Detailed SYLLABUS of the course

LABORATORY OF ADVANCED ELECTRONICS

Department of Physics, University of Trento, a.y. 2022–2023

Lecturers: **LEONARDO RICCI**, **ALESSIO PERINELLI**

(last updated on December 15, 2022)

LECTURES

(12/09/2022) 1. **Introduction to the course. Signals and systems. Sequences.**

- Signal processing:
  - analog signal processing;
  - digital signal processing (DSP).
- Noise (a very short review):
  - Johnson noise;
  - shot noise;
  - quantum noise.

- Signals:
  - a formal definition of signals (pictures included!);
  - continuous signals;
  - sequences (discrete–time signals).
- Sequences:
  - graphical representation;
  - remarkable sequences: impulse–sequence $\delta[n]$, step–sequence $u[n]$;
  - delayed sequences, relation between $\delta[n]$ and $u[n]$;
  - representation of a generic sequence by means of delayed impulse–sequences;
  - periodic sequences;
  - energy of a sequence.

- Systems:
  - a formal definition of systems.

(19/09/2022) 2. **LTI systems. Basic properties of LTI systems. Difference equations.**

- Linear, time–invariant (LTI) systems:
  - action on a sequence;
  - impulse–response (a.k.a. transfer function) $h[n]$, and convolution;
  - a comment on the symmetry between input $x[n]$ and impulse–response $h[n]$: their roles can be swapped.
- Properties of an LTI system:
  - reality;
  - causality;
  - marginal stability;
  - *bound–input, bound–output* ("BIBO") stability,
    * definition,
    * necessary and sufficient condition for BIBO stability;
  - examples ($h[n] = u[n]$, $h[n] = a^n u[n]$).

- Difference equations:
  - similarity with the continuous case: a difference equation characterizes an LTI system;
  - example,
    * $h[n]$ in the case of $y[n] = x[n] + x[n-1]$,
    * $h[n]$ in the case of $y[n] = a \cdot y[n-1] + b \cdot x[n]$.

(20/09/2022) 3. **z–transform and its inversion.**

- z–transform:
  - a short discussion on the importance of changing space, in analogy with continuous systems, to solve difference equations;
  - definition and region of convergence ("ROC");
  - remarkable examples,
    * $\delta[n]$,
    * $u[n]$,
    * $-u[-n-1]$.
- Basic properties of z–transform:
  - linearity (important: beware of intersecting ROCs!);
  - time–shift (important: beware of new ROC!);
  - convolution theorem (important: beware of intersecting ROCs!).
- Complementary topics: graphical representation of systems:

- linear combinations of systems;
- cascade systems, and invertibility of two systems (proof via z–transform).

- Summary of residue calculus:
  - $\frac{1}{2\pi i}\oint_{\Gamma\ um\ z_o}(z-z_o)^n dz = \delta_{n,\ -1}$;
  - $G(z)$ has an n^{th}–order pole in $z_o \Longrightarrow$
    $\frac{1}{2\pi i}\oint_{\Gamma\ um\ z_o} G(z)dz = \frac{1}{(n-1)!}\frac{d^{n-1}}{dz^{n-1}}[G(z)(z-z_o)^n]_{z=z_o}$.
- Inversion of the z–transform:
  - derivation of the inversionexpression;
  - examples,
    * $X(z) = 1$ with ROC $\mathbb{C}$,
    * $X(z) = \frac{z}{z-1}$ with ROC $|z| < 1$,
    * $X(z) = \frac{z}{z-1}$ with ROC $|z| > 1|$.

(27/09/2022) 4. **Basic properties of LTI systems as seen in the z–domain. Inversion of the z–transform: examples.**

- Basic properties of systems as seen in the z–domain:
  - reality,
    * reality $\Leftrightarrow \overline{X}(\overline{z}) = X(z)$,
    * examples,
    * existence of complex-conjugate zeroes and poles;
  - causality,
    * beware: causal $\neq \overline{\text{noncausal}}$ ($k \cdot \delta[n] \ldots$ is both!),
    * $x[n] = 0$ for $n < 0$ (causal) $\Leftrightarrow 0 \notin$ ROC, $\infty \subset$ ROC (implication $\Leftarrow$ via Taylor series),
    * $x[n] = 0$ for $n > 0$ (noncausal) $\Leftrightarrow 0 \in$ ROC, $\infty \not\subset$ ROC (implication $\Leftarrow$ via Taylor series),
    * $x[n] = 0$ for $n \neq 0$ (causual & noncausal; $x[n] = k \cdot \delta[n] \ldots$) $\Leftrightarrow$ $0 \in$ ROC, $\infty \subset$ ROC $\Leftrightarrow X(z)$ uniform on $\mathbb{C}$ (the implication $\Leftarrow$ corresponds to the Liouville theorem: a bounded holomorphic function whose ROC coincides with $\mathbb{C}$ is uniform),
    * examples;
  - BIBO stability,
    * BIBO stability $\Leftrightarrow \Gamma_1 \subset$ ROC,
    * examples.

- Remarkable examples concerning the inversion of the z–transform:

- basic example,
  - evaluation of the z–transform of $a^n u[n]$, $-a^n u[-n-1]$,
  - $X(z) = \frac{z}{z-a}$ with ROC $|z| < |a|$,
  - $X(z) = \frac{z}{z-a}$ with ROC $|z| > |a|$.
- remarkable examples,
  - Fibonacci sequence,
  - Poisson process and distribution;

(03/10/2022) 5. **Nyquist–Shannon sampling theorem.**

- Sampling:
  - generation of a sequence $x[n]$ starting from a continuous signal $x_a(t)$: $x[n] = x_a(nT)$;
  - basic issue: how good a continuous signal can be reconstructed from a sequence.
- Important definitions:
  - sampling time/period T;
  - sampling frequency/rate $f_s = \frac{1}{T}$, $\omega_s = \frac{2\pi}{T}$;
  - Nyquist frequency $f_{\rm Ny} = \frac{1}{2T} = \frac{f_s}{2}$, $\omega_{\rm Ny} = \frac{\pi}{T}$;
  - Nyquist band $\left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$.
- Nyquist–Shannon sampling theorem:
  - starting conditions:
    1a) an analog, continuous signal $x_a(t)$ is L^2 and
    1b) its sampled sequence $x[n]$ BIBO stable;
  - *"Nyquist–Shannon precursor relation"* between the $\mathcal{F}$–transform $\tilde{X}(\omega)$ of $x_a(t)$ and the z–transform $X(z)$ of $x[n]$:
    $X\left(e^{-i\omega T}\right) = \frac{1}{T}\sum_{\forall k} \tilde{X}(\omega + \frac{2\pi}{T}k)$,
    which is valid for $\omega \in \left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$ and thus, because of periodicity, $\forall \omega$.
  - final condition (the crucial one!):
    2) the analog, continuous signal $x_a(t)$ is $\frac{\pi}{T}$–BL (band–limited);
  - relation (*"Nyquist–Shannon sampling theorem" in the frequency domain*) between the $\mathcal{F}$–transform $\tilde{X}(\omega)$ of $x_a(t)$ and the z–transform $X(z)$ of $x[n]$:
    if $\omega \in \left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$ then $\tilde{X}(\omega) = \sum_{\forall k} \tilde{X}(\omega + \frac{2\pi}{T}k)$, and thus, because of the precursor relation,
    $\tilde{X}(\omega) = TX\left(e^{-i\omega T}\right)$ if $\omega \in \left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$;
  - *"Nyquist–Shannon sampling theorem" in the time domain*:

$x_a(t) = \sum_{\forall k} x[k] \, \mathrm{sinc}\left[(t - kT)\frac{\pi}{T}\right]$.

- Summary:
  1. additional, useful definitions,
     - let $\tilde{X}_{folding}(\omega)$ be defined as $\tilde{X}_{folding}(\omega) \equiv \sum_{\forall k} \tilde{X}(\omega + \frac{2\pi}{T}k)$, and
     - let the *reconstruction* $x_{rec}(t)$ be defined as $x_{rec}(t) \equiv \sum_{\forall k} x[k] \, \mathrm{sinc}\left[(t - kT)\frac{\pi}{T}\right]$;
  2. under the conditions 1a) $x_a(t) \in L^2$ and 1b) $x[n]$ BIBO stable, one has the *"Nyquist–Shannon precursor relation"*: $TX\left(e^{-i\omega T}\right) = \tilde{X}_{folding}(\omega)$;
  3. under the additional condition 2) $x_a(t)$ is $\frac{\pi}{T}$–BL, one has...
     - the *"Nyquist–Shannon sampling theorem" in the frequency domain*: $\tilde{X}(\omega) = TX\left(e^{-i\omega T}\right)$ if $\omega \in \left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$,
     - *"Nyquist–Shannon sampling theorem" in the time domain*: $x_{rec}(t) = x_a(t)$.
- Aliasing in the case of a non–$\frac{\pi}{T}$–BL signal $x_a(t)$:
  - additional definitions,
    * $W(\omega) \equiv [\theta(\omega + \frac{\pi}{T}) - \theta(\omega - \frac{\pi}{T})]$ ($\frac{\pi}{T}$–window),
    * $\tilde{X}_{alias}(\omega) \equiv \tilde{X}_{folding}(\omega) \cdot W(\omega)$;

  - inverse Fourier transform of $\tilde{X}_{alias}(\omega)$,

    a: $\tilde{X}_{alias}(\omega)$ is $\frac{\pi}{T}$–BL,

    b: $\sum_{\forall k} \tilde{X}_{alias}(\omega + \frac{2\pi}{T}k) = \tilde{X}_{folding}(\omega)$ (proof via definition of $\tilde{X}_{alias}(\omega)$ and periodicity of $\tilde{X}_{folding}(\omega)$),

    a, b $\Rightarrow$ c: if $\omega \in \left(-\frac{\pi}{T}, \frac{\pi}{T}\right)$, then $\tilde{X}_{alias}(\omega) = \tilde{X}_{folding}(\omega) = TX\left(e^{-i\omega T}\right)$,

    c $\Rightarrow$ d: $\mathcal{F}^{-1}\left[\tilde{X}_{alias}(\omega)\right] = x_{rec}(t)$;

  - aliasing: if $x_a(t)$ is not $\frac{\pi}{T}$–BL, then $\tilde{X}_{alias}(\omega) = \tilde{X}_{folding}(\omega) \cdot W(\omega) \neq \tilde{X}(\omega)$, and, by evaluating the inverse Fourier transform, $x_{rec}(t) \neq x_a(t)$, i.e. $x_{rec}(t)$ is not equal to $x_a(t)$ but it is *something else* ($\sim$ latin *alias*);

  - nevertheless: $x_{rec}[n] = x_a(nT) = x[n]$.
- Practical statement of Nyquist–Shannon sampling theorem: if f_0 is the maximum frequency occurring in a signal, use a sampling frequency f_s

such that $f_s > 2f_0$.

- Remarkable example: aliasing in the case of sinusoidal and cosinusoidal signals.

(10/10/2022) 6. **Simulation of an analog system by means of a digital one.**

- Theorem concerning the simulation of an analog system by means of a digital one ("simulation theorem"):
  - general discussion;
  - proof assuming...
    - 1a) Fourier-transformable input signals and transfer function, i.e. $x(t)$, $G(t) \in L^2$,
    - 1b) BIBO-stable simulator $h[n]$,
    - 2) $\frac{\pi}{T}$–BL input signals;
  - expression of the ideal transfer function $h[n]$ of the simulator.
- Implementation issues:
  - reality: $G(t)$ is real $\Rightarrow h[n]$ is real;
  - stability of the ideal transfer function $h[n]$, to be assessed case by case, and ...
  - non–causality issue for the ideal transfer function $h[n]$ (example: first-order low-pass filter);
  - difficulty – in the general case – of calculating $h[n]$ (example: first-order low-pass filter);

  - a remarkable example: digital differentiator (ideal solution $\propto (1 - \delta[n])\frac{(-1)^n}{n}$).
- "Backward interpretation of the simulation theorem":
  - statement: any digital system characterized by an impulse sequence $v[n]$—and the related z-transform $V(z)$—is a perfect simulator of an analog system whose impulse response's Fourier transform $\tilde{V}(\omega)$ is given by $V(e^{-i\omega T})$.
- Approximated simulation via "backward interpretation of the simulation theorem":
  - an approach to overcome the implementation issues: requiring the simulator's output $y'[n]$ to approximate the sampled analog output $y[n]$, $y'[n] \cong y[n]$, rather than imposing $h[n] = g[n]$.
    So, because it is mostly impossible to find $H(z)$, and thus $h[n]$, such that it exactly simulates a given $\tilde{G}(\omega)$,

find an implementable $V(z)$, and thus $v[n]$, such that its Fourier–transform equivalent function $V(e^{-i\omega T})$ suitably approximates $\tilde{G}(\omega)$;

– (approximation of an ideal system $h[n]$, $H(z)$ through a real one $v[n]$, $V(z)$ via, ex. gr., minimization of the Tchebycheff error or the root–mean–square error;)

– a remarkable example: digital differentiator,
  * solution $\delta[n] - \delta[n-1]$,
  * solution via a non recursive (FIR) filter based on delays up to 2 periods;

– a remarkable example: low pass filter,
  * solution via a recursive (IIR) filter $y[n] = a \cdot y[n-1] + b \cdot x[n]$.

(17/10/2022) 7. **Bilinear transform.**

- Bilinear transform:
  – an issue: how to express ω, or s, in terms of z by relying on $z = e^{-i\omega T}$ and exploiting the approximation approach to simulation provided by the backward interpretation of the simulation theorem;
  – bilinear transform statement $\omega \to \frac{2i}{T}\frac{z-1}{z+1}$, $s \to \frac{2}{T}\frac{z-1}{z+1}$
  $\Longrightarrow V(z) = \tilde{H}\left(\omega = \frac{2i}{T}\frac{z-1}{z+1}\right)$; $V(z) = \tilde{H}_{\text{Laplace}}\left(s = \frac{2}{T}\frac{z-1}{z+1}\right)$;
  – desirable frequency behaviour for $T \ll bandOfInterest^{-1}$.
- Simulation via bilinear transform of an LTI system characterized by a rational transfer function:
  – basic properties,
    * reality $\Big(h(t)$ is real $\Longleftrightarrow \tilde{H}^*(\omega) = \tilde{H}(-\omega)\Big)$,
    * causality (to be imposed),
    * BIBO–stability, $h(t)$ is BIBO–stable $\Leftrightarrow$ stable system with rational transfer function $\tilde{H}(s)$,
      · $s \to \frac{2}{T}\frac{z-1}{z+1} \quad \Longleftrightarrow \quad z \to \frac{sT/2+1}{sT/2-1}$,
      · $\Longrightarrow \quad \mathrm{Re}\, s < 0 \leftrightarrow |z| < 1$;
  – in the case of systems characterized by rational transfer functions $\tilde{H}(\omega) = N(\omega)/D(\omega)$, with degree$[N(\omega)] <$ degree$[D(\omega)]$,
  superiority of bilinear transform with respect to other transforms (for example the one in which $\tan(\omega T/2)$ is replaced by $\sin(\omega T)$),
  due to a desirable behaviour in the neighbourhood of the Nyquist frequency.
- Example (short overview):
  – low-pass filter simulator via bilinear transform;

- implementation via a difference equation;
- frequency response.

(24/10/2022) 8 **Filter design: examples.**

- Example: design of a first-order low-pass filter via bilinear transform:
  - derivation of $V(z)$ via bilinear transform;
  - basic properties,
    - reality,
    - causality (to be imposed),
    - BIBO–stability;
  - desirable behaviour in the neighbourhood of the Nyquist frequency;
  - implementation via a difference equation;
  - transfer function $\tilde{V}(\omega)$ via "backward interpretation of the simulation theorem".
- Structure of a filter implementing a rational function in the z–domain:
  - effect of zeroes and poles on $\tilde{V}(\omega)$;
  - example: design of a notch filter at $f = f_{Nyquist}/2 = f_{sampling}/4$:
    - positioning of zeroes and poles, by taking into account reality, causality, BIBO–stability,
    - final expression for $V(z)$,
    - frequency response $\tilde{V}(\omega)$,
    - inversion of z–transform $V(z)$ and practical implementation (left as a homework).

(07/11/2022) 9. **Discrete–Fourier–Transform (DFT).**

- Discrete–Fourier–Transform (DFT):
  - assumption of the periodicity of $x_a(t)$, with period NT, $N \in \mathbb{N}^+$ and calculation of the coefficients of the Fourier series;
  - assumption of the $\frac{\pi}{T}$–band–limitedness of $x_a(t)$;
  - (assumption of N being a power of 2);
  - derivation of DFT and relation with the Fourier series coefficients;
  - inversion, periodicity, Parseval's theorem.
- Matrix representation of DFT:
  - notation,
    - $f_n \equiv f[n]$,

    - ∗ *twiddle factor* $W_N \equiv e^{2\pi i/N}$;
  - matrix representation;
  - dependence $O(N^2)$ of the number of operations required for the DFT on the dimension N of the input string.
- Example: DFT of $\cos(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$ in the case $N = 8$.

(14/11/2022) 10. **Fast–Fourier–Transform (FFT).**

- A summary of last lecture.

- *Radix–2 decimation–in–time* Fast–Fourier–Transform (FFT) algorithm:
  - short history: Gauss 1805, Cooley and Tukey 1965;
  - Danielson–Lanczos lemma (*decimation*, i.e. split);
  - *butterfly* diagram (basic computational unit);
  - element ordering via *bit swapping*.
- Dependence $O(N \log N)$ of the number of operations required for the FFT on the dimension N of the input string.
- A short mention on DFT (and FFT) of a nonperiodic function and windowing.

- Examples of FFT in the case $N = 8$:
  - FFT of $\cos(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$;
  - FFT of $\delta[n-3]$;

  - ... proposed as a homework,
    - ∗ generic sequence $\{f_0, f_1, \ldots, f_7\}$, and equivalence with the DFT,
    - ∗ FFT of $\sin(2\pi f t)$ with $f = 2 \cdot \frac{1}{NT}$,
    - ∗ FFT of a constant,
    - ∗ FFT of $\delta[n]$,
    - ∗ FFT of $\delta[n-4]$.

(21/11/2022) 11. **An introduction to information theory: Shannon entropy of an ensamble.**

- The basic issue: a definition of information, i.e., how it can be measured:
  - Shannon's approach (1948) with uncertainty;
  - example of a sport match.

- Ensambles (Khinchin's finite schemes; "Experiment") as triples made of. . . :
  - random variable;
  - alphabet of symbols (or vocabulary of words), each corresponding to a realization of the r.v.;
  - probability distribution of the r.v.
- Shannon entropy of an ensamble:
  - Shannon-Khinchin axioms;
  - proof of uniqueness theorem;
  - examples,
    * Bernoulli trial,
    * Battleship (game).

(25/11/2022) 12. **An introduction to information theory: Shannon's source coding theorem for symbol codes; compression.**
(3 hours)

- A summary of last lecture.

- Kullback–Leibler divergence and Gibbs' inequality:
  - relative entropy, aka Kullback–Leibler divergence between two probability distributions;
  - Gibbs' inequality.

- Binary symbol codes and compression issue:
  - definitions related to symbol codes,
    * (binary) symbol code, codewords, length of codewords,
    * extended code,
    * uniquely decodable symbol code and prefix code,
      a prefix code is uniquely decodable; the contrary is not true (counterexample $\{1, 101\}$),
      a prefix code is (generally) easy to decode,
    * examples;
  - expected length $L(C, X)$ of a symbol code C that encodes an ensamble X;
  - compression issue: given an ensamble X, generation of a symbol code C that. . .
    * (compulsory) is uniquely decodable,
    * (desirable) is easy to decode (so, possibly, a prefix code),

    * (compulsory) minimizes the expected length $L(C, X)$;
  - Kraft inequality in the case of unique decodeability,
    * expression and proof,
    * definition of a complete symbol code,
    * given the size $|A_X|$ of an alphabet, existence of a complete, prefix code that encodes it (provable by construction).

- Shannon's source coding theorem for symbol codes.

- Huffman *lossless* coding algorithm:
  - algorithm;
  - example;
  - properties,
    * Huffman algorithm generates prefix symbol codes (provable by construction),
    * Huffman is complete (provable by construction),
    * Huffman coding is optimal (statement only),
    * (a "con" indeed:) necessity of knowing $p(x)$ in advance;
  - compression improvement via "syllables".
- Lempel–Ziv *lossless* compression algorithm (LZ77):
  - algorithm;
  - example.
- A mention of lossy compression algorithms.

LABORATORY CLASSES

(13/09/2022) L1. **Introduction to Verilog programming on a FPGA device. Counters and frequency dividers.**

- Safety rules.
- A short overview on FPGA devices and on HDL programming languages.
- Introductory problem: design of a 1 Hz counter with 8–LED array display, relying on standard analog and digital circuitry and an 8 Hz clock source.

- Basic hardware circuits:
  - frequency divider;
  - counter.
- Verilog programming language:
  - module architecture;
  - template example: development of an 8–bit, 1 Hz counter with an 8–LED array display, working from 0 to 255;

  - basic modules implementing basic hardware circuits:
    * frequency divider;
    * counter.

- Problems:
  - development of an 8–bit, 1 Hz counter with an 8–LED array display, working from 0 to 9;
  - development of an 8–bit, 10 Hz counter with an 8–LED array display, working from 0 to 9.
  - development of a 10 Hz counter, working from 0 to 99, with a 2–digits BCD coding and a 2 x 4–LED array display;
- Additional problems:
  - development of a 10 Hz down counter.

(14/09/2022) L2. **Multiplexers and demultiplexers. Synchronous counters.**

- Solutions to the problems assigned in the previous lab class.

- Hardware and software architectures:
  - combinatorial and sequential circuits;
  - synchronous and asynchronous circuits;
  - example: asynchronous counter and synchronous version by means of a finite-state machine.
- Risetime issue when clocking a flip-flop.
- Basic hardware circuits:
  - multiplexer;
  - demultiplexer.
- Verilog programming language:
  - combinatorial (*assign*) and sequential (*always*) Verilog modules;
- Problems:
  - development of a stopwatch from 0 to 99.99 s (1/100 s resolution), with 2–digits BCD coding and a 2 x 4–LED array display;
  - development of a syncronous counter based on a *"master clock"*.

(15/09/2022) L3. **Toggle flip-flops. Monostable multivibrators.**

- Solutions to the problems assigned in the previous lab class.
- Verilog programming language:
  - basic modules implementing basic hardware circuits:
    - multiplexer (synchronous and asynchronous version);
    - synchronous counter with set to a preset value, and reset.
  - basic modules implementing basic hardware circuits:
    - synchronous toggle flip-flop [*];
    - synchronous monostable multivibrator [*].

  [*] to be developed within the problems.
- Problems:
  - implementation of a synchronous module **toggle flip-flop** and, through this, of a *toggle pushbutton* to switch on/off an LED;
  - implementation of a synchronous module **monostable multivibra-**

tor and, through this, of an improved *toggle pushbutton* to switch on/off an LED; observation of the bouncing effect in a pushbutton;

  - implementation, by means of a monostable multivibrator and an improved *toggle pushbutton*, of a timer to switch on an LED for a given time (programmable through the switches).

(30/09/2022) L4. **Implementation of a stopwatch with OLED display.**

- Finite-state machines.

- Basic hardware circuits:
  - data latches.
- Verilog programming language:
  - peripheral device drivers (ex.gr.: OLED driver with display of the lap mode).

- Problems:
  - implementation of a synchronous 5-state finite-state machine with two pulse control;
  - final implementation of a stopwatch with 1/100 s resolution, start/stop, lap/reset function, and OLED display.

(12/10/2022) L5. **Development of drivers for hardware devices.**

- Numerical representation of natural and integer numbers:
  - *2's complement* representation of integers;
  - inversion of an integer $(-1) \cdot n = (\sim n) + 1$;
  - sum, difference $a - b = a + (-1) \cdot b$, multiplication $a \cdot b = |a| \cdot |b| \cdot sign(a) \cdot sign(b)$;
  - multiplication times 2^k and division by 2^k by means of the *shift* operator;

  - using the *2's complement* representation with ADC/DACs.

- Verilog programming language:
  - handling signed numbers.

- Problems:
  - implementation of an RGB driver to control a variable-color LED;
  - implementation of an *RGB simplex*.

(26/10/2022) L6. **Nyquist–Shannon sampling theorem made real. Waveform generation.**

- Using the *2's complement* representation with ADC/DACs.

- Basic hardware circuits:
  - shift register.
- Verilog programming language:
  - driver of the ADCs placed on the development board;
  - driver of the DACs placed on the development board;
  - shift register.

- Problems:
  - evaluation of the Nyquist frequency and the transfer function gain (voltage-to-number-to-voltage) of a ADC-DAC feedthrough system;
  - implementation of a delayer;
  - implementation of sawtooth waveform generators.

(09/11/2022) L7. **Implementation of a harmonic oscillator.**

- Theoretical and experimental aspects linked to the development of a harmonic oscillator:
  - general discussion on the difficulty of implementing an oscillator;
  - from the differential equation of a forced oscillator to the z–transform of the simulator response function $V(z)$;
  - derivation of the difference equation;
  - setting of the boundary conditions for the *cosine* operation;
  - dependency of the working frequency $f_0 = \omega_0/(2\pi)$ on the parameter k, provided that $\omega_0 T \ll 1$:
    $f_0 = f_s/(\pi 2^{\frac{k}{2}+1})$, with $f_s = 1/T$.
- Practical demonstration of FFT windowing.

- Problems:

– implementation and characterization of a harmonic oscillator.

(23/11/2022) L8. **Digital simulation of analog filters.**

- A short mention to the solutions to the problems assigned in the lab classes L4, L5, L6, L7.
- A summary on Bode diagrams (and on frequency roll-off in filters).
- Theory problem: design of a first-order low-pass filter via bilinear transform:
  - from the Fourier transform of the real system's response function to the z–transform of the simulator's response function $V(z)$;
  - difference equation and block diagram of the simulator;
  - implementation by using a pole placed at $1 - 2^{-k}$;
  - dependency of the cutoff frequency $f_{3\,\mathrm{dB}} = (2\pi\tau)^{-1}$ on parameter k, provided that $\omega_0 T \ll 1$;
  - frequency behaviour via "backward interpretation of the simulation theorem".
- Problems:
  - implementation of a first-order low-pass filter;
  - assessment of the transfer function (to be displayed via Bode–diagrams).