} = \sum_{k=0}^{L-1} \tilde{f}[k]\, e^{-j\frac{2\pi kn}{L}} \end{aligned}$$

- **Sampling** in the **frequency domain** causes **time-domain aliasing**, which is obviously the **dual** of the well-known **aliasing** in the **frequency domain** caused by **sampling** in the **time domain**

Discrete Fourier transform

F.Y.I.

Discrete Fourier transform

The **DFT** of a signal $f[k]$ of length N is obtained by choosing $L \geq N$ with

$$\tilde{f}[k] = f[k] \text{ for } 0 \leq k < N, \;\; \tilde{f}[k] = 0 \text{ for } N \leq k < L,$$

$$F[n] = \sum_{k=0}^{L-1} \tilde{f}[k]\, e^{-j\frac{2\pi kn}{L}}, \;\; 0 \leq n \leq L-1.$$

The **inverse DFT** is

$$\tilde{f}[k] = \frac{1}{L} \sum_{n=0}^{L-1} F[n]\, e^{j\frac{2\pi kn}{L}}, \;\; 0 \leq k \leq L-1.$$

Note that both $F[n]$ and $\tilde{f}[k]$ are **periodic** of **period** L.

Zero padding: why $L > N$?

Hypothesis: DFT of an **aperiodic** signal $f[k]$ of length N

- If N is **small** and $L = N$, the DFT produces the DTFT spectrum sampled over N points, i.e. the spectral resolution is **poor**.

  To **increase** the **spectral resolution**, $L > N$ is chosen.

  In practice, a **number of zeros are added** at the **end** of the sequence. This is called **zero padding**.
- When the FFT is used (efficient implementation of the DFT), N needs to be a **power 2**. If this not case a number of zeros are added at the end of the sequence so that the length of the resulting sequence becomes a **power of 2**.

Hypothesis: DFT of a **periodic** signal[89] $f[k]$ of **period** N_p and length a multiple of N_p, i.e. $N = K\,N_p$.

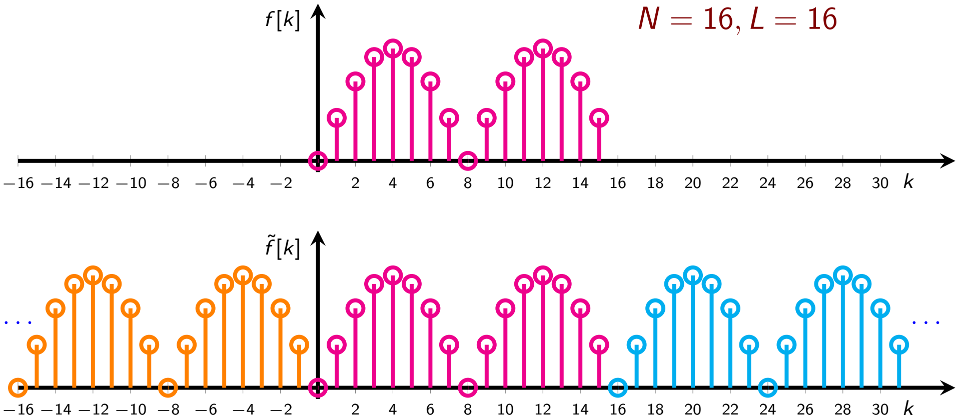
- **Several periods** of the signal ($K > 1$) can be **considered**. **Zero padding** should <u>**not**</u> be used in this case.

[89] This sequence $f[k]$ results from a periodic signal $f(t)$ that has been sampled with $T_s = m\,T/N_p$.

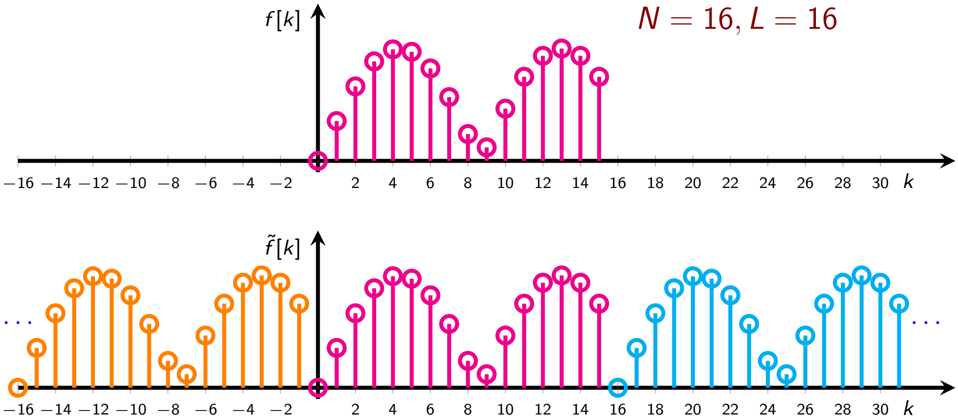
Spectral leakage

- Practice: a signal is measured over a **limited time span**.
- The DFT **presupposes** that the measured signal is **repeating** itself. It is assumed that the input signal is **one or several periods** of a **periodic signal**.
- Most measured signals are **not periodic** in the **sample interval**. Since the DFT interprets the data as a **repeating sequence**, it sees **discontinuities**.
- The discontinuities between the ends and the beginnings manifest themselves as **frequency components** that **do not really exist**. The components are called **spurious frequencies** or **spectral leakage**.

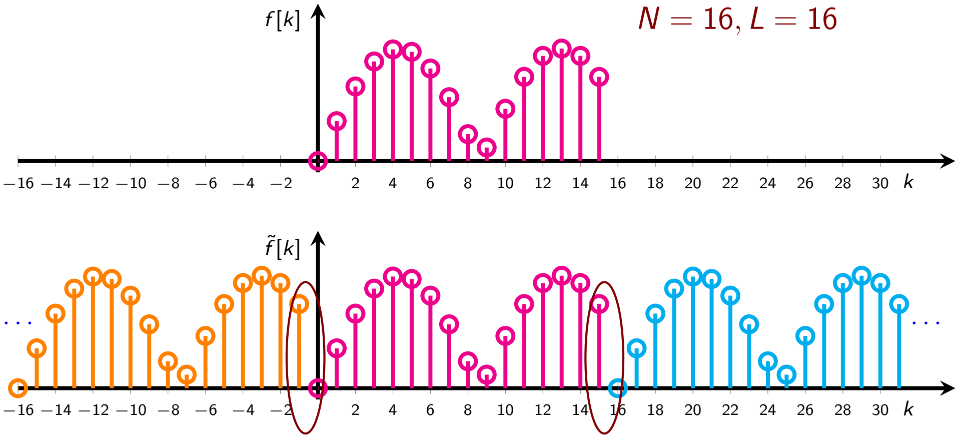
Discrete Fourier transform: no spectral leakage



Discrete Fourier transform: spectral leakage



Discrete Fourier transform: spectral leakage

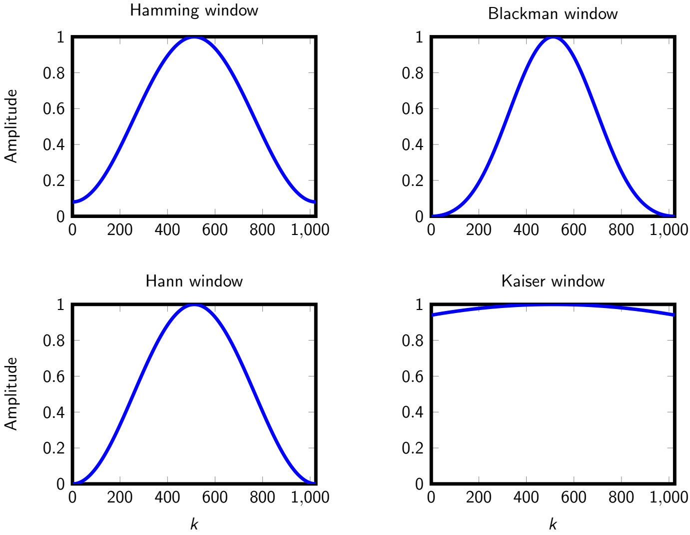


Discontinuities $\Rightarrow$ spectral leakage

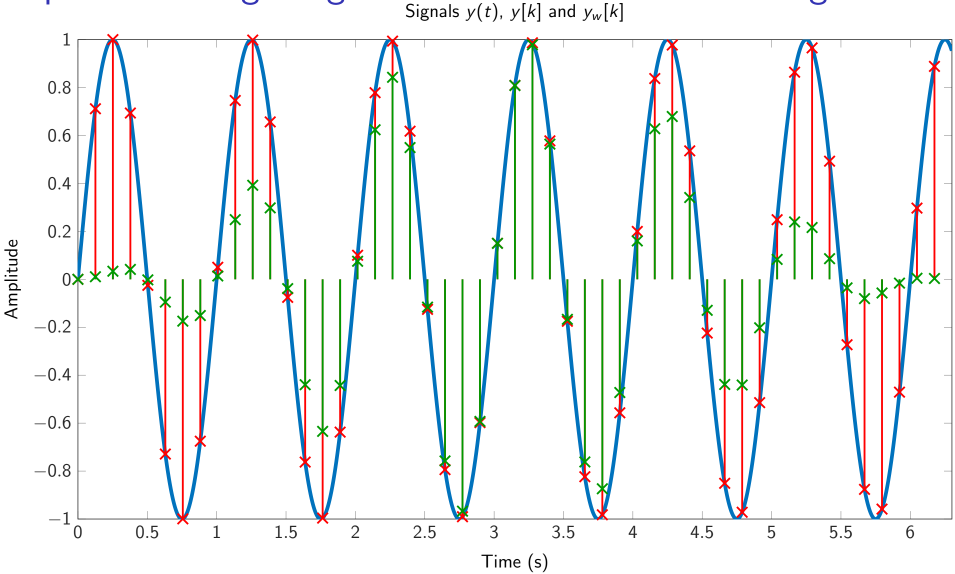
Windowing

- The discontinuities between the ends and the beginnings manifest themselves as frequency components that **do not really exist**. The components are called **spurious frequencies** or **spectral leakage**.
- **Solutions:**
  - **Windowing** functions effectively taper the ends of the segments to 0 or small values so that they connect from beginning to end. A **compromise** between reducing spectral leakage and smearing the spectrum needs to be thought.
  - Select N (large, appropriate choice) to **attenuate** the effects of spectral leakage.

Window functions



Spectral leakage: signal before and after windowing



Z-transform, DTFT and DFT: $L = N$

Z-transform

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

$$z = e^{j\Omega}$$

$$\boxed{|z| = 1 \text{ in ROC}}$$

DTFT

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k] e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

Z-transform, DTFT and DFT: $L = N$

Z-transform

$$F(z) = \sum_{k=-\infty}^{\infty} f[k] z^{-k}$$

$$z = e^{T_s s}$$

$z = e^{j\Omega}$

$\boxed{|z| = 1 \text{ in ROC}}$

DTFT

$$F(\Omega) = \sum_{k=-\infty}^{\infty} f[k] e^{-j\Omega k}$$

$$\Omega = \omega T_s$$

$$\Omega = \Omega_n = \frac{2\pi n}{N},$$
$$\omega = 2\omega_N \frac{n}{N},$$
$$n = 0, \cdots, N-1,$$
$$k = 0, \cdots, N-1.$$

DFT

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi kn}{N}}$$

Discrete Fourier transform

Discrete Fourier transform ($L = N$)

The **DFT** of a signal $f[k]$ of length N is

$$F[n] = \sum_{k=0}^{N-1} f[k]\, e^{-j\frac{2\pi kn}{N}}, \;\; 0 \leq n \leq N-1.$$

The **inverse DFT** is

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n]\, e^{j\frac{2\pi kn}{N}}, \;\; 0 \leq k \leq N-1.$$

Note that both $F[n]$ and $f[k]$ are **periodic** of **period** N.

DFT: practical aspects

- Remember that the **inverse DFT** is

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n]\, e^{j\frac{2\pi kn}{N}}, \;\; 0 \le k \le N-1.$$

- If we want to **reconstruct** a signal from the Fourier transform, it is necessary to **normalize** using a **factor** $1/N$.
- This is why the **spectrum** is represented **after normalisation** by a **factor** 1/N.

DFT: practical aspects

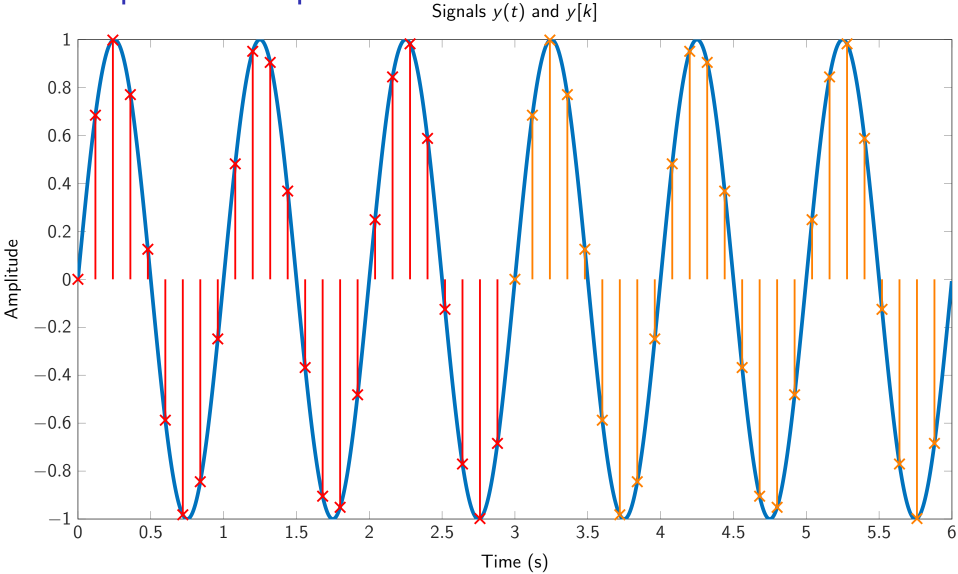
- **Real-valued** input sequence $f[k]$ with $k = 0, \cdots, N-1$.
- DFT produces a **complex** sequence $F[n]$ with $n = 0, \cdots, N-1$.
- Index n samples $\Omega_n = \frac{2\pi n}{N}$ over a period $\left[0, \cdots, 2\pi \frac{N-1}{N}\right]$ of the DFT.
- **Remember** that $n = {}^{N}\!/_{2}$ corresponds to $\Omega = \pi$, i.e. $\omega = \frac{\pi}{T_s} = \omega_N$ or, similarly $f = \frac{1}{2T_s} = f_s/2$.
- Instead of considering the **spectrum** over $[0, \cdots, 2\pi]$, we could also consider it over a <u>**shifted**</u> **period**, i.e. $[-\pi, \cdots, \pi]$, $[-\omega_N, \cdots, \omega_N]$ or $[-f_s/2, \cdots, f_s/2]$.
- Using Matlab, this can be achieved using the command `fftshift`. This command shifts the DC (zero frequency component) to the **center of the spectrum**.

DFT: practical aspects

```
close all; clear all

% Period of sine
T = 1;

% Sampling period
m = 3; % Test m = 1, m = 3, m = 3.1, m = 15
Np = 25;
Ts = m*T/Np; fs = 1/Ts; fprintf('fs = %4.2f Hz\n', fs)

% Number of data
K = 2; N = K*Np; % K discrete periods will be considered

tsampled = (0:N-1)*Ts; ysampled = sin((2*pi/T)*(tsampled));

fact = 100;
tsim = (0:N*fact)*Ts/fact; ysim = sin((2*pi/T)*(tsim));

figure
plot(tsim,ysim)
hold on
stem(tsampled,ysampled)
hold off
title('Signals y(t) and y[k]')
xlabel('Time (s)'), xlim([0 tsim(end)])
ylabel('Amplitude')
```

```
fs = 8.33 Hz
```

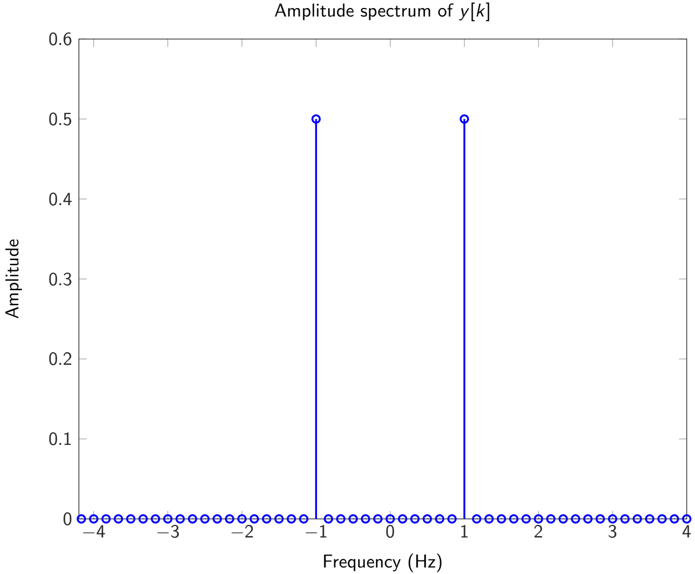
DFT: practical aspects



DFT: practical aspects

```
spectrum_y = fft(ysampled,N);
f = (-N/2:N/2-1)*(fs/N);
mag = (1/N)*abs(fftshift(spectrum_y));

figure
stem(f,mag)
title('Amplitude spectrum of y[k]')
xlabel('Frequency (Hz)')
ylabel('Amplitude'), ylim([0 0.6])
```

DFT: practical aspects



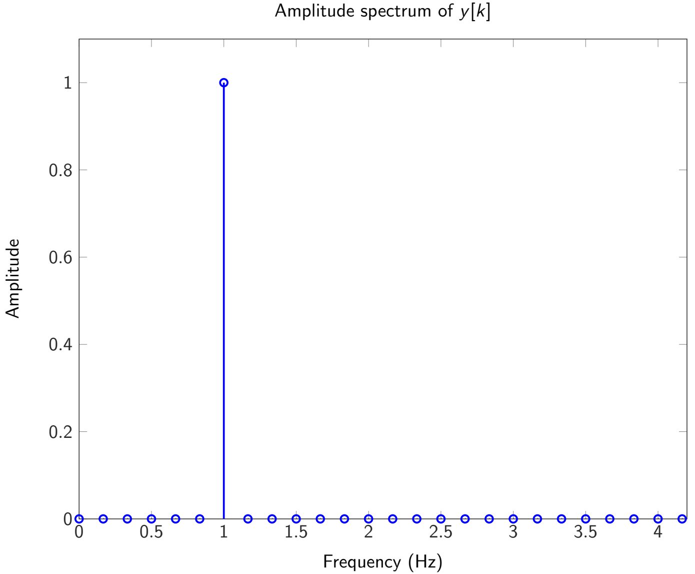
DFT: practical aspects

- The **amplitude** and **phase spectra** are, respectively, **even** and **odd**. The $F[n]$ with indices $n = 0, \cdots, \frac{N}{2}$ contain **all** the relevant information.
- A **sinusoisal signal** is equivalent to **2 purely imaginary complex exponentials** of **half amplitude**. The resulting **spectrum** has **two spikes** corresponding to the **original** and **mirrored frequencies**.
- If we want to **retrieve** the **amplitude** of the **original sinusoidal signal**, we have
  - to use a **normalisation factor** $2/N$. Note that for $n = 0$, the **DC** or **zero-frequency** component does **not** have to be **multiplied** by 2 !
  - to work over a **half period**, i.e. $[0, \cdots, \pi]$, $[0, \cdots, \omega_n]$ or $[0, \cdots, f_s/2]$ and work with **indices** $n = 0, \cdots, \frac{N}{2}$.

DFT: practical aspects

```
spectrum_y = fft(ysampled,N);
f = (0:N/2)*(fs/N);
mag = (2/N)*abs(spectrum_y(1:N/2+1));
mag(1) = mag(1)/2;

figure
stem(f,mag)
title('Amplitude spectrum of y[k]')
xlabel('Frequency (Hz)')
ylabel('Amplitude'), ylim([0 1.1])
```

DFT: practical aspects



DFT: practical aspects

F.Y.I.

[1]:
```
from scipy.fftpack import fft
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
# Signal parameters
T = 1   # seconds
f = 1/T # Hz

# Correct sampling condition: m*T = Np*Ts. m and Np are coprime integers
# We will use N = K*Np
m = 3; Np = 25; Ts = m*T/Np # seconds

# Sampling frequency
fs = 1/Ts # Hz

# Number of data
K = 2; N = K*Np # K*Np periods wil be considered

# Time signals
t = np.linspace(0, (N-1)*Ts, N)

factSim = 100; NSim = N*factSim
tSim = np.linspace(0, (NSim-1)*Ts/factSim, NSim)

# Output signals
y = np.sin(2*np.pi*f*t) # sampled signal
ySim = np.sin(2*np.pi*f*tSim) # "continuous-time" signal
```

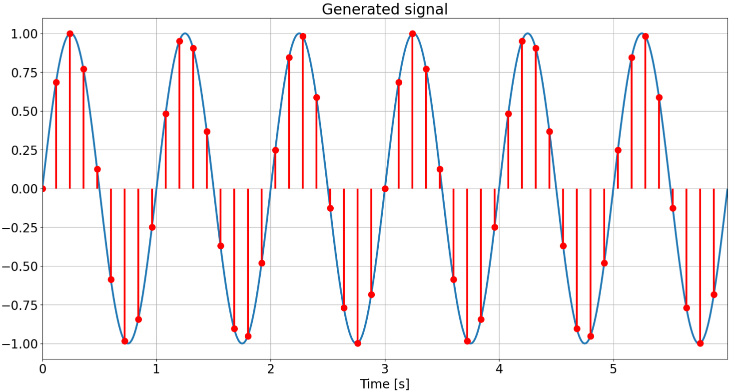
DFT: practical aspects

F.Y.I.

[3]:
```
plt.figure(figsize=(20,10))
plt.plot(tSim,ySim, linewidth = 3)
plt.stem(t,y,'r')
plt.grid()
plt.title('Generated signal')
plt.xlim([0,tSim[-1]])
plt.xlabel('Time [s]')
plt.show()

# print
print(' ')
print('fs = ', fs, 'Hz')
print("Ts = ", Ts, "s")
print(' ')
print("N = ", N)
fN = fs/2 # Nyquist frequency
if (fN > f):
    print('Nyquist condition is verified !')
else:
    print('Nyquist condition is NOT verified !')
print("fN = ", fN, "Hz")
print("Spectral resolution = ", fs/N, 'Hz')
print(' ')
print('Number of periods of sinusoid at f =', N*Ts*f)
```

DFT: practical aspects

F.Y.I.




```
fs =  8.333333333333334 Hz
Ts =  0.12 s

N =  50
Nyquist condition is verified !
fN =  4.166666666666667 Hz
Spectral resolution =  0.16666666666666669 Hz

Number of periods of sinusoid at f = 6.0
```

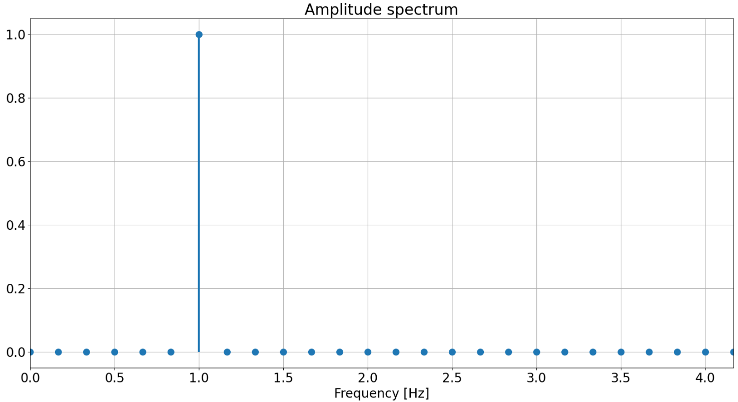
DFT: practical aspects

F.Y.I.

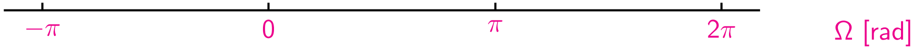
[4]:
```
f = np.linspace(0, fN, N//2 + 1)

spectrum_y = fft(y)
spectrum_y[0] = spectrum_y[0]/2
mag = (2/N)*np.abs(spectrum_y[0:N//2 + 1]);

plt.figure(figsize=(20,10))
plt.title('Amplitude spectrum')
plt.grid()
plt.xlabel('Frequency [Hz]')
plt.stem(f,mag)
plt.xlim([0,f[-1]])
plt.show()
```

DFT: practical aspects

F.Y.I.



DFT: practical aspects[90]



[90]Variables in magenta are related to **periodic spectra**.

DFT: practical aspects[90]

$-\pi$ | 0 | π | 2π | Ω [rad]

$-\frac{\pi}{T_s} = -\omega_N$ | 0 | $\frac{\pi}{T_s} = \omega_N$ | $\frac{2\pi}{T_s} = \omega_s$ | ω [rad/s]

[90] Variables in magenta are related to **periodic spectra**.

DFT: practical aspects[90]

$-\pi$	0	π	2π	Ω [rad]
$-\frac{\pi}{T_s} = -\omega_N$	0	$\frac{\pi}{T_s} = \omega_N$	$\frac{2\pi}{T_s} = \omega_s$	ω [rad/s]
$-\frac{1}{2T_s} = -f_N$	0	$\frac{1}{2T_s} = f_N$	$\frac{1}{T_s} = f_s$	f [Hz]

[90]Variables in magenta are related to **periodic spectra**.

DFT: practical aspects[90]

$-\pi$	0	π	2π	Ω [rad]
$-\frac{\pi}{T_s} = -\omega_N$	0	$\frac{\pi}{T_s} = \omega_N$	$\frac{2\pi}{T_s} = \omega_s$	ω [rad/s]
$-\frac{1}{2T_s} = -f_N$	0	$\frac{1}{2T_s} = f_N$	$\frac{1}{T_s} = f_s$	f [Hz]
	0	$\frac{N}{2}$	N	n [integer]

[90] Variables in magenta are related to **periodic spectra**.

DFT: practical aspects[90]

$-\pi$	0	π	2π	Ω [rad]
$-\frac{\pi}{T_s} = -\omega_N$	0	$\frac{\pi}{T_s} = \omega_N$	$\frac{2\pi}{T_s} = \omega_s$	ω [rad/s]
$-\frac{1}{2T_s} = -f_N$	0	$\frac{1}{2T_s} = f_N$	$\frac{1}{T_s} = f_s$	f [Hz]
	0	$\frac{N}{2}$	N	n [integer]

The **indices** n correspond to the **frequencies**

$$\boxed{f[n] = n\,\frac{f_N}{N/2} = n\,\frac{f_s}{N}}$$

[90]Variables in magenta are related to **periodic spectra**.

DFT: practical aspects

- The **indices** $n = 0, \cdots, \frac{N}{2}$ of the DFT correspond to a **frequency scale** in the **interval** $[0 \;\; f_N]$, i.e.

$$\boxed{\left[0 : 1 : \tfrac{N}{2}\right] \quad \Longleftrightarrow \quad \left[0 : \tfrac{f_s}{N} : f_N = \tfrac{f_s}{2}\right]}$$

- Another way to see this is that the **indices** $n = 0, \cdots, \frac{N}{2}$ correspond to the **frequencies**

$$\boxed{f[n] = n \, \frac{f_N}{N/2} = n \, \frac{f_s}{N}}$$

- The <u>**spectral resolution**</u> is $f_s/N = 1/(T_s \, N)$. We see that the **spectral resolution** in **inversely proportional** to the **duration** of the **experiment**, i.e. $T_s \, N$.
- Note that the **highest frequency** in the **spectrum** is the **Nyquist frequency** $f_N = {}^{f_s}/_2$.

DFT: practical aspects for a sinusoidal signal

Assume a **sinusoidal signal** of period T, i.e. frequency $f = 1/T$, sampled with

$$\boxed{mT = N_p T_s} \Longleftrightarrow f_s = \frac{N_p}{m} f$$

and $\boxed{f \leq f_N = f_s/2}$. Assume the **number of data** is $\boxed{N = K\,N_p}$.

The **spectral resolution** is

$$df = \frac{f_s}{N} = \frac{f}{m\,K}$$

It now follows that the **frequency** f will appear at index $n = m\,K$ in the **spectrum**.

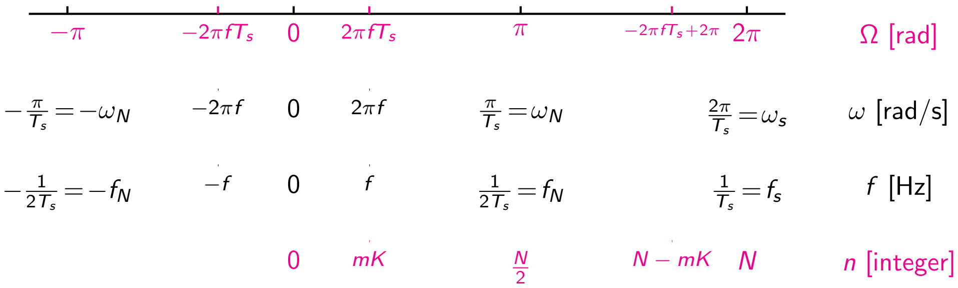
DFT: practical aspects for a sinusoidal signal

$-\pi$	0	π	2π	Ω [rad]
$-\frac{\pi}{T_s} = -\omega_N$	0	$\frac{\pi}{T_s} = \omega_N$	$\frac{2\pi}{T_s} = \omega_s$	ω [rad/s]
$-\frac{1}{2T_s} = -f_N$	0	$\frac{1}{2T_s} = f_N$	$\frac{1}{T_s} = f_s$	f [Hz]
	0	$\frac{N}{2}$	N	n [integer]

The **indices** n correspond to the **frequencies**

$$\boxed{f[n] = n\,\frac{f_N}{N/2} = n\,\frac{f_s}{N}}$$

DFT: practical aspects for a sinusoidal signal



The **indices** n correspond to the **frequencies**

$$\boxed{f[n] = n\,\frac{f_N}{N/2} = n\,\frac{f_s}{N}}$$

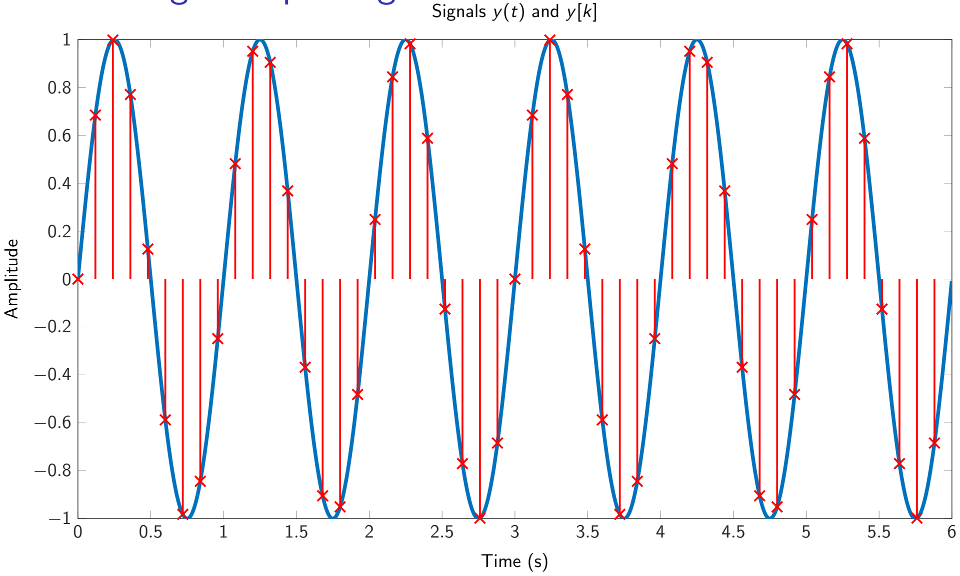
Illustrating example

```
close all; clear all

% Period of sine
T = 1;

% Sampling period
m = 3; Np = 25; Ts = m*T/Np; K = 2; N = K*Np; NFFT = N; window = 0;  % Test 1
% m = 3; Np = 25; Ts = m*T/Np; K = 2; N = K*Np; NFFT = 2*N; window = 0; % Test 2
% m = 3.15; Np = 25; Ts = m*T/Np; K = 2; N = K*Np; NFFT = N; window = 0; % Test 3
% m = 3.15; Np = 25; Ts = m*T/Np; K = 2; N = K*Np; NFFT = N; window = 1; % Test 4

fs = 1/Ts; fprintf('fs = %4.2f Hz\n', fs)

tsampled = (0:N-1)*Ts; ysampled = sin((2*pi/T)*(tsampled));

fact = 100;
tsim = (0:N*fact-1)*Ts/fact; ysim = sin((2*pi/T)*(tsim));
```

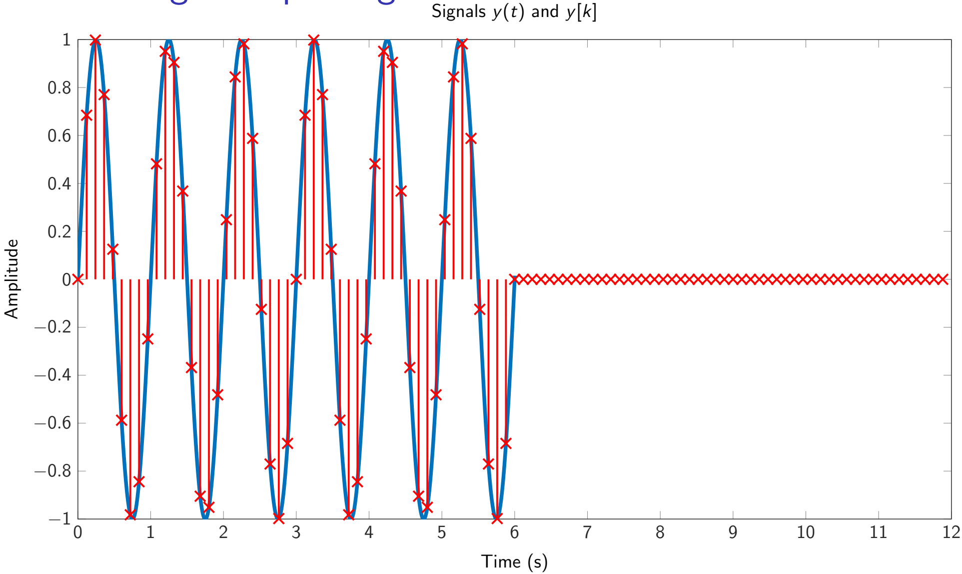
Illustrating example

```
    if window % Test 4
        ysampled = hanning(length(ysampled')) .* ysampled';
    end

    if NFFT == 2*N  % Test 2
        tsampled = [tsampled'; tsampled'+N/fs]; ysampled = [ysampled'; 0*ysampled'];
        tsim = [tsim'; tsim'+N/fs]; ysim = [ysim'; ysim'];
    end

    figure(1)
    plot(tsim,ysim)
    hold on
    stem(tsampled,ysampled)
    hold off
    title('Signals y(t) and y[k]')
    xlabel('Time (s)'), xlim([0 tsim(end)])
    ylabel('Amplitude')

    spectrum_y = fft(ysampled,NFFT);
    f = (0:NFFT/2)*(fs/NFFT);
    mag = (2/NFFT)*abs(spectrum_y(1:NFFT/2+1));
    mag(1) = mag(1)/2;

    figure(2)
    stem(f,mag)
    title('Amplitude spectrum of y[k]')
    xlabel('Frequency (Hz)')
    ylabel('Amplitude'), ylim([0 1.1])
```

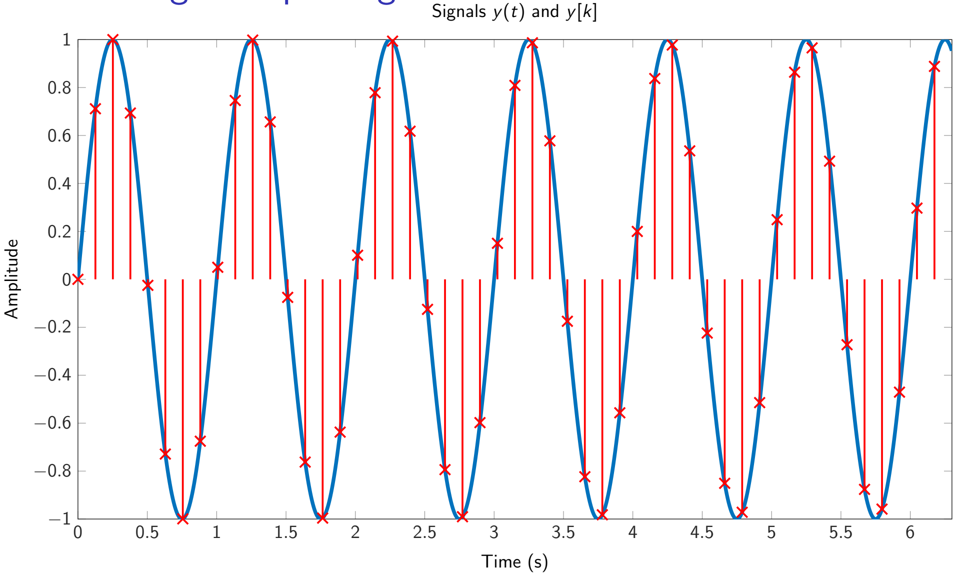
Illustrating example: signal of test 1



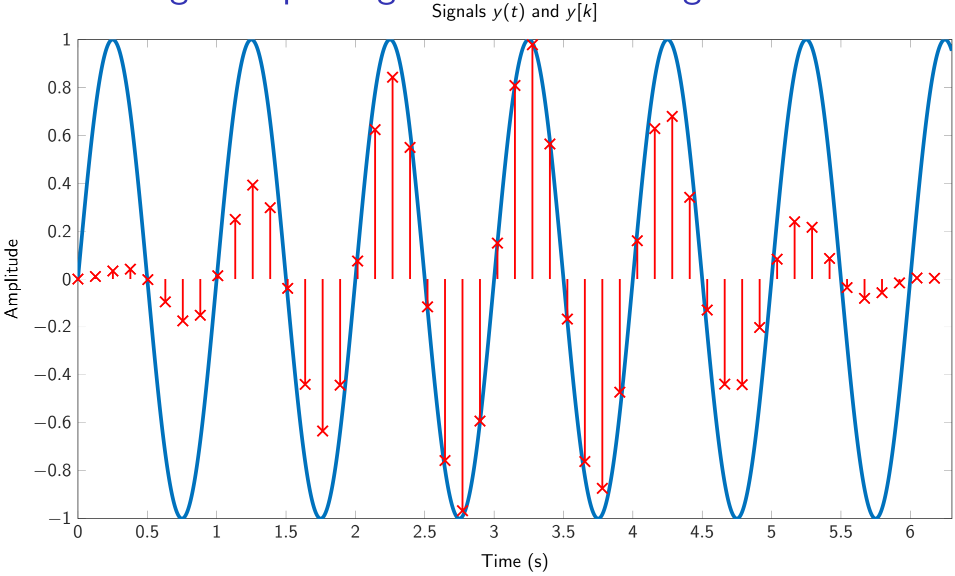
Illustrating example: signal of test 2



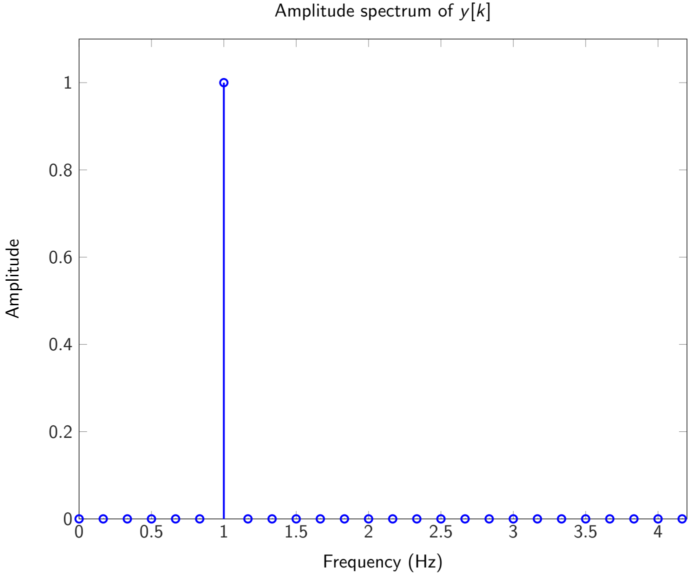
Illustrating example: signal of test 3



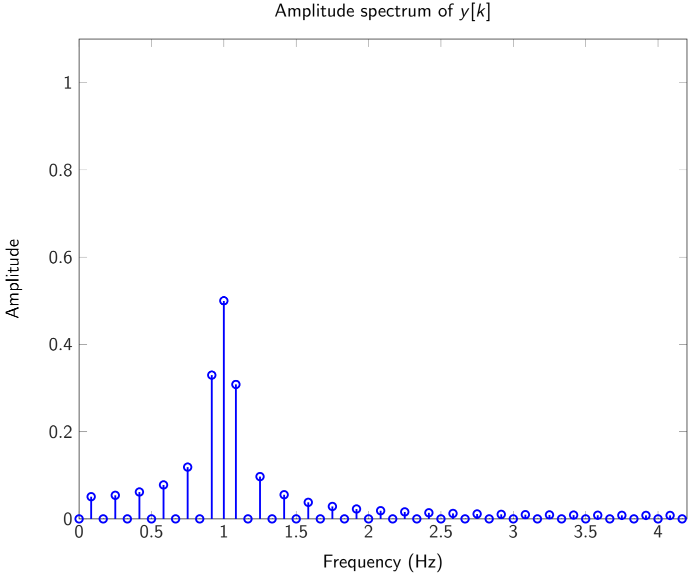
Illustrating example: signal after windowing of test 4



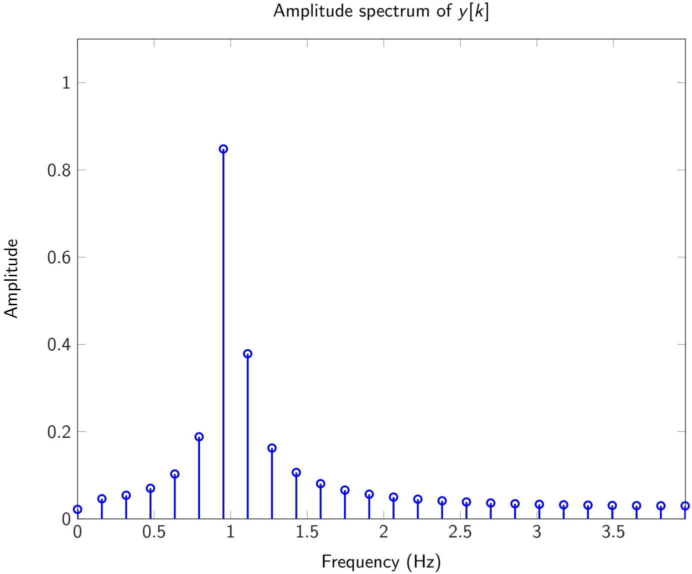
Illustrating example: spectrum of test 1



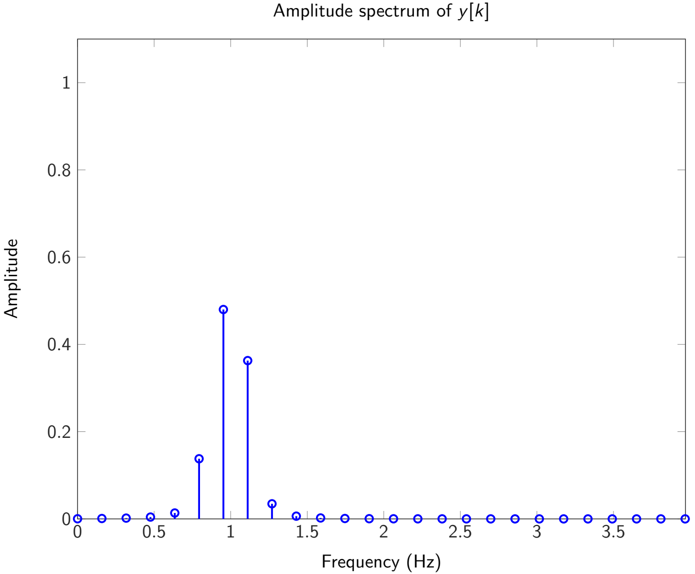
Illustrating example: spectrum of test 2



Illustrating example: spectrum of test 3



Illustrating example: spectrum of test 4



Fourier series and transforms: summary

- **Fourier Series (FS)**: applicable to **continuous-time** and **periodic** signals. The resulting **spectrum** is **discrete**.
- **Fourier Transform (FT)**: applicable to **continuous-time** signals. The resulting **spectrum** is **continuous**[91] and aperiodic.
- **Discrete-Time Fourier Transform (DTFT)**: applicable to **discrete-time** signals. The resulting **spectrum** is **continuous** and **periodic**.
- **Discrete Fourier Transform (DFT)**: applicable to **discrete-time** signals. The resulting **spectrum** is **discrete** and **periodic**.

[91] For periodic signals, the resulting FT spectrum is discrete.

Fast Fourier transform

- Cooley and Tukey are the 2 engineers credited for the **invention** of the modern **Fast Fourier Transform** (FFT), in an article published in 1965.
- Although the development of fast algorithms for DFT can be traced back to Gauss's work in 1805, the authors realised that Gauss's methods could be implemented efficiently with computers.
- The FFT is **not** a new transform but rather the **efficient implementation** of the **DFT**.
- The FFT can be used to compute the inverse DFT with minor modifications, i.e. a change of sign.

Fast Fourier transform

- As opposed to the DTFT, the **DFT** can **easily** be implemented on a **processor**.
- An implementation of the DFT based on its definition is not efficient because it does not use all symmetry properties of the DFT.
- The FFT is an efficient implementation of the DFT that uses these symmetries:
  - Evaluating the DFT's sums involves N^2 complex multiplications.
  - The well-known radix-2 Cooley–Tukey FFT algorithm requires $N \log_2 N$ complex multiplications[92].
  - The gain in complexity is approximately $\frac{N}{\log_2 N}$. For $N = 1024$, the **gain in efficiency is around 100**.
  - The practical implementation of the FFT requires N to be a **power of 2**.

[92]We have ignored the number of complex additions for simplicity.

Applications of the FFT

- Efficient discrete approximation of the continuous Fourier transform for **spectral analysis**.
- Computation of the **convolution of time sequences** of length M and K using the property $\mathcal{F}\left[h[k] * x[k]\right] = H(\Omega)X(\Omega)$.
  - First the DFT is used. The resulting spectra are multiplied. Finally, the inverse DFT is used.
  - The gain is substantial for long sequences.
  - **Zero padding** is used to make sure that the 2 sequences have a minimum length $N \geq M + K - 1$. N is chosen as the first power of 2 that verifies the condition.
- **Denoising**: noise can often be detected and removed more easily in the **frequency domain**.

Discrete Fourier transform

F.Y.I.

Discrete Fourier transform

The Discrete Fourier Transform (DFT) of a periodic signal of period N is

$$F[n] = \sum_{k=0}^{N-1} f[k]\, W_N^{kn}, \;\; 0 \leq n \leq N-1.$$

The inverse DFT is

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n]\, W_N^{-kn}, \;\; 0 \leq k \leq N-1$$

with

$$W_N = e^{-j\frac{2\pi}{N}}.$$

Symmetry and periodicity of W_N

F.Y.I.

Properties:

$$W_N^{k+N} = e^{-j\frac{2\pi(k+N)}{N}} = e^{-j\frac{2\pi k}{N}} e^{-j\frac{2\pi N}{N}} = e^{-j\frac{2\pi k}{N}} = W_N^k$$

$$W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi(k+\frac{N}{2})}{N}} = e^{-j\frac{2\pi k}{N}} e^{-j\frac{\pi N}{N}} = -e^{-j\frac{2\pi k}{N}} = -W_N^k$$

$$W_N^{Nn} = e^{-j\frac{2\pi(Nn)}{N}} = e^{-j2\pi n} = 1$$

$$W_N^{2kn} = e^{-j\frac{2\pi}{N}2kn} = e^{-j\frac{2\pi}{\frac{N}{2}}kn} = W_{\frac{N}{2}}^{kn} \quad \text{(N even)}$$

The elementary DFT of length $N = 1$ is

$$F[0] = \sum_{k=0}^{N-1} f[k]\, W_N^{kn} = f[0].$$

Towards an efficient implementation of the DFT

F.Y.I.

- Choose a value of $N = r^l$.
- Often $r = 2$ is selected $\Longrightarrow$ **radix-2** algorithm[93].
- The radix-2 decimation-in-time algorithm rearranges the DFT equation into two parts: a sum over the even-numbered discrete-time indices and a sum over the odd-numbered indices.
- Efficient use of memory known as "in-place computation": FFT computations are normally performed in place in a one-dimensional array, with new values overwriting old values.

[93] The meaning of radix is "base", a Latin word meaning "root".

Radix-2 FFT algorithm

F.Y.I.

Suppose $N = 2^l$. Decompose the DFT sum into 2 parts corresponding to, respectively, the even and odd indices of the sequence $f[k]$

$$
\begin{aligned}
\underbrace{F[n]}_{\substack{\text{DFT over } N \\ \text{samples}}} &= \sum_{k=0}^{N-1} f[k] W_N^{kn} \\
&= \sum_{k=0}^{\frac{N-1}{2}} f[2k] W_N^{2kn} + \sum_{k=0}^{\frac{N-1}{2}} f[2k+1] W_N^{(2k+1)n} \\
&= \sum_{k=0}^{\frac{N-1}{2}} f[2k] W_N^{2kn} + W_N^n \sum_{k=0}^{\frac{N-1}{2}} f[2k+1] W_N^{2kn} \\
&= \underbrace{\sum_{k=0}^{\frac{N-1}{2}} f[2k] W_{\frac{N}{2}}^{kn}}_{\substack{\text{DFT over } \frac{N}{2} \\ \text{samples} \\ \text{even indices}}} + W_N^n \underbrace{\sum_{k=0}^{\frac{N-1}{2}} f[2k+1] W_{\frac{N}{2}}^{kn}}_{\substack{\text{DFT over } \frac{N}{2} \\ \text{samples} \\ \text{off indices}}}
\end{aligned}
$$

Radix-2 FFT algorithm: recursion

F.Y.I.

Idea: recursive use of the relation $\Longrightarrow$ decompose the DFT of length $N = 2^l$ in l steps in order to obtain 2^l DFTs of length 1.

In step i, $F_i[n]$ is a DFT of length L and therefore $F_{ie}[n]$ and $F_{io}[n]$ are DFTs of length ${}^L\!/_2$

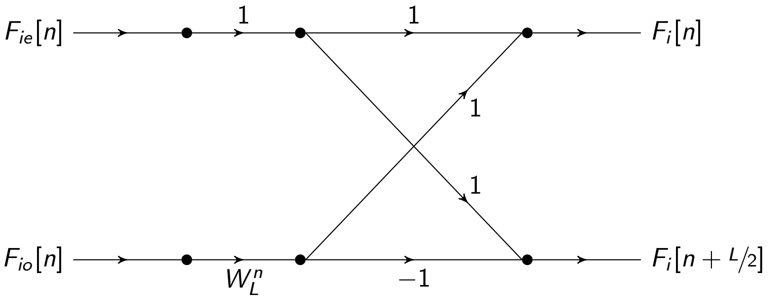
$$\begin{aligned} F_i[n] &= F_{ie}[n] + W_L^n \, F_{io}[n] \\ F_i[n + {}^L\!/_2] &= F_{ie}[n + {}^L\!/_2] + W_L^{(n+{}^L\!/_2)} \, F_{io}[n + {}^L\!/_2] \\ &= F_{ie}[n] + W_L^{(n+{}^L\!/_2)} \, F_{io}[n] \\ &= F_{ie}[n] - W_L^n \, F_{io}[n] \end{aligned}$$

We obtain the elementary **butterfly**:

$$\boxed{\begin{aligned} F_i[n] &= F_{ie}[n] + W_L^n \, F_{io}[n] \\ F_i[n + {}^L\!/_2] &= F_{ie}[n] - W_L^n \, F_{io}[n] \end{aligned}}$$

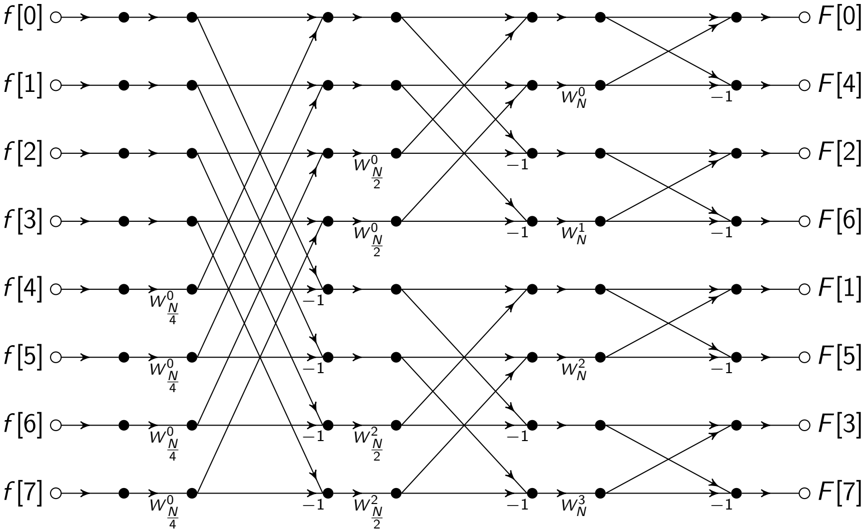
Radix-2 FFT algorithm: butterfly

F.Y.I.



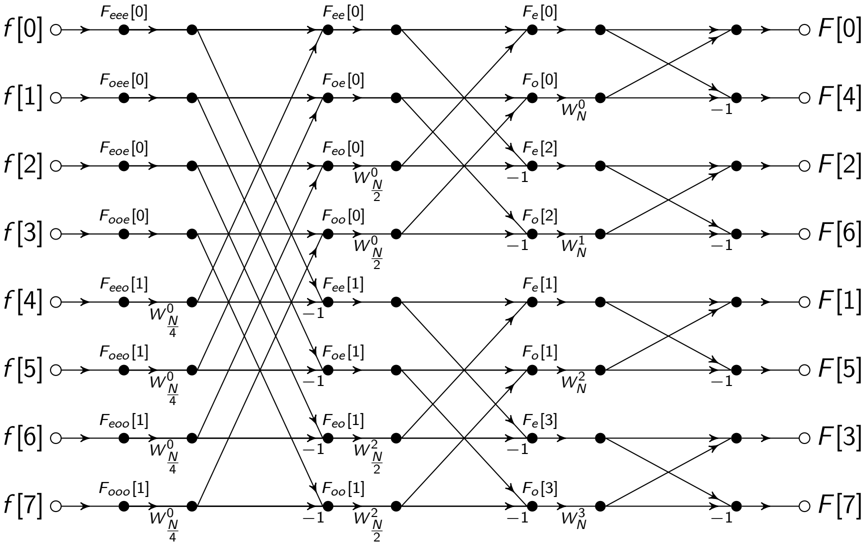
Radix-2 FFT algorithm: $N = 8$

F.Y.I.



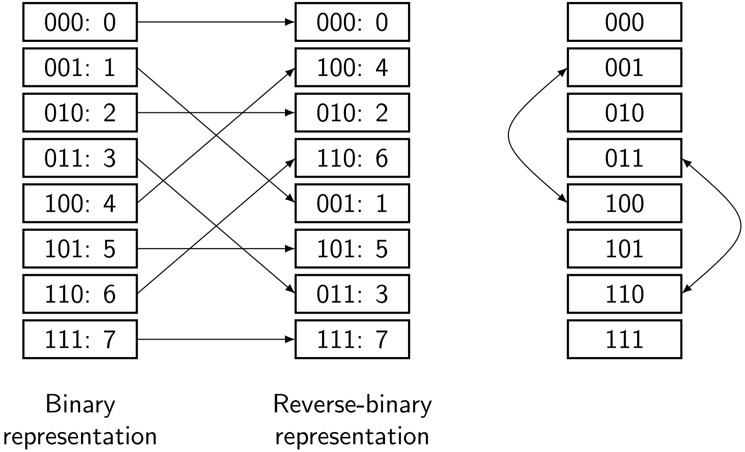
Radix-2 FFT algorithm: $N = 8$

F.Y.I.



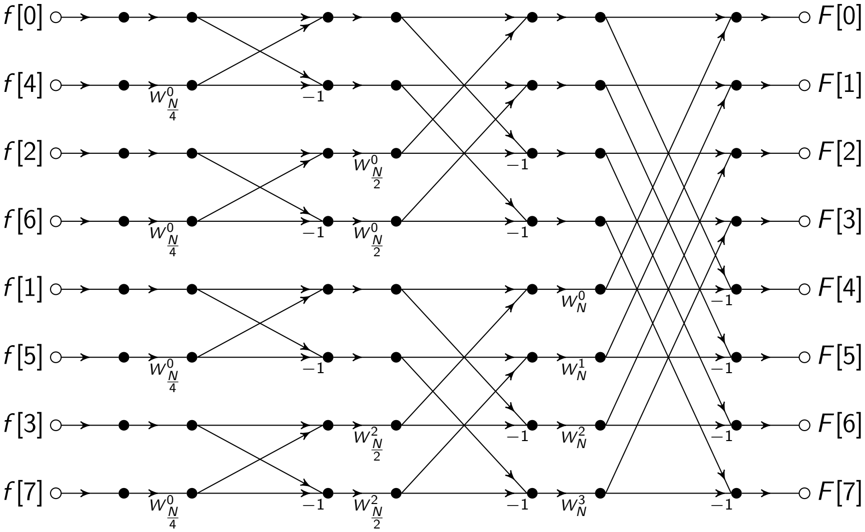
Radix-2 FFT algorithm: reverse-binary representation

F.Y.I.



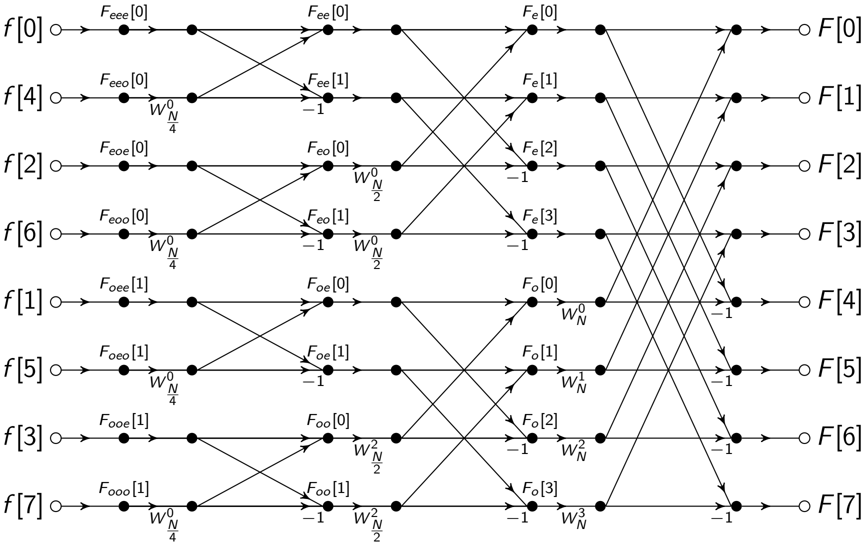
Radix-2 FFT algorithm: $N = 8$ with reorganisation

F.Y.I.



Radix-2 FFT algorithm: $N = 8$ with reorganisation

F.Y.I.



Radix-2 FFT algorithm: summary

F.Y.I.

- **Rearrange** the order of the N time domain samples by counting in binary with the bits flipped left-for-right.
- Find the frequency spectra of the **1 point time domain signals**. This step is trivial as the frequency spectrum of a 1 point signal is equal to itself.
- **Combine** the N frequency spectra in the exact reverse order that the time domain decomposition took place. The basic **butterfly pattern** is **repeated** over and over.

Radix-2 FFT algorithm

F.Y.I.

```
function [fft] = fft_comp(data,L)
%
% Franky De Bruyne
%
% Data is a column (!!!) vector with complex data
%
% N = 2^L
% if data has less than N points, it is padded with zeroes
% if data has more than N points, it is truncated to N points

% Make length a power of 2
% ------------------------

N = pow2(L);
N1 = length(data);
if (N1 < N)
    temp = [data; zeros(N-N1,1)];
elseif (N1 > N)
    temp = data(1:N);
else
    temp = data;
end
```

Radix-2 FFT algorithm

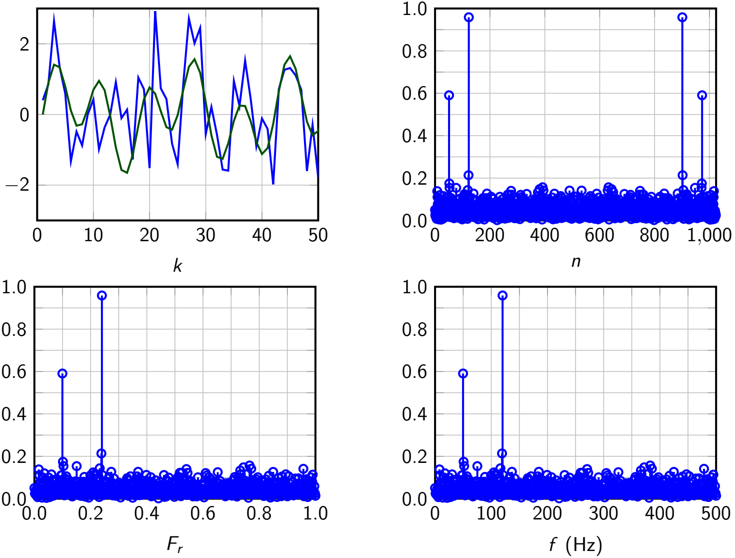
F.Y.I.

```
% Bit reversal routine
% ---------------------

for j=0:N-1
    nj = bin2dec(fliplr(dec2bin(j,L)));
    if ((j ~= nj) && (j < nj))
       tmp = temp(j+1);
       temp(j+1) = temp(nj+1); temp(nj+1) = tmp;
    end
end

% FFT computation
% ---------------

for j = 1:L     % Loop for each stage, i.e. DFTs with increasing data size
    theta = -2*pi/pow2(j);      % DFT of size 2^j
    Wj = exp(theta*1i); W = 1+0*1i;
    for k = 1:pow2(j-1)         % Loop for each sub DFT
        for l = k:pow2(j):N     % Implementation of butterfly
            tmp = temp(l+pow2(j-1))*W;
            temp(l+pow2(j-1)) = temp(l) - tmp; temp(l) = temp(l) + tmp;
        end
        W = W*Wj;
    end
end

fft = temp;
```

Noisy sinusoid

```
Fs = 1000;                  % Sampling frequency
T = 1/Fs;                   % Sample time
t = (0:999)*T;t = t';       % Time vector
t = t';
N = 2^10;

yr = randn(size(t));        % Random noise
k50 = 0.7; k120 = 1;        % Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
ys = k50*sin(2*pi*50*t) + k120*sin(2*pi*120*t);
yrs = yr + ys;

figure(1)
plot([yrs ys]);
axis([0 50 -3 3]), grid on, xlabel('k')

%X = fft_comp(yrs',10);  % own function: see previous slides
X = fft(yrs,N);             % built in function
f = (0:Nfft/2)*(Fs/Nfft);
ffa = (2/Nfft)*abs(X(1:Nfft/2+1));
ffa(1) = ffa(1)/2;

figure(2)
stem(f,(ffa))
axis([0 Fs/2 0 1]), grid on, xlabel('f (Hz)')
ylim([0 1.1])
```

Noisy sinusoid



Convolution using the FFT

```
x = [1 -3 1 5];              % Length of convolution product
y = [4 3 2 1];               % 4 + 4 - 1 = 7
x_padded = [x zeros(1,3)];   % => Zero padding (3 samples)
y_padded = [y zeros(1,3)];
X = fft(x_padded,8);         % FFT
Y = fft(y_padded,8);         % FFT
Z = X.*Y;                    % Point to point multiplication
z = ifft(Z,8);               % Inverse FFT
```

Convolution using the FFT

Give $x[k] = \{1, -3, 1, 5\}$ and $h[k] = \{4, 3, 2, 1\}$, compute their **convolution** using the DFT.

The resulting convolution sequence will have a **support** of $4 + 4 - 1 = 7$. It is therefore necessary to **add 3 zeros** at the end of each sequence. The **DFT** of each sequence yields

$$X[n] = \{4.0, -5.60 - j0.80, 3.88 + j7.27, 3.21 - j2.79, 3.21 + j2.79, 3.88 - j7.27, -5.60 + j0.80\},$$

$$H[n] = \{10, 4.52 - j4.73; 2.15 - j1.28; 2.32 - j0.72, 2.32 + j0.72, 2.15 + j1.28, 4.52 + j4.73\}.$$

Point to point multiplications yields

$$Y[n] = \{40, -29.11 + j22.86, 17.63 + j10.72, 5.47 - j8.77, 5.47 + j8.77, 17.63 - j10.70, -29.11 - j22.86\}.$$

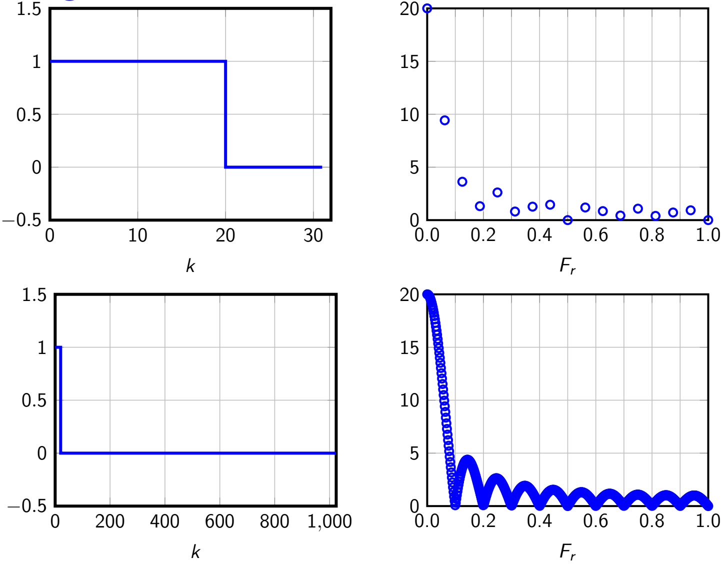
Finally, the **inverse DFT** gives

$$y[k] = \{4, -9; -3; 18; 14; 11; 5\}.$$

Zero padding

```
x1 = [ones(20,1); zeros(12,1)];
fft1 = fft(x1,32);
x2 = [x1;zeros(992,1)];
fft2 = fft(x2,1024);

subplot(2,2,1)
k1 = 0:1:31;
plot(k1',x1)
axis([0 32 -0.5 1.5]), xlabel('k')

subplot(2,2,2)
plot(k1/16,abs(fft1),'o')
hold on
plot(k1/16,abs(fft1))
axis([0 1 0 20]), xlabel('$F_r$','interpreter','latex')

subplot(2,2,3)
k2 = 0:1:1023;
plot(k2',x2)
axis([0 1024 -0.5 1.5]), xlabel('k')

subplot(2,2,4)
plot(k2/512,abs(fft2),'o')
hold on
plot(k2/512,abs(fft2))
hold off
axis([0 1 0 20]), xlabel('$F_r$','interpreter','latex')
```

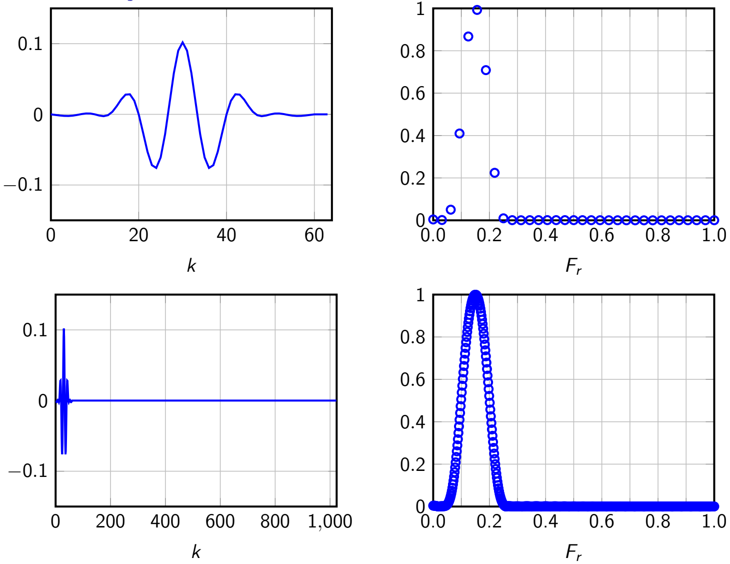
Zero padding



Spectral analysis

```
h_band_pass = fir1(60,[0.1 0.2]);    % FIR bandpass filter: pkg load signal
imp = zeros(64,1); imp(1) = 1;       % if using Octave with signal toolbox
x1 = filter(h_band_pass,1,imp);      % Impulse response
fft1 = fft(x1,64);                   % FFT N=64
x2 = [x1;zeros(960,1)];
fft2 = fft(x2,1024);                 % FFT N=1024

subplot(2,2,1)
k1 = 0:1:63;
plot(k1',x1)
axis([0 64 -0.15 0.15]), xlabel('k')

subplot(2,2,2)
plot(k1/32,abs(fft1),'o'); hold on
plot(k1/32,abs(fft1)); hold off
axis([0 1 0 1]), xlabel('F')

subplot(2,2,3)
k2 = 0:1:1023;
plot(k2',x2)
axis([0 1024 -0.15 0.15]), xlabel('k')

subplot(2,2,4)
plot(k2/512,abs(fft2),'o'); hold on
plot(k2/512,abs(fft2)); hold off
axis([0 1 0 1]), xlabel('$F_r$','interpreter','latex')
```

Spectral analysis



10. Getting started with ...

Introduction
Variables
Script files
Strings
Graphics
Function files
Programming
Polynomials
Numerical methods
Symbolic computations

Getting started with ...

Matlab

www.mathworks.com

registered trademarks of The MathWorks, Inc.

Octave

www.octave.org

distributed under the terms of the GNU General Public License

Scilab

www.scilab.org

open source software distributed under CeCILL license

Simulink

Xcos

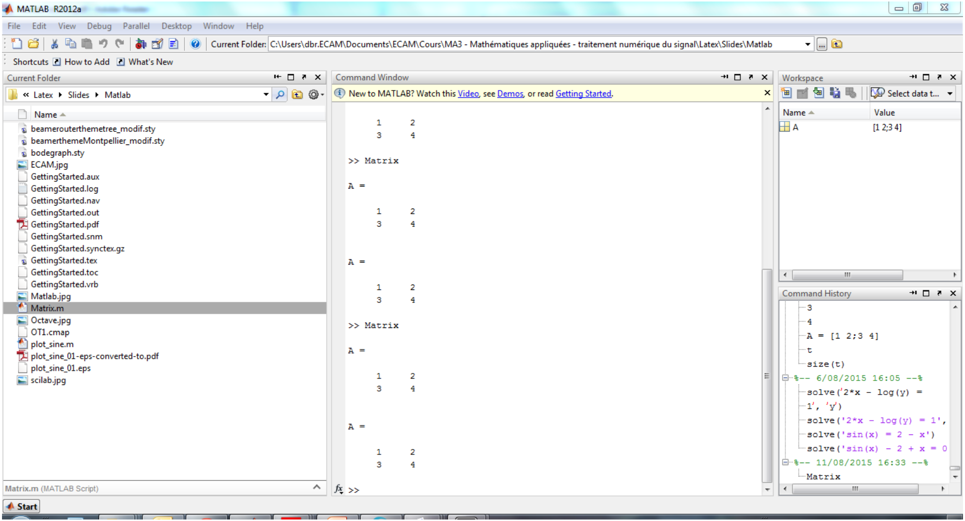
Getting started with ...

Matlab, Octave and Scilab are high-level languages and mathematical programming environments for:

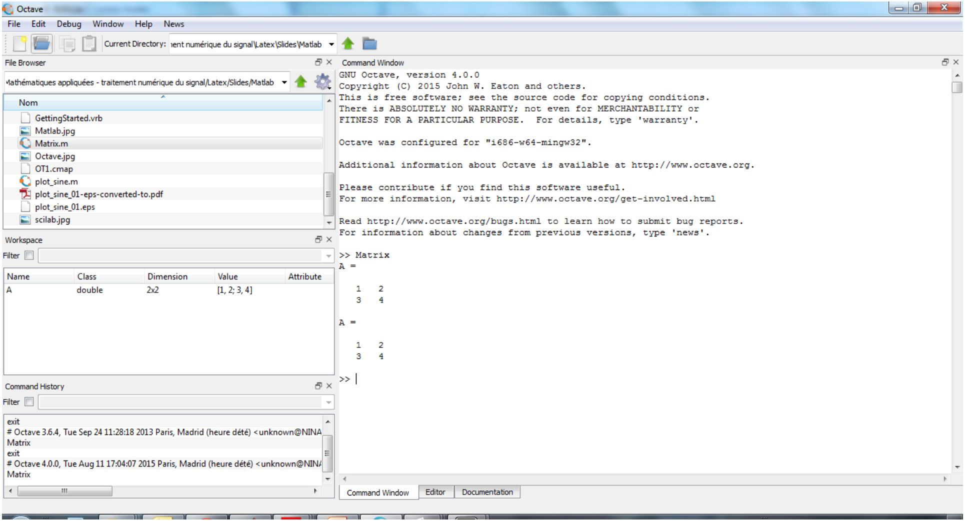
- Visualization
- Programming, algorithm development, prototyping
- Scientific computing: linear algebra, optimization, control, statistics, signal and image processing, etc.

This tutorial applies to Matlab, Octave and Scilab unless stated otherwise !

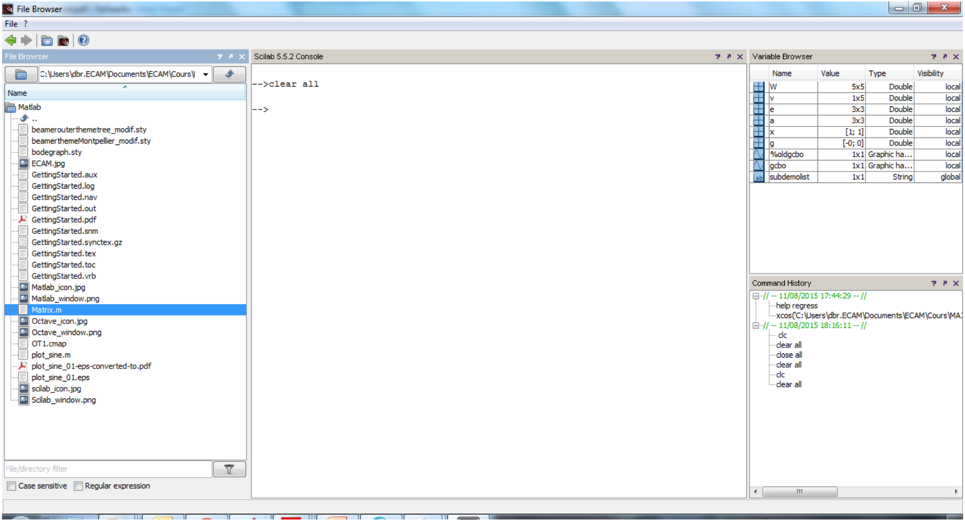
Matlab window (Matlab 2012a)



Octave window (version 4.0.0)



Scilab window (version 5.5.2)



Working environment

- Command window
- Command history
- Workspace / Variable browser
- Current Directory / File browser

Getting help

To get help on a specific command (= built-in function), use the `help` command.

```
help plot

PLOT   Linear plot.
    PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
    then the vector is plotted versus the rows or columns of the matrix,
    whichever line up.  If X is a scalar and Y is a vector, disconnected
    line objects are created and plotted as discrete points vertically at
    X.

    PLOT(Y) plots the columns of Y versus their index.
    If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
    In all other uses of PLOT, the imaginary part is ignored.

    Various line types, plot symbols and colors may be obtained with
    PLOT(X,Y,S) where S is a character string made from one element
    from any or all the following 3 columns:

           b     blue          .     point              -     solid
```

Input and output

Input commands are entered in the command window. The output is returned in two ways:

- typically, **text or numerical output** is returned in the same **command window**,
- but **graphical output** appears in a separate **figure window**.

Arithmetics

Matlab, Octave and Scilab can be used to do arithmetic as we would with a calculator. You can add with +, subtract with −, multiply with ∗, divide with /, and exponentiate with ^.

For example:

```
3^2 - (5 + 4)/2 + 6*3


ans =

   22.5000
```

The answer is assigned to a variable called `ans`. If we want to perform further calculations with the answer, we can use the variable `ans` rather than retype the answer.

Arithmetics

A number of **elementary functions** are available. Look for the associated help functionalities or google

"elementary functions program"

with program replaced by Matlab, Octave or Scilab.

For example, the Matlab help yields:

```
help elfun
```

```
  Elementary math functions.

  Trigonometric.
    sin         - Sine.
    sind        - Sine of argument in degrees.
    sinh        - Hyperbolic sine.
    asin        - Inverse sine.
```

Use of the semicolon (;)

If we want to suppress and hide the output for an expression, we can add a semicolon (;) after the expression.

For example:

```
x = 3;
y = x + 5
```

```
y =

     8
```

Adding comments

The percent symbol (%) is used to indicate a comment line in .

Two consecutives slashes (//) are used in .

For example:

```
x = 3;      % Assign the value 3 to x
y = x + 5   % Add 3 to x and assign the result to y
```

```
y =

     8
```

Retrieving previous commands

- To select commands in the command history, press the up-arrow and down-arrow keys, $\uparrow$ and $\downarrow$, in the command window.
- To retrieve a command using a partial match, type any part of the command at the prompt, and then press the $\uparrow$ key.
- A new command can sometimes be entered more efficiently by modifying a previous command.

Saving the workspace

The `save` command is used for saving all the variables in the workspace in the current directory. The extension is program dependent. For example:

```
save myfile
```

You can reload the file anytime later using the `load` command.

```
load myfile
```

Variables

In the considered environments, every variable is an array or matrix.

You can assign variables in a simple way. For example:

```
x = 3
```

```
x =

     3
```

It creates a 1-by-1 matrix named `x` and stores the value 3 in its element.

Variables

Another example:

```
x = sqrt(16)
```

```
x =

     4
```

Please note that:

- Once a variable is entered into the system, we can refer to it later.
- Variables must have values before they are used.
- When an expression returns a result that is not assigned to any variable, the system assigns it to a variable named `ans`, which can be used later.

Variables: useful commands

The `who` command displays all the variable names we have used. The `whos` command displays more informations about the variables:

```
whos
```

```
Name      Size              Bytes  Class     Attributes

A          2x2                 32  double
ans        1x1                  8  double
x          2x1                112  sym
y          2x1                112  sym
```

Variables: useful commands

The `clear` command deletes all (or the specified) variable(s) from the memory. For example:

```
x = 3; y = 4;
who
clear x
who
clear
who
```

```
Your variables are:

x  y

Your variables are:

y
```

Variables: useful commands

By default, numbers are displayed with four decimal place values. This is known as the short format.

```
format short
x = 7 + 10/3 + 5^1.2
```

```
x =

  17.2320
```

However, if we want more precision, we need to use the `format` command. The format long command displays 16 digits after decimal.

```
format long
x = 7 + 10/3 + 5^1.2
```

```
x =

  17.231981640639408
```

Creating vectors

Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.

For example:

```
x = [1 2 3]
y = [1,2,3]
```

```
x =

     1     2     3

y =

     1     2     3
```

Creating vectors

Column vectors are created by enclosing the set of elements in square brackets, using semicolon(;) to delimit the elements.

For example:

```
x = [1; 2; 3]
```

```
x =

     1
     2
     3
```

Creating matrices

A matrix is a two-dimensional array of numbers.

A **matrix** is created by entering each row as a sequence of space or comma separated elements; and end of a row is demarcated by a semicolon.

For example:

```
x = [1 2 3;4 5 6;7 8 9]
```

```
x =

     1     2     3
     4     5     6
     7     8     9
```

Accessing parts of a matrix

Accessing element (i, j), i.e. element in row i and column j.

For example:

```
x = [1 2 3;4 5 6;7 8 9];
y = x(2,3)
```

```
y =

     6
```

Accessing parts of a matrix

Accessing a row.

For example:

```
x = [1 2 3;4 5 6;7 8 9];
y = x(2,:)
```

```
y =

     4     5     6
```

Accessing parts of a matrix

Accessing a column.

For example:

```
x = [1 2 3;4 5 6;7 8 9];
y = x(:,3)
```

```
y =

     3
     6
     9
```

Accessing parts of a matrix

Accessing a given range.

For example:

```
x = [1 2 3;4 5 6;7 8 9];
y = x(1:2,2:3)
```

```
y =

     2     3
     5     6
```

Combinations of matrices

Combining matrices.

For example:

```
x = [1 2 3;4 5 6;7 8 9];
y = [3; 4; 9];
z = [0 9 8 7];
A = [x y; z]


A =

     1     2     3     3
     4     5     6     4
     7     8     9     9
     0     9     8     7
```

Matrix operations

Matlab, Octave and Scilab can be used to do **matrix operations**.

Assuming compatibily, we can add with +, subtract with −, multiply with ∗. The operator `.*` is used to do multiplication element by element.

For example:

```
A = [1 2;3 4]; B = [1; 2]; C = [1 2];
X = A^2
Y = A*B
Z = C*A
```

```
X =

    7    10
   15    22
Y =

    5
   11
Z =

    7    10
```

Matrix transpose

The operator ' is used for **matrix transposition**.

For example:

```
B = [1 2;3 4;5 6]
C = B'
```

```
B =

     1     2
     3     4
     5     6

C =

     1     3     5
     2     4     6
```

Matrices functions

Matrix functions.

For example:

```
x = [1 2 3;4 0 6;7 8 9];
d = det(x)
ix = inv(x)

d =

   60.0000

ix =

   -0.8000    0.1000    0.2000
    0.1000   -0.2000    0.1000
    0.5333    0.1000   -0.1333
```

Matrices functions

A number of **matrix functions** are available. Look for the associated help functionalities or google

"matrix functions program"

with program replaced by Matlab, Octave or Scilab.

For example, the Matlab help yields:

```
help matfun
```

```
  Matrix functions - numerical linear algebra.

  Matrix analysis.
    rank        - Matrix rank.
    det         - Determinant.
    trace       - Sum of diagonal elements.

  Linear equations.
    \ and /     - Linear equation solution; use "help slash".
    inv         - Matrix inverse.
```

Special matrices

Matrices of ones, matrices of zeros and identity matrices can easily be constructed using the commands `ones`, `zeros` and `eye`.

For example:

```
A = eye(3)
B = eye(2,3)
```

```
A =

     1     0     0
     0     1     0
     0     0     1

B =

     1     0     0
     0     1     0
```

Diagonal matrices

Diagonal matrices can be constructed using the command `diag`.

For example:

```
diag([1 2 3])
```

```
ans =

     1     0     0
     0     2     0
     0     0     3
```

Vector and matrix size

The size of a matrix array can be obtained using the command `size`.
The length of a vector can be ontained using the command `length`.

For example:

```
a = [1 2];
length(a)
B = eye(2,3);
size(B)
```

```
ans =

     2

ans =

     2     3
```

Complex computations

Complex numbers can be constructed using the function `complex`. A number of functions like real, `imag`, `abs` and `conj` can be used.

For example:

```
a = complex(1,1);
real(a)
abs(a)
conj(a)
```

```
ans =

     1

ans =

    1.4142

ans =

   1.0000 - 1.0000i
```

Complex computations

Complex numbers can also be entered directly. The function `angle` can also be used.

For example:

```
a = 1 + 1i;
real(a)
abs(a)
angle(a)
```

```
ans =

     1

ans =

    1.4142

ans =

    0.7854
```

Script files

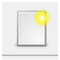
- For simple problems, entering your requests at the prompt is fast and efficient.
- However, as the number of commands increases typing the commands over and over at the prompt becomes tedious.
- Script files are the **main tool** for writing code.

Script files

In order to create and run a script file, we need to:

- Open a new script file

Matlab



Octave

Scilab

- Give it a name. Be sure the name is not an existing function !
- Write your instructions inside the file. Write comments in your program !
- Save it in the current directory.
- "Call it", i.e. type the file name on the command window.

Strings

- Strings in MATLAB are written between **single quotes**.
- A quotation within the string is indicated by two quotation marks.
- **Concatenation** of strings is done by using **square brackets**.
- The function `num2str` is used to convert a numerical result into a string.
- **Reading from the keyboard** can be accomplished by using the `input` function.

Strings

```
clear all
str = 'Example''s result';
a = input('Enter a real value to be multiplied by 2: ');
a = 2*a;
str = [str ': ' num2str(a)];
disp(str)
whos
```

```
Enter a real value to be multiplied by 2: 3.3
Example's result: 6.6

  Name      Size            Bytes  Class     Attributes

  a         1x1                 8  double
  str       1x21               42  char
```

Plotting

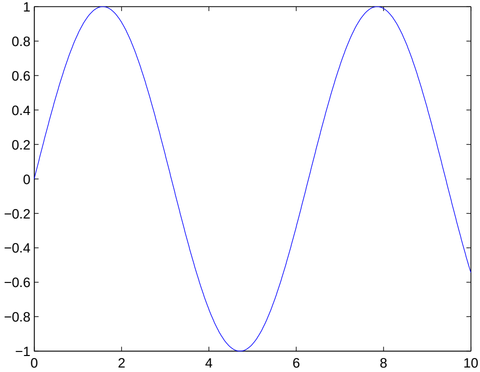
To plot the graph of a function, we need the following steps:

- Define `x`, by specifying the range of values for the variable `x`, for which the function is to be plotted.
- Define the function, `y = f(x)`.
- Call the plot command as `plot(x,y)`.

Plotting

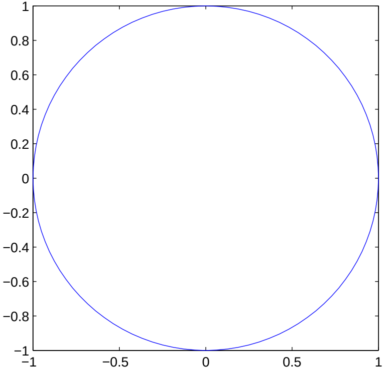
Create a script file and type the following code and execute it !

```
x = 0:0.01:10;
y = sin(x);
plot(x,y);
```



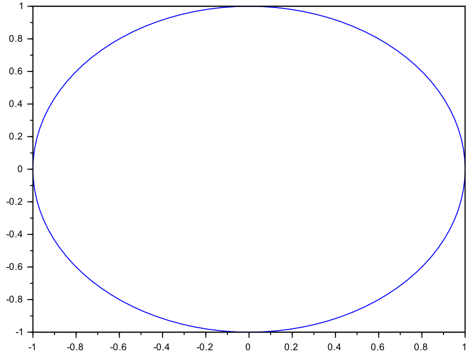
Parametric plot

```
t = 0:0.01:1;
plot(cos(2*pi*t),sin(2*pi*t))
axis square
```



Parametric plot

```
t = 0:0.01:1;
plot(cos(2*%pi*t),sin(2*%pi*t))
```



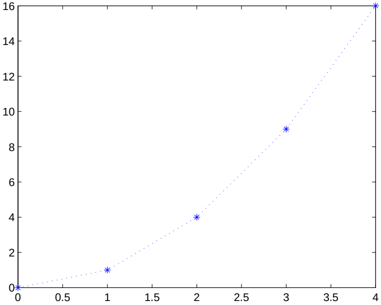
help plot

```
help plot
```

```
PLOT   Linear plot.
   PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
   then the vector is plotted versus the rows or columns of the matrix,
   whichever line up.  If X is a scalar and Y is a vector, disconnected
   line objects are created and plotted as discrete points vertically at X.

   Various line types, plot symbols and colors may be obtained with
   PLOT(X,Y,S) where S is a character string made from one element
   from any or all the following 3 columns:

          b     blue          .     point              -     solid
          g     green         o     circle             :     dotted
          r     red           x     x-mark             -.    dashdot
          c     cyan          +     plus               --    dashed
          m     magenta       *     star             (none)  no line
          y     yellow        s     square
          k     black         d     diamond
          w     white         v     triangle (down)
                              ^     triangle (up)
```

Plotting points

```
x = [0 1 2 3 4];
y = [0 1 4 9 16];
plot(x,y,'b:*')
```

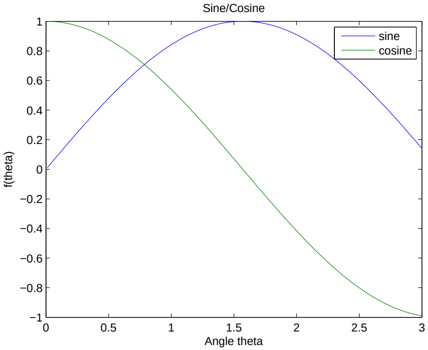


Plotting: summary of useful commands

- `figure` - opens window
- `hold on, hold off` - between these commands everything is plotted in the same window
- `grid on, grid off` - add or remove grid lines
- `subplot` - create several different plots in one figure
- `xlabel, ylabel` - adds text beside the X-axis/Y-axis on the current axis
- `title` - adds text at the top of the current axis
- `legend` - display legend
- `axis` - control axis scaling and appearance

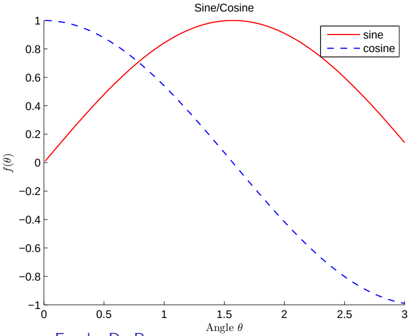
Several plots

```
figure
x = 0.01:0.01:3;
y1 = sin(x); y2 = cos(x);
plot(x,[y1; y2]);
legend('sine','cosine');
xlabel('Angle theta');
ylabel('f(theta)');
title('Sine/Cosine');
```



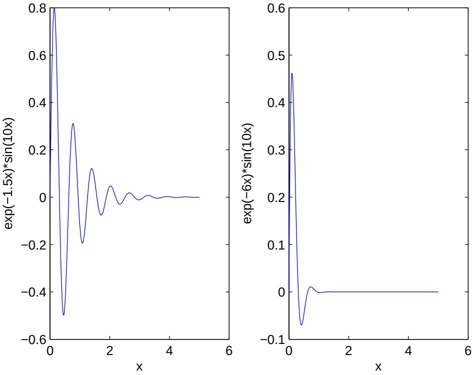
Several plots with `hold` function

```
figure, hold on
x = 0.01:0.01:3;
y1 = sin(x); y2 = cos(x);
plot(x,y1,'-r','LineWidth',1);
plot(x,y2,'--b','LineWidth',1);
legend('sine','cosine');
xlabel('Angle $\theta$','interpreter','latex');
ylabel('$f(\theta)$','interpreter','latex');
title('Sine/Cosine');
hold off
```



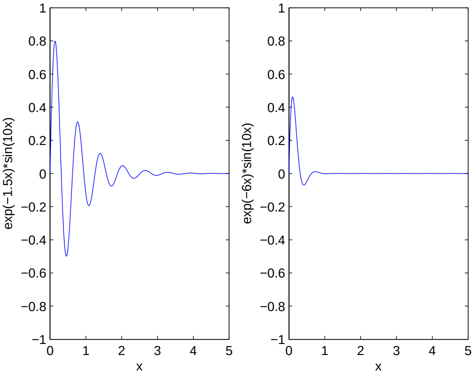
Subplots

```
x = 0:0.01:5;
y = exp(-1.5*x).*sin(10*x);
subplot(1,2,1)
plot(x,y), xlabel('x'), ylabel('exp(-1.5x)*sin(10x)')
y = exp(-6*x).*sin(10*x);
subplot(1,2,2)
plot(x,y),xlabel('x'), ylabel('exp(-6x)*sin(10x)')
```



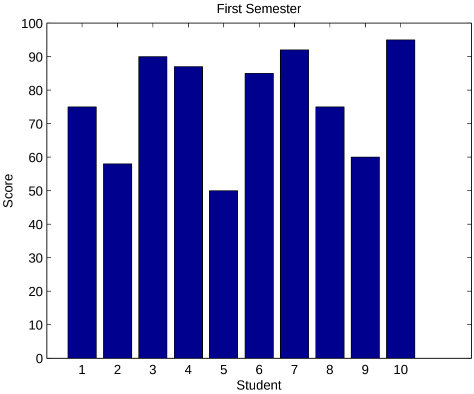
Subplots with axis function

```
x = 0:0.01:5;
y = exp(-1.5*x).*sin(10*x);
subplot(1,2,1)
plot(x,y), xlabel('x'), ylabel('exp(-1.5x)*sin(10x)'), axis([0 5 -1 1])
y = exp(-6*x).*sin(10*x);
subplot(1,2,2)
plot(x,y),xlabel('x'), ylabel('exp(-6x)*sin(10x)'), axis([0 5 -1 1])
```



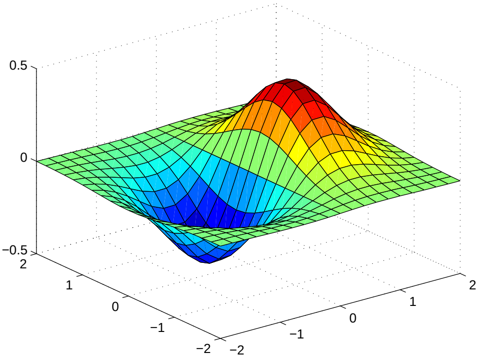
Bar charts

```
x = 1:10;
y = [75,58,90,87,50,85,92,75,60,95];
bar(x,y), xlabel('Student'), ylabel('Score'),
title('First Semester')
```



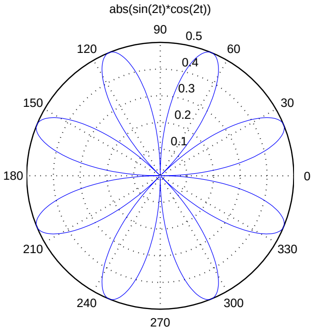
3D surface map

```
[x, y] = meshgrid(-2:.2:2);
g = x .* exp(-x.^2- y.^2);
surf(x, y, g)
```



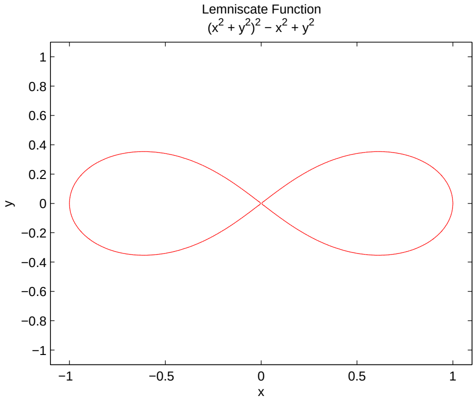
Polar plots

```
% Create data for the function
t = 0:0.01:2*pi;
r = abs(sin(2*t).*cos(2*t));

% Create a polar plot using polar
figure;
polar(t, r);

% Add a title
title('abs(sin(2t)*cos(2t))');
```



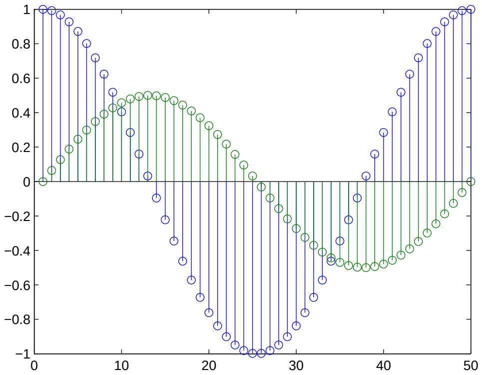
Plotting implicitly defined functions

```
figure;
h = ezplot('(x^2 + y^2)^2 - x^2 + y^2',[-1.1, 1.1], [-1.1, 1.1]);
set(h,'color','red')
grid on

title({'Lemniscate Function';'(x^2 + y^2)^2 - x^2 + y^2'});
```



Discrete time plots

```
figure
X = linspace(0,2*pi,50)';
Y = [cos(X), 0.5*sin(X)];
stem(Y)
```



Function files

- Function files play the role of **user defined commands** that often have input and output.
- One can create own commands for specific problems this way, which will have the same status as other commands.

Function files

```
function [a] = log3(x)
% [a] = log3(x) - Calculates the base 3 logarithm of x.
a = log(abs(x))./log(3);
% End of function
```

```
help log3
log3(2)
```

```
[a] = log3(x) - Calculates the base 3 logarithm of x.

ans =

    0.6309
```

Looping with `for`

- Structure:

```
for var = expression,
    body;
end
```

- Nesting of `for` loops is allowed.

Looping with `for`

```
A = [1 5 -3;2 4 0;-1 6 9];
for i = 1:3
    for j = 1:3
        B(i,j) = A(i,j)^2;
    end
end
B
```

```
B =

     1    25     9
     4    16     0
     1    36    81
```

Looping with `while`

- Structure:

```
while condition,
    body;
end
```

- Nesting of `while` loops is allowed.

Looping with while

```
i = 1;
while (i < 3)
  disp(i);
  i = i + 1;
end
```

1

2

Branching with `if`

- Structure:

```
if condition
    then-body;
elseif condition
    elseif-body;
else
    else-body;
end
```

- The `else` and `elseif` clauses are optional.
- Any number of `elseif` clauses may exist.
- Nesting of `if` branches is allowed.

Branching with if

```
a = 100;
if a == 10
    disp('Value of a is 10');
elseif( a == 20 )
    disp('Value of a is 20');
elseif a == 30
    disp('Value of a is 30');
else
    disp('None of the values are matching');
end
```

```
None of the values are matching
```

Branching with `switch`

- Structure:

```
switch expression

case label
    command-list;
case label
    command-list;
...
otherwise
    command-list;
end
```

- Any number of case labels are allowed.

Branching with switch

```
grade = 'B';
switch(grade)

case 'A'
        disp('Excellent!');
case 'B'
        disp('Above average');
case 'C'
        disp('Average');
case 'D'
        disp('Acceptable');
case 'F'
        disp('Fail');
otherwise
        disp('Invalid grade');
end
```

```
Above average
```

Relational operators

- `x < y` true if x is less than y
- `x <= y` true if x is less than or equal to y
- `x == y` true if x is equal to y
- `x >= y` true if x is greater than or equal to y
- `x > y` true if x is greater than y
- `x ~= y` true if x is not equal to y

Boolean expressions

- `B1 & B2` Element-wise logical and
- `B1 | B2` Element-wise logical or
- `~B` Element-wise logical not

```
A = [0 1 1 0 1];
B = [1 1 0 0 1];
A & B
```

```
ans =

     0     1     0     0     1
```

Polynomials

- The polynomial
$$p(x) = x^4 + 7x^2 - x$$
is represented by `p = [1, 0, 7, -1, 0]`.
- An n-th order polynomial is represented by vector of length $n + 1$.
- if the polynomial is missing any coefficients, zeros must be entered in the appropriate place(s) in the vector, as done above.
- The value of a polynomial at a given x can be found using the `polyval` command.
- Finding the roots of a polynomial is done using the `root` command.
- The product of two polynomials is found by taking the convolution of their coefficients using the command `conv`.

Polynomials

```
p = [1 0 7 -1 0];
polyval(p,-1)
roots(p)
```

```
ans =

     9

ans =

        0
  -0.0712 + 2.6486i
  -0.0712 - 2.6486i
   0.1424
```

Polynomials

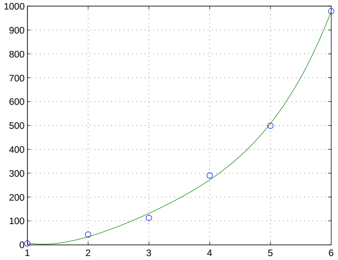
```
p = [1 0 -1];
q = [1 3];
r = conv(p,q)
roots(r)
```

```
r =

     1     3    -1    -3


ans =

   -3.0000
    1.0000
   -1.0000
```

Polynomial curve fitting

The `polyfit` function finds the coefficients of a polynomial that fits a set of data in a least-squares sense.

```
x =[1 2 3 4 5 6];
y = [5.5 43.1 112.8 290.7 498.4 978.7];

p = polyfit(x,y,4);

x2 = 1:.1:6;
y2 = polyval(p,x2);
plot(x,y,'o',x2,y2)
grid on
```



Numerical methods

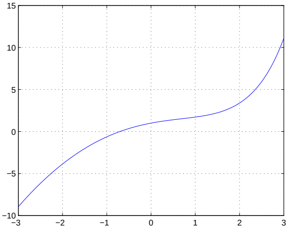
- Suppose we are interested in finding the **roots** of a general non-linear function. This can be done in through the command `fzero`, which is used to approximate the root of a function of one variable, given an initial guess.
- Another useful command is `fminsearch` which finds the **minimum of a function**.
- The function `quad` can be used for **numerical integration**.

Numerical methods

```
function [val] = func(x)
val = exp(x) - x.^2;
```

```
x=-3:.01:3;
plot(x,func(x));
grid
fzero('func',-0.5)
```

```
ans =

   -0.7035
```



Numerical methods: minimisation

```
fun = inline('100*(x(2)-x(1)^2)^2+(1-x(1))^2');
x0 = [5 5];
[x,fval,exitflag,output] = fminsearch(fun,x0)
```

```
x =

    1.0000    1.0000

fval =

   5.6197e-10

exitflag =

     1

output =

    iterations: 106
     funcCount: 201
     algorithm: 'Nelder-Mead simplex direct search'
       message: [1x194 char]
```

Numerical methods: integration

```
function [val] = func(x)
val = exp(x) - x.^2;
```

```
quad('func',0,1)
quad('sin',0,pi)
```

```
ans =

    1.3849

ans =

    2.0000
```

Symbolic computations

- Using Matlab's and Octaves symbolic toolbox, we can carry out algebraic or symbolic calculations such as factoring and expanding polynomials.
- Type `help symbolic` to **make sure** that the symbolic toolbox is **installed** in Matlab.
- Type `pkg load symbolic` to **make sure** that the symbolic toolbox is **installed** in Octave. The installation procedure is quite elaborate.
- Use `syms` or `sym` to declare to declare the variables that we plan to be symbolic variables.
- Use the commands `expand`, `factor`, `simplify`, `subs` to perform symbolic computations.
- The commands `diff` and `int` can be used to differentiate and integrate symbolic expressions.
- The commands `limit` and `taylor` to compute limits and Taylor series.

Symbolic computations

```
syms x y
factor(x^3- y^3)
expand((x+2)*(x-3)*(x-5)*(x+7))
simplify((x^4-16)/(x^2-4))
z = x^3- y^3;
subs(z, x, 1)
```

```
ans =

(x - y)*(x^2 + x*y + y^2)

ans =

x^4 + x^3 - 43*x^2 + 23*x + 210

ans =

x^2 + 4

ans =

1 - y^3
```

Symbolic computations

```
syms x
diff(sin(x^2))
int(x^3)

limit(abs(x)/x, x, 0,'left')
taylor(sin(x),x,pi/2,'Order',6)
```

```
ans =

2*x*cos(x^2)

ans =

x^4/4

ans =

-1

ans =

(pi/2 - x)^4/24 - (pi/2 - x)^2/2 + 1
```

Symbolic computations

```
pkg load symbolic
x = sym('x'); y = sym('y');
factor(x^3- y^3)
expand((x+2)*(x-3)*(x-5)*(x+7))
simplify((x^4-16)/(x^2-4))
z = x^3- y^3;
subs(z, x, 1)
```

```
ans = (sym)

          / 2          2\
  (x - y)*\x  + x*y + y /

ans = (sym)

   4    3       2
  x  + x  - 43*x  + 23*x + 210

ans = (sym)

   2
  x  + 4

ans = (sym)

     3
  - y  + 1
```

Symbolic computations

```
pkg load symbolic
x = sym ('x');
diff(sin(x^2))
int(x^3)
limit(abs(x)/x, x, 0,'left')
taylor(sin(x),x,pi/2,'Order',6)
```

```
ans = (sym)

         / 2\
  2*x*cos\x /

ans = (sym)

   4
  x
  --
  4

ans = (sym) -1

ans = (sym)

          4            2
  /    pi\    /    pi\
  |x - --|    |x - --|
  \    2 /    \    2 /
  --------- - --------- + 1
      24          2
```

11. Dictionary

Dictionary: E - FR

- aliasing (frequency folding) - repli spectral
- Analog-to-Digital Converter (ADC) - Convertisseur Analogique Numérique (CAN)
- band-pass filter - filtre passe bande
- Bounded-Input-Bounded-Output (BIBO) stability - Stabilité Entrée-Bornée-Sortie-Bornée (EBSB)
- cardinal sine - sinus cardinal
- cut-off frequency - fréquence de coupure ou de brisure
- to damp - amortir
- Digital-to-Analog Converter (DAC) - Convertisseur Numérique Analogique (CNA)
- digitalisation - numérisation

Dictionary: E - FR

- even - pair
- frequency folding (aliasing) - repli spectral
- high-pass filtre - filtre passe haut
- inner (scalar or dot) product - produit scalaire
- to lag - retarder, être en retard
- to lead - avancer, mener
- Least Significant Bit (LSB) - bit de poids faible
- low-pass filter - filtre passe bas
- odd - impair
- partial fraction expansion - décomposition en fractions simples
- piecewise constant - constant par morceaux
- quantisation - quantification

Dictionary: E - FR

- quantisation grid - grille de quantification
- recursive - récurrent
- region of convergence - domaine de convergence
- relaxed system - système au repos
- root - racine
- sampling - échantillonnage
- sifting property - propriété de localisation
- to span - engendrer
- spectral leakage - fuite spectral
- steady-state response - réponse de régime, régime permanent
- stop-band filter - filtre coupe bande
- time aliasing - repli temporel

Dictionary: E - FR

- wavelength - longueur d'onde
- zero-state response - réponse forcée
- zero-input response - réponse libre
- Zero-Order-Hold (ZOH) - bloqueur d'ordre zéro