Digital Signal Processing

Jiří Málek

Part I

Formal requirements

Subject content and dependencies I.

- **Topics:**
  - Discrete (deterministic) signals and systems/filters
  - Spectrum of signal, short-time Fourier transform
  - Block-wise signal processing
  - Digital filters - frequency selective and optimal
  - Selected applications - spectral masking, event detection
- **Aim of the subject:**
  - Extends (or establishes) knowledge about classical signal processing methods
  - Introduces selected advanced topics
  - Explains context in signal/filter properties, which simplifies understanding of advanced concepts
  - Demonstrates the discussed principles on real-world signals (audio, ECG)

Subject content and dependencies II.

- **Previous subject:** SGI (Signals and information, in Czech)
  - Passing is not required, but it is an advantage
  - PZS revisits part of the topics contained in SGI, extends mathematical tools and provides more detailed explanations of the signal/filter properties
- **Subsequent subjects:**
- **DAI** (Digital audio engineering, in Czech)
  - Digital audio effects
  - Speech enhancement
  - Reconstruction of distorted signals (de-clipping, dereverberation)
  - Audio compression and psychoacoustics
- **MMZ** (Modern methods for signal processing, in Czech)
  - Biomedical signals and their processing
  - Optimal filtration
  - Multi-channel signal processing (beamforming, blind source separation)
  - Compressed sensing (sampling)

Requirements

- Knowledge of computing environment MATLAB
- Elementary knowledge of following mathematical topics:
  - Set of linear equations
  - Quadratic equation
  - Complex numbers
  - Complex functions (with real or complex domain)

Study materials

Recommended litarature:

- Detailed textbook:
  BOAZ PORAT, A Course in Digital Signal Processing. Wiley, 1997. ISBN 10: 0471149616.
- Overview textbook with solved examples:
  MONSON H. HAYES, Schaum's outlines of Theory and Problems of Digital Signal Processing, McGraw-Hill, 1999, ISBN 0-07-027389-8

Lecture notes and current information:

- Location will be specified by lecturer during first lecture

Seminars

Seminars: each lesson contains theory followed by practicals; seminars and the exam are connect via a scoring system

- Possibility to obtain at least +10 points (more for active students)
- **Absences:** Participation on most seminars is mandatory!!
  - Automatically excused absences: 2
  - Additional absences *may be* excused for *verified* serious reasons under condition that the student quickly learns the missing topics
  - **Unexcused absences** lead to:
    - **No credit for seminars:** if 3.5 points of seminars or less
    - **Mandatory additional oral exam:** if 7.5 points of seminars or less
- **Participation:** *activity* and *good manners* are required (penalization as for an unexcused absence or as by disciplinary regulations)
- Students with acute respiratory disease should stay at home!!
- **Points can be gathered for:**
  - Discussion at the beginning of the seminar (max 2×1.5 points)
  - 2 short tests during semester (0-1.5 points)
  - 3-4 voluntary homeworks - tasks in Matlab (+1/+1.5 point according to the difficulty)
  - Activity during the seminars (at the discretion of the lecturer)

Exam

Written exam with possible additional oral examination:

- **Scoring system:** 30 points computational exercises, 10 points theory, 10 points from the seminars
- **Successful completion:**
    - 28-50 points
      (no unexcused absences or more than 7.5 points of seminars)
    - (Voluntary) **grade improvement** - e.g., due to scholarship
    - Required: at least 35 points from exam and more than +5 points from seminars
    - Form: additional oral examination
    - Success: grade improved by up to two degrees
    - Failure: grade worsened by up to one degree
- **Additional oral examination**:
    - **Borderline exam point count** (24-27.5)
    - Success: grade E
    - Failure: grade F, repetition of the whole exam
    - **Unexcused absences** (and 7.5 points of seminars or less)
    - Success: grade according to the exam points
    - Failure: grade F, repetition of the oral exam

What will you learn...

By completion of the subject you gain:

- Solid basics from the vast topic of signal processing
- The ability to solve basic DSP problems independently(and usually very fast)
- Knowledge of key terms, which are essential for *searching* of additional information and the *understanding* of these terms
- The ability to continue in gaining advanced DSP knowledge through TUL subjects, literature and Matlab help
- The training to formulate precisely a technical information (during discussion during laboratories)
- The ability to actively use standard mathematical notation (examples)

Part II

Introduction into digital signal processing

- **Signal** is a mathematical function, which represents information about evolution of some physical quantity
- Most of the real-world signals are **analog**
    - Electrical signals (voltage, currents ...)
    - Mechanical Signals (shifts, angles, speeds, accelerations)
    - Acoustic signals (vibrations, sound, speech)
    - Many others (pressure, temperature, concentration, ...)
- Usually, non-electrical quantities are transformed into electrical, to enable practical processing

Analog Signal Processing - ASP:

- Operations: Amplification, filtration, integration, nonlinear operations..
- Processing tools: electrical circuits using amplifiers, resistors, capacitors...

Analog signals II.

Limitations of ASP:

- Limited accuracy (component tolerances, component nonlinearity ...)
- Limited reproducibility (environment conditions - temperature, vibrations)
- Vulnerability to electrical hum/noise
- Limited processing speed due to physical delays
- Limited flexibility (filter coefficients cannot be changed, once the circuit is designed)
- Difficult implementation of time-variant systems
- Difficult and costly data saving

ASP is used for:

- simple tasks, where is it more economical than DSP
- tasks, where sampling frequency is prohibitively high
- tasks, where analog signals are inputs into digital processors

Digital signal processing I.

- Continuous functions are replaced by number sequences
- Digital operations: Summing, multiplication, logical operations,
- Realization of digital systems requires the following steps:
  - Transformation of analog signal into digital (sampling, A/D conversion)
  - Application of DSP algorithm
  - Inverse transformation to analog signal (D/A conversion, reconstruction)

DSP limitations:

- Sampling may lead to information loss
- The cost of complexity of A/D and D/A convertors is high for ultra high speeds
- Distortions are introduced due to algorithmic errors (rounding errors, sample skipping)
- The speed for some applications is beyond technical capability of today's technologies (radio waves up to 300 GHz).

Digital signal processing II.

Utilization of DSP:

- *Biomedical applications* (diagnostics, patient monitoring, prevention)
- *Communication* (encoding, decoding, encrypting, filtration)
- *Control systems* (servomechanisms, autopilots)
- *Signal analysis* (signal modeling, classification, compression)
- *Image processing* (image modifications, computer vision)
- *Multimedia* (movies, digital television, video-conferences)
- *Musical and sound applications* (recording, reproduction, special effects)
- *Speech applications* (denoising, compression, recognition, synthesis)

Digital signal processing III.

- **Goal:** extraction of a relevant information from a signal
- **Techniques:** mathematics, algorithms
- **Complications:** Measurements of a desired signal usually contain also undesired phenomena
  - EXAMPLE: Speech signal recorded on a microphone consists of desired speech component and undesired environmental noise component
  - PHENOMENA: Noise, sensor and measuring principle imperfections, measurement errors, time-variant properties of the environment/signals
  - In the first part of DSP, we will omit these complications (for simplicity sake)
- Tasks can be divided into two (overlapping) groups
  - Signal analysis - learning of important signal properties
  - Signal modification (processing, e.g., via filtration) - compensation of measurement imperfections, amplification of desired signal components

Digital signal processing IV.

- What typical tasks are solved via signal processing?

Signal analysis:

- Activity detection (presence of speech in noisy signal)
- Event detection (seeking of QRS complex in ECG - heart beat detection, seeking of drum beats in music - tempo measurement)
- Detection of specific quantities (seeking of pitch in speech/music - height of the voice)
- Prediction - estimation of future values based on the previous ones (prediction of stock/commodity prices)

Signal modification:

- Removal of undesired signal components (removal of noise from speech recording, removal of artifacts from ECG)
- Amplification of desired signal components (seeking of trend in a sequence of newly infected people)
- Modulation - Change of a signal for effective transmission purposes
- Compression - Change of a signal for effective storing purposes

Classical versus data-driven signal processing

Conventional signal processing: (model-based)

- Original approach, suitable for tasks with available mathematical description / physical model / scenarios with simplifying conditions
- Example: Removal of power hum, voice activity detection in undistorted recording, detection of QRS complex in ECG, resampling etc.
- Requires rather detailed knowledge and/or statistical assumptions about the solved task
- Specific models, small number of free parameters (units to thousands)
- Usually less computationally demanding
- Discussed in the PZS subject

Machine learning for signal processing: (data-driven)

- State-of-the-art approach suitable for tasks without known mathematical models and general scenarios
- Example: Removal of an unwanted voice from a mixture of utterances, voice activity detection in distorted recordings, detection of epileptic seizures in EEG data
- Based on large databases (Big Data) of signals with reference solutions (often supervised learning)
- General models with many free parameters (millions to tens of millions)
- Training is very computationally demanding (GPU acceleration), demands during test phase depend on the size of the model

Both principles supplement each other: both have specific advantages and use cases.

Part III

Basics

Basic notations

- *Real numbers:* $\mathcal{R}$
- *Complex numbers:* $\mathcal{C}$
- *Integers:* $\mathcal{Z}$
- *Analog signals:* $x(t)$, (continuous functions on $\mathcal{R}$ or $\mathcal{C}$)

Discrete signals and systems:

- **Discrete signals:**
- Indexed infinite sequence of numbers from $\mathcal{R}$ or $\mathcal{C}$
- Notation: $x[n]$
- **Discrete systems:**
- Mathematical operator transforming input discrete signal $x[n]$ into output discrete signal $y[n]$
- Notation: $y[n] = T(x[n])$

Models of signals

- **Deterministic signals**
  - Description via a mathematical formula (or sequence of samples) in any instant, e.g., $x[n] = \sin(\omega n)$
  - Suitable for signals generated by human activity (rotating machinery, musical instruments etc.)
  - Assumed to be known at any time
  - Suitable for study of deterministic systems / filters (exact response - identification of systems)
  - Simpler analysis
  - Focus of the PZS subject
- **Stochastic signals**
  - Cannot be described by mathematical formula, uses statistical description, e.g., $x[n] = N(0, 1)$ (characterization through typical parameters, e.g., expected value and variance)
  - Description of biological / physical / social phenomena with multiple unknown factors
  - Future values are assumed unknown and need to be estimated from known past values
  - More complex analysis and mathematical apparatus

Part IV

Discrete deterministic signals

Discrete signal

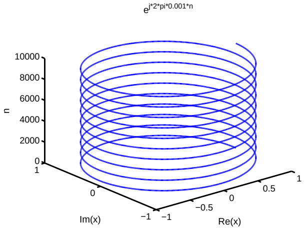
- Signal $x[n]$ - indexed infinite sequence of numbers from $\mathcal{R}$ or $\mathcal{C}$
- Now assumed: discrete signals are not quantized
- **Discrete signal origins:**
  1. Sampling of continuous signal $x(t)$ at equidistant instants with distance T_s:
     - If **sampling frequency** $F_s = 1/T_s$, then:
     - $x[n] = x(nT_s)$
  2. Naturally discreet sequence (daily development of stock prices, exchange rates, etc.)
- **Signal periodicity:**
- $x[n] = x[n+N], \forall n, \; N \in \mathcal{Z}, \; N > 0$
- Signal values $x[n]$ are repeated with *period* N

Basic discrete signals I.

Complex sequences are often decomposed into a sum of simpler functions.

- **Unit sample / impulse**
- $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$
- **Unit step**
- $u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$
- *Unit sample* and *Unit step* connects the following relation:
- $u[n] = \sum_{k=-\infty}^{n} \delta[k]$
- $\delta[n] = u[n] - u[n-1]$

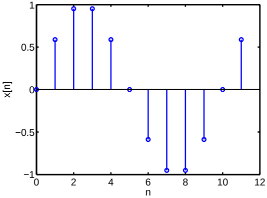
- **Exponential sequence**
- $x[n] = a^n$
  $a \in \mathcal{R}$ or $a \in \mathcal{C}$
- Special case - *complex exponential*
  - Important for digital signal processing (DSP), utilized in Fourier decomposition of signals
  - $x[n] = e^{j\omega_0 n}$
    $\omega_0 \in \mathcal{R}$
  - *Euler formula:*
    $e^{j\omega_0 n} = \cos[\omega_0 n] + j\sin[\omega_0 n]$

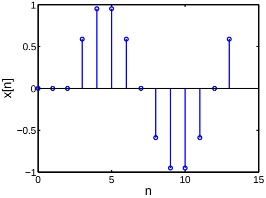


Complex signal manipulations can often be decomposed into several basic ones

Transformation of the independent variable n

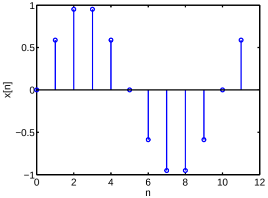
- $y[n] = x[f(n)]$
  $f(n)$ is an arbitrary function of n
  If $f(n)$ returns non-integer number, then $y[n] = x[f(n)]$ is undefined
- **Shifting**
  - $f(n) = n - n_0$
  - Positive n_0 means *a delay* - Shift to the right
  - Negative n_0 means *an advance* - Shift to the left

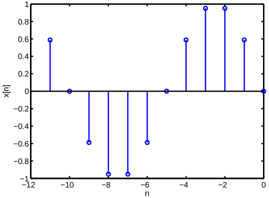




Basic signal manipulations II.

- **Reversal**
  - $f(n) = -n$
  - Reversal around the amplitude axis

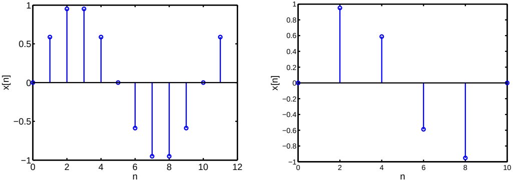




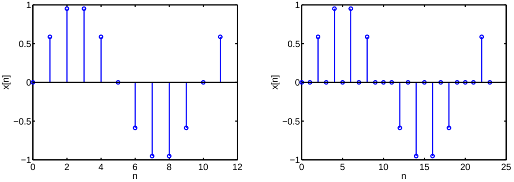
- **Time scaling**
  - **Down-sampling**
  - $f(n) = Mn, \; M \in \mathcal{Z},$
  - New signal contains every Mth sample of the original
  - **Up-sampling**
  - $f(n) = n/N, \; N \in \mathcal{Z},$
  - $N-1$ zeros are inserted between two samples of the original
- BEWARE - These operations are order dependent

Basic signal manipulations III.

- Down-sampling:



- Up-sampling:



- EXAMPLE: Signal is given by $x[n] = (6 - n)(u[n] - u[n-6])$. Draw a graph of signal $y[n] = x[2n-3]$.
- EXAMPLE: Decomposition of an arbitrary $x[n]$ into a sum of $\delta[n]$.

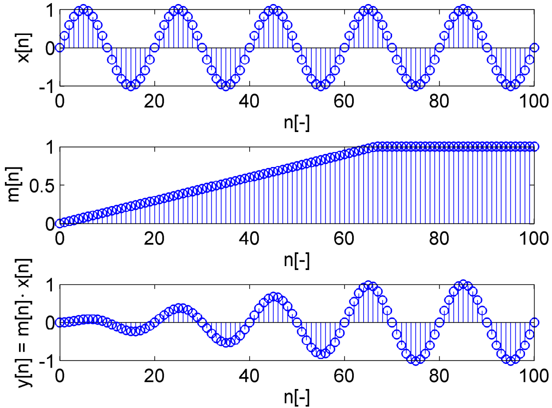
Basic signal manipulations IV.

Transformation of a dependent variable $x[n]$ - Change of amplitude

- **Addition**
  - $y[n] = x_1[n] + x_2[n], \ -\infty < n < \infty$
  - Sample-wise summation of signals
- **Multiplication**
  - $y[n] = x_1[n] \cdot x_2[n], \ -\infty < n < \infty$
  - Sample-wise multiplication of signals
- **Scaling**
  - $y[n] = c \cdot x[n], \ -\infty < n < \infty$
  - Amplitude $x[n]$ is amplified c-times.

Transformation of a dependent variable $x[n]$ - Change of amplitude

- EXAMPLE: Audio-effects fade-in / fade-out



Fade-in effect

Part V

Discrete systems

Discrete-time system I.

- Mathematical operator transforming an input discrete signal into another output discrete signal, denoted as $T(\cdot)$
- $y[n] = T(x[n])$,
  $y[n]$ - response of the system $T(\cdot)$ to an input signal $x[n]$
- **Difference Equation:**
- A relation (recursively) defining the output of the system as (in general time-variant) combination of values of the input and output signal.
- EXAMPLE: $y[n] = x[n]^2$ or $y[n] = 0.5 \cdot n \cdot y[n-1] + x[n]$

System properties:

- **Memoryless**
- Output at time $n = n_0$ depends only on the input at time $n = n_0$.

- **Causality**
- System is causal when for each $n_0 \in \mathcal{Z}$ the response at time n_0 depends only on input values corresponding to $n \leq n_0$.
- LTI system is causal when $h[n] = 0, \ \forall n < 0$.
- EXAMPLE: Decide about causality of the following systems:
  $y_1[n] = x[n] + x[n-1]$
  $y_2[n] = x[n] + x[n+1]$

Discrete-time system II.

- **Stability**
- BIBO stability (Bounded Input - Bounded Output)
- System is stable if for $|x[n]| < A < \infty$ holds that $|y[n]| < B < \infty, \; A, B \in \mathcal{R}$
- Concerning LTI systems, this conditions is equal to $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$
- EXAMPLE: Decide about stability of the systems:
  $h_1[n] = a^n u[n]$
  $y_2[n] = nx[n]$

- **Shift invariance**
- Let $y[n]$ be the response of $T(\cdot)$ on input $x[n]$
- Then $T(\cdot)$ is shift invariant, if for arbitrary delay n_0 holds that the response to $x[n-n_0]$ is $y[n-n_0]$.

- **Additivity**
- $T(x_1[n] + x_2[n]) = T(x_1[n]) + T(x_2[n])$

- **Homogenity**
- $T(cx[n]) = cT(x[n])$

Part VI

LTI systems

LTI system

- **Linearity**
- System is linear if it is *additive* and *homogeneous*
- $T(a_1x_1[n] + a_2x_2[n]) = a_1T(x_1[n]) + a_2T(x_2[n])$
- **Linear time-invariant system**
- LTI system is linear and shift invariant

Linear constant coefficient difference equation (LCCDE)

- Special case of difference equations describing LTI systems

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] - \sum_{k=1}^{p} a[k]y[n-k] \qquad (1)$$

$a[k], b[k]$ - Constants defining the system

- Relation defining the output of the system as linear combination of input and output values
- EXAMPLE:

$$y[n] = 3x[n] + x[n-1] - 5x[n-2] - 2y[n-1] + 0.5y[n-2],$$

$$y[-1] = 2, y[-2] = 4$$

- *Recursive/non-recursive* LCCDEs
- Recursive LCCDEs require *initial conditions*

Solving of difference equations:

- Formulation of the system output (for a specific input) using non-recursive function with independent variable n
  1. Numerical solution using recursive substitution (table of input and corresponding output values)
     MATLAB: `y = filter(b,a,x)`
     For the system in the example on the previous slide:
     `y = filter([3 1 -5],[1 2 -0.5],x);`
     BEWARE the sign of coefficients $a[k]$, $a[0]$ is always 1 (coefficient corresponding to $y[n]$)
  2. (Analytical solution in the time-domain (using *homogeneous* and *particular* solutions))
  3. Analytical solution using DTFT (when initial conditions are zero, in lecture 3)
  4. Analytical solution using Z-transform (in lecture 10)

EXAMPLE: Computation of output for system given by LCCDE; recursive substitution

- **Impulse response** $h[n]$
- $h[n]$ is a response of the LTI system to unit impulse $\delta[n]$
- **Computation of impulse response from LCCDE:**
- Solution of LCCDE for $x[n] = \delta[n]$ and zero initial conditions
- For non-recursive systems:

$$h[n] = \sum_{k=0}^{q} b[k]\delta[n-k] \tag{2}$$

  Finite impulse response - FIR system
  EXAMPLE: FIR system
- For recursive systems is the impulse response infinite
  Infinite impulse response - IIR system
  EXAMPLE: IIR system
- **Meaning of the impulse response:**
- Impulse response uniquely describes LTI system (as LCCDE does)
- EXAMPLE: Why is it so?

Convolution I.

- Expresses relation between input and output of the LTI system given by an impulse response

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

- **Convolution computation:**
  - **Direct evaluation by the definition sum:**
  - Advantageous for sequences given by explicit formulas

  - **Graphical approach:**
  - Plot the samples of sequences $x[n]$ a $h[-n]$ (reversed $h[n]$)
  - *Value* $y[0]$: Align below each other samples $x[0]$ and $h[0]$ and multiply them
  - *Value* $y[1]$: Shift $h[-n]$ by one sample to the right, multiply corresponding values ($x[0] \cdot h[1]$,$x[1] \cdot h[0]$) and sum them together

Convolution II.

Convolution computation:

- **Multiplication of polynomials:**
- Power coefficients correspond to shifted samples of sequences
- BEWARE - No signal is reversed

- **Composition of shifted impulse responses:**
- Convolution corresponds to the sum of responses to each (amplified and shifted) unit sample in the input signal

- **MATLAB:** `y = conv(h,x)`
- $\mathtt{Lh} = \mathtt{length(h)}$
- $\mathtt{Lx} = \mathtt{length(x)}$
- $\mathtt{Ly} = \mathtt{Lh} + \mathtt{Lx} - 1$
- EXAMPLE: Compute convolution of sequences $x[n], h[n]$:
  $h[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$
  $x[n] = u[n-1] - u[n-4]$
  `y = conv([1 2 3],[0 1 1 1]);`

Convolution III.

Convolution properties:

- **Commutative property:**
- $x[n] * h[n] = h[n] * x[n]$

- **Associative property:**
- $\{x[n] * h_1[n]\} * h_2[n] = x[n] * \{h_1[n] * h_2[n]\}$
- Serial interconnection of systems $h_1[n], h_2[n]$ can be replaced by a single system with impulse response $h_{eq} = h_1[n] * h_2[n]$

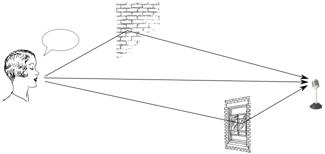
- **Distributive property:**
- $x[n] * \{h_1[n] + h_2[n]\} = x[n] * h_1[n] + x[n] * h_2[n]$
- Parallel interconnection of systems $h_1[n], h_2[n]$ can be replaced by a single system with impulse response $h_{eq} = h_1[n] + h_2[n]$

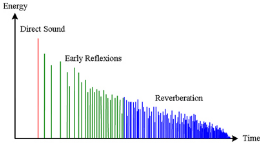
EXAMPLE: Modeling of acoustic environment through room impulse responses (RIRs)

- Let $s_m[n]$ be a version of an original $s[n]$ measured in an reverberant (echoic) environment on a microphone
- The relation between the original (anechoic) $s[n]$ and the reverberant $s_m[n]$ can be modeled through

$$s_m[n] = \sum_{\tau=0}^{M-1} h[\tau] \cdot s[n-\tau], \tag{3}$$

- $h[n]$ - **Room impulse response (RIR)** - Impulse response modeling sound propagation from the source to a sensor
- RIR arises through (partial) reflections of the sounds on walls/obstructions in the environment
- The length and the shape of the RIR greatly differs with respect to the corresponding enclosure (small room / concert hall)





ZDROJ: `http://www.acoustics.org`

Thank you for attention!