Digital Signal Processing

Jiří Málek

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).

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:
     - $x[n] = x(nT_s)$ ... n-násobek periody
  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.

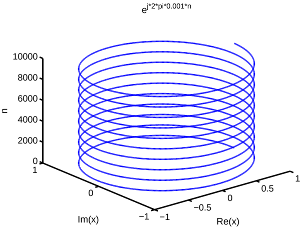
Complex sequences are often decomposed into a sum of simpler functions.

- **Unit sample / impulse**
- $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$ - všude krom nuly má hodnotu 1
- **Unit step**
- $u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$ - tam kde je 0 a vtší má hodnotu 1
- *Unit sample* and *Unit step* connects the following relation:
- $u[n] = \sum_{k=-\infty}^{n} \delta[k]$
- $\delta[n] = u[n] - u[n-1]$

- vztah jednotkového kroku a impulzu

Basic discrete signals II.

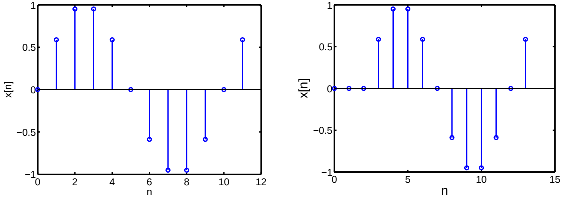
- **Exponential sequence**
- $x[n] = a^n$
  $a \in \mathcal{R}$ or $a \in \mathcal{C}$
- Special case - *complex exponential*
  - Important for digital signal processing (DSP), utilized in Fourier decomposition of signals
  - $x[n] = e^{j\omega_0 n}$
    $\omega_0 \in \mathcal{R}$
  - *Euler formula:*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



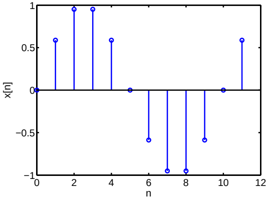
Complex signal manipulations can often be decomposed into several basic ones

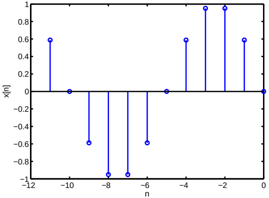
Transformation of the independent variable n

- $y[n] = x[f(n)]$
  $f(n)$ 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



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

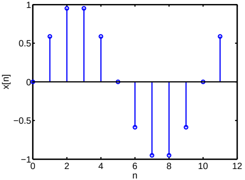


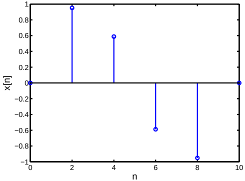


- **Time scaling**
  - **Down-sampling**
  - $f(n) = Mn, \; M \in \mathcal{Z},$
  - New signal contains every Mth sample of the original
  - **Up-sampling**
  - $f(n) = n/N, \; N \in \mathcal{Z},$
  - $N - 1$ zeros are inserted between two samples of the original
- BEWARE - These operations are order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

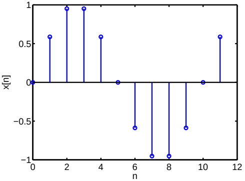
Basic signal manipulations III.

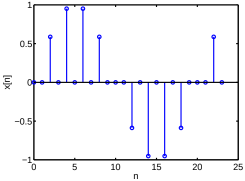
- Down-sampling:





- 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 $\delta[n]$.

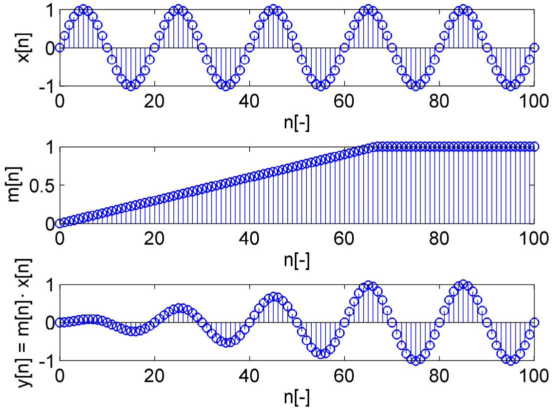
Basic signal manipulations IV.

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** - pokud výstup závisí pouze na aktuálním vstupu
- Output at time $n = n_0$ depends only on the input at time $n = n_0$.

- **Causality** - pokud systém závisí pouze na aktuální a predchozích hodnotách
- 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** - na konecný vstup odpoví konecným výstupem (ve smyslu aplitudy)
- 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:
  $h_1[n] = a^n u[n]$
  $y_2[n] = n x[n]$

- **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 $y[n - n_0]$. - filter se chová stejne pri casových posunech
- **Additivity** - nezalezi na poradi scitani a transformace
- $T(x_1[n] + x_2[n]) = T(x_1[n]) + T(x_2[n])$

- **Homogenity** - nezalezi jestli nejdriv transformujeme a pak nasobime, nebo na opak
- $T(cx[n]) = cT(x[n])$

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

- **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 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)`
- $\mathtt{Lh} = \mathtt{length}(\mathtt{h})$
- $\mathtt{Lx} = \mathtt{length}(\mathtt{x})$
- $\mathtt{Ly} = \mathtt{Lh} + \mathtt{Lx} - 1$
- 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 $h_{eq} = h_1[n] + h_2[n]$

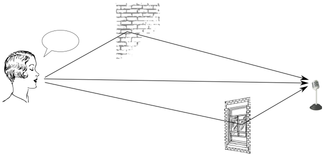
EXAMPLE: Modeling of acoustic environment through room impulse responses (RIRs)

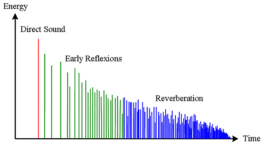
Room impulse responses

- 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`

Part I

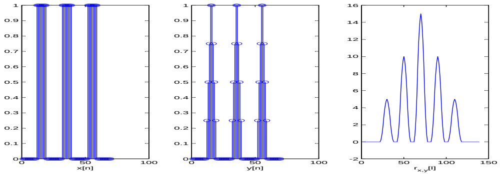
Correlation of deterministic signals, energy

Cross-correlation of deterministic signals

(podobne skalarnimu soucinu)

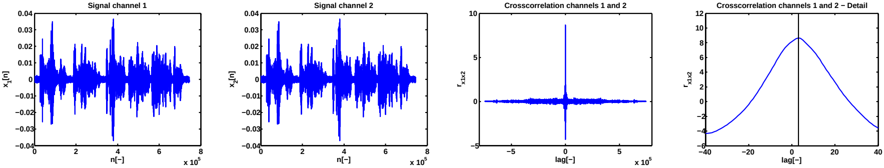
- Operator, which computes mutual similarity of two signals $x[n]$ and $y[n]$ with respect to their mutual shift ℓ (lag)
- For $x[n], y[n] \in \mathcal{R}$ is the cross-correlation $r_{xy}[\ell]$ defined by:

$$r_{xy}[\ell] = \sum_{n=-\infty}^{\infty} x[n+\ell]y[n] \qquad (1)$$



- Cross-correlation of real-valued signals is a real-valued function, migth be positive or negative and is not symetric in general
- **Time delay analysis:**
- EXAMPLE: Determination of time-delay and direction of arival of an acoustic signal using binaural microphone array

Time difference of Arrival: (TDOA)



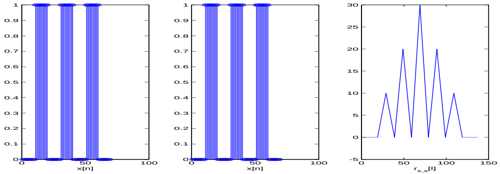
1. Value $\ell_{\max}$ is determined, for which $|r_{xy}[\ell]|$ yields the highest value
2. $\text{TDOA} = \ell_{\max} T_s$, ($T_s$ - sampling period)
3. Angle between the axis of the microphone array and the direction of sound arrival is given by

$$\text{DOA} = \arcsin\left(\frac{\text{TDOA} \cdot c}{d}\right) \tag{2}$$

c - speed of sound (343m/s in the air), d - microphone distance

- Operator, which quantifies the similarity of the signal to itself, with respect to a shift ℓ
- For $x[n] \in \mathcal{R}$ is the autocorrelation $r_{xx}[\ell]$ defined by:

$$r_{xx}[\ell] = \sum_{n=-\infty}^{\infty} x[n+\ell]x[n] \tag{3}$$



- Autocorrelation function is real, might be positive or negative and has always even symetry around $\ell = 0$.
- **Periodicity detection:**
- Autocorrelation is often used as a simple tool to determine a periodicity of a general noisy signal.

Correlation properties

- **Cross/Auto-correlation** can also be computed using convolution by

$$r_{x,y}[n] = x[n] * y[-n] \tag{4}$$

- (This feature is exploited, when the correlation is computed using the Fast Fourier Transform (FFT))
- Cross-correlation is **not commutative**

$$r_{x,y}[n] = r_{y,x}[-n] \tag{5}$$

- The interchange of variables results into time-reversed correlation function.

Signal energy:

- Energy is a value of auto-correlation $r_{xx}[0]$, i.e. a quantity given by

$$E_x = \sum_{n=-\infty}^{\infty} x[n]^2 \tag{6}$$

- Scalar variable measuring "activity"/"size" of the signal
- Finite-energy signals, $E_x < \infty$
- Energy of the harmonic periodic signal is infinite

Signal power:

- Average energy of a signal within the time-interval of length N

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} x[n]^2 \tag{7}$$

- The power of periodic signal is computed for a single period
- The power of harmonic signals is therefore finite
- Instantaneous power is computed for $N = 1$

Part II

Spectral analysis - introduction and motivation

Spectral analysis - motivation I

Discrete signal $x[n]$:

- Real/complex function defined on a set of integers
- In other words: Indexed infinite sequence of numbers from $\mathcal{R}$ or $\mathcal{C}$
- This type of signal description is often (and in a slightly misleading way) denoted as description **in time-domain**, because n has often the meaning of a time index
- For many applications, this description does not explicitly reflect the most important information stored in the signal
- EXAMPLE: In the speech signal, the value of $x[n]$ has the meaning of instantaneous loudness
  - However, the speech signal carries much more important information in a hidden form (what was said, who speaks, emotion of the speaker)
  - **Spectral analysis** attempts to reveal such information

Spectral analysis - motivation II

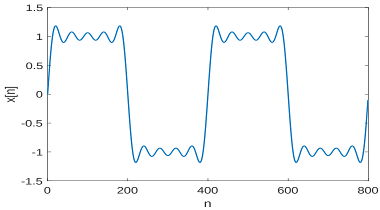
- EXAMPLE: In a musical audio signal, $x[n]$ has also the meaning of the instantaneous loudness
  - The information about tone pitch or used musical instrument is encoded as a speed of (periodical) changes (oscillations) of $x[n]$
  - Fast changes of $x[n]$ correspond to high-pitched tones, slow ones to low-pitched tones
- Similar to physics, oscillations are described using harmonic functions (cosine) and their **frequencies**
- For continuous signals, an analog frequency F[Hz] is used (as in physics, number of repetitions per unit of time), or its scaled version $\Omega = 2\pi F$
- For discrete signals, similar variable is defined - **digital frequency** ω[-]

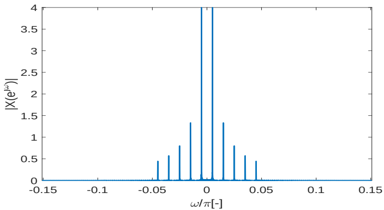
$$\omega = \frac{2\pi F}{F_s}, \tag{8}$$

F_s - sampling frequency

Spectral analysis - motivation III

- One of the results of the spectral analysis is the **magnitude spectrum**, which has several forms (will be discussed lated)
- Magnitude spectrum quantifies, how much is the signal similar to a harmonic function (cosine) on a given frequency
  - In other words: it quantifies an extend, to which are the individual harmonic functions on various frequencies present in the signal
- Spectrum is another view of a signal $x[n]$, it is a description in the **frequency domain**
- Spectrum carries the same information as the sequence $x[n]$, it just explicitly states another its part
- Basic tools of the spectral analysis are various forms of the Fourier transform (DTFT, DFT, STFT)





Part III

Discrete Time Fourier Transform (DTFT)

Discrete Time Fourier Transform (DTFT)

- Mapping from a set of sequences (signal, impulse response) into a set of continuous complex functions of real variable (*DTFT spectrum*, *frequency response*)
- Mapping from *time-domain* into *frequency-domain*, where many properties of signals and systems can be more easily studied
- It is given by:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega} \tag{9}$$

$\omega \in \mathcal{R}$ - digital frequency, $-\infty < n < \infty$

- $X(e^{j\omega})$ is continuous complex function of real variable ω.
- The value of $X(e^{j\omega})$ at ω_0 is *cross-correlation* of sequence $x[n]$ and complex exponential $e^{jn\omega_0}$ (for $\ell = 0$)
- Meaning of $X(e^{j\omega_0})$:
  - *Magnitude* $\left|X(e^{j\omega_0})\right|$ - the level of correlation between $x[n]$ and $e^{jn\omega_0}$
  - *Phase* $\phi(\omega_0)$ - shift between $x[n]$ and $e^{jn\omega_0}$

$$X(e^{j\omega_0}) = \left|X(e^{j\omega_0})\right| e^{j\phi(\omega_0)} \tag{10}$$

Discrete Time Fourier Transform II

- EXAMPLE: Computation of $X(e^{j\omega})$ using the DTFT definition formula
- **Inverse discrete time Fourier transform**:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{jn\omega} d\omega \qquad (11)$$

- List of commonly used DTFT pairs:

Sequence	DTFT image
$\delta[n]$	1
$\delta[n-n_0]$	$e^{-jn_0\omega}$
1	$2\pi\delta(\omega)$
$e^{jn\omega_0}$	$2\pi\delta(\omega-\omega_0)$
$\alpha^n u[n], \; \lvert\alpha\rvert < 1$	$\frac{1}{1-\alpha e^{-j\omega}}$
$-\alpha^n u[-n-1], \; \lvert\alpha\rvert > 1$	$\frac{1}{1-\alpha e^{-j\omega}}$
$(n+1)\alpha^n u[n], \; \lvert\alpha\rvert < 1$	$\frac{1}{(1-\alpha e^{-j\omega})^2}$
$\cos([n\omega_0])$	$\pi\delta(\omega-\omega_0)+\pi\delta(\omega+\omega_0)$

DTFT properties I

- **Periodicity**:
- DTFT is periodic with period 2π, i.e.,
- $X(e^{j\omega}) = X(e^{j(\omega+2\pi)})$
- QUESTION: Why is it so?

- **Symmetry:**

$x[n]$	$X(e^{j\omega})$
Real, even	Real, even
Real, odd	Imaginary, odd
Imaginary, even	Imaginary, even
Imaginary, odd	Real, odd

DTFT properties II

- **Linearity:**
- If $X_1(e^{j\omega})$ is DTFT of $x_1[n]$ and $X_2(e^{j\omega})$ is DTFT of $x_2[n]$, then:
- $ax_1[n] + bx_2[n] \overset{\text{DTFT}}{\Longleftrightarrow} aX_1(e^{j\omega}) + bX_2(e^{j\omega})$
- **Time-reversal:**
- Time-reversal $x[n]$ leads to reversal of $X(e^{j\omega})$ in the frequency domain, i.e.:
- $x[-n] \overset{\text{DTFT}}{\Longleftrightarrow} X(e^{-j\omega})$
- **Shifting:**
- Shift of the sequence $x[n]$ leads multiplication of $X(e^{j\omega})$ by a complex exponential, i.e.,:
- $x[n-n_0] \overset{\text{DTFT}}{\Longleftrightarrow} e^{-j\omega n_0} X(e^{j\omega})$

DTFT properties III

- **Modulation:**
- Multiplication of a sequence by a complex exponential leads to a shift in the frequency domain, i.e.,
- $e^{jn\omega_0} x[n] \overset{\text{DTFT}}{\Longleftrightarrow} X(e^{j\omega - \omega_0})$
- Multiplication of a sequence by a signal $\cos(\omega_0 n)$ leads to two shifted copies of $X(e^{j\omega})$ in the frequency domain, i.e.,
- $\cos(\omega_0 n) x[n] \overset{\text{DTFT}}{\Longleftrightarrow} \frac{1}{2} X(e^{j\omega - \omega_0}) + \frac{1}{2} X(e^{j\omega + \omega_0})$
- **Convolutional theorem**:
- Convolution of two signals in time-domain equals the multiplication of the DTFTs of these signals in the frequency-domain
- $h[n] * x[n] \overset{\text{DTFT}}{\Longleftrightarrow} H(e^{j\omega}) X(e^{j\omega})$

DTFT properties IV

- **Multiplication theorem:**
- Multiplication in the time-domain corresponds to a (periodic) convolution in the frequency-domain
- $x[n]y[n] \overset{\text{DTFT}}{\Longleftrightarrow} \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})d\theta$
- **Parseval theorem:**
- DTFT preserves the energy of a signal when transitioned from the time- into the frequency-domain
- $\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|X(e^{j\omega})\right|^2 d\omega$

Part IV

DTFT spectrum

DTFT spectrum $X(e^{j\omega})$

- Complex-value function of independent variable $\omega \in \mathcal{R}$, expresses correlation of the signal $x[n]$ and a complex exponential on a specific frequency ω_0.
- *Computation:* application of DTFT on the signal $x[n]$
- Meaning for *real-valued* $x[n]$: it is a decomposition of $x[n]$ into the sum (of an infinite number) of harmonic functions $\cos(\omega n + \phi(\omega))$ (called also frequency components)
- Euler formula:
$$\cos(\omega) = \frac{1}{2} \cdot (e^{j\omega} + e^{-j\omega}) \tag{12}$$
- EXAMPLE: Spectrum of a real-valued signal $x[n]$
- Notation: $x[n] \overset{DTFT}{\rightarrow} X(e^{j\omega})$
- $X(e^{j\omega})$ can be decomposed into two real-valued functions via
$$X(e^{j\omega}) = \left|X(e^{j\omega})\right| e^{j\phi(\omega)} \tag{13}$$
    - *magnitude spectrum* $\left|X(e^{j\omega})\right|$
    - *phase spectrum* $\phi(\omega)$

DTFT spectrum $X(e^{j\omega})$ II

- DTFT spectrum is independent of time (it does not have any time resolution, analyzes the signal as a whole)
- This analysis is suitable for **stationary signals:**
  - expected statistical properties (and consequently the spectrum) does not change in time
  - e.g. hum of a fan, constant vibrations of rotating machines, white noise, constant tone/accord
- This type of analysis is also suitable for study of LTI systems, which do not change their properties in time

Part I

Application of DTFT to impulse response $h[n]$, frequency response

Frequency response - motivation

- Both difference equation (LCCDE) and impulse response describe the LTI system in the time-domain
- These system models do not provide any information, how the LTI system influences the DTFT spectrum of a signal
- The influence of system on an input signal in the frequency domain is given by the **frequency response**

What is frequency response:

- The response of a system given by impulse response $h[n]$ to signal $x[n]$ is given in the time-domain via

$$y[n] = h[n] * x[n]. \tag{1}$$

- Applying the convolution theorem of the DTFT gives us the response in the frequency domain

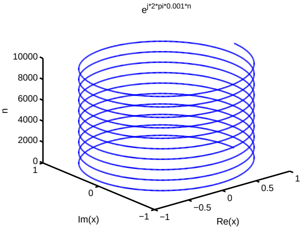
$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}), \tag{2}$$

$Y(e^{j\omega})$ - DTFT spectrum $y[n]$, $X(e^{j\omega})$ - DTFT spectrum $x[n]$

- Thus, the frequency response is a ratio between the output spectrum $Y(e^{j\omega})$ and the input spectrum $X(e^{j\omega})$.

Eigenfunctions of the LTI system:

- Sequences that pass through the LTI system unchanged, up to a change in *complex amplitude*
- If input into the system is $x[n]$, then output $y[n] = \lambda x[n]$, where $\lambda \in \mathcal{C}$ is the *eigenvalue* corresponding to the eigenfunction $x[n]$.
- The complex eigenvalue has absolute value and argument, which can be interpreted as *amplification* and *delay* of the eigenfunction $x[n]$, respectively.
- Eigenfunction of the LTI systems have the form $x[n] = e^{jn\omega_0}$ $\omega_0 \in \mathcal{R}$, $-\infty < n < \infty$
- QUESTION: Why complex exponentials?



Frequency response III

- Eigenvalue corresponding to the complex exponential $e^{j\omega_0 n}$ is denoted $H(e^{j\omega_0})$
- The function, which describes the dependency of the eigenvalues on frequency ω is denoted by $H(e^{j\omega})$ and called the **frequency response** (FR) of the LTI system
- FR states, how the complex exponential on frequency ω_0 is **amplified** and **delayed** when passed through LTI system
- FR is computed by application of DTFT to impulse response $h[n]$ of the system

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-jn\omega} \tag{3}$$

- FR is a complex-valued function of a real-valued independent variable ω
- FR is the third description of the LTI system (along with impulse response and difference equation)

Frequency response IV

- Discrete signal can be (using DTFT) decomposed into a spectrum of complex exponentials (or harmonic functions for $x[n] \in \mathcal{R}$)
- If the input into the LTI system is in the form

$$x[n] = \sum_{k=1}^{K} \alpha_k e^{jn\omega_k} \tag{4}$$

then the output is

$$y[n] = \sum_{k=1}^{K} \alpha_k H(e^{j\omega_k}) e^{jn\omega_k} \tag{5}$$

- The response of a system with impulse response $h[n]$ to signal $x[n]$ is given by

$$y[n] = h[n] * x[n], \tag{6}$$

then (due to convolutional theorem of DTFT) this results in the frequency domain into

$$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega}), \tag{7}$$

$Y(e^{j\omega})$ - DTFT spectrum of $y[n]$, $X(e^{j\omega})$ - DTFT spectrum of $x[n]$

Properties of the frequency response

- **Periodicity**
- Frequency response is *periodical* with period 2π
- QUESTION: Why is it so?
- **Symmetry** for systems with real-valued impulse response $h[n]$
- For such a system, FR is *conjugate symmetric* function of ω
  $H(e^{-j\omega}) = H^*(e^{j\omega})$
- This stems from the symmetry properties of the DTFT
- Due to this, the cosine and sine functions are eigenfunctions of such systems along with the exponentials

Frequency response V

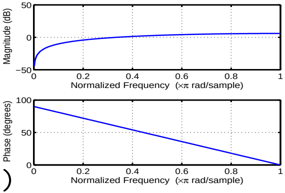
- FR is usually formulated as a pair of functions:
  - *magnitude response* $\left|H(e^{j\omega})\right|$
  - *phase response* $\phi(e^{j\omega})$

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\phi(e^{j\omega})} \tag{8}$$

- A more practical substitute of phase characteristic is the **phase delay**

$$\tau_p(\omega) = -\frac{\phi(e^{j\omega})}{\omega} \tag{9}$$

- It states the delay of signal $e^{j\omega n}$ after a pass through LTI system in samples (in contrast to the angle given by phase characteristic)
- MATLAB: `[PHI,W]=phasedelay(b,a,N)`



Frequency response VI

- Another measure of the delay when signal passes through the LTI systems is the **group delay**

$$\tau_g(\omega) = -\frac{d\phi(e^{j\omega})}{d\omega} \tag{10}$$

- It states a delay (in samples) of a narrow-band signal consisting of a "group" of harmonic components with frequencies close to ω_0
- Let us consider a signal $a[n]$ modulated by a carrier harmonic wave $\cos(\omega_0 n)$, i.e., $x[n] = a[n] \cdot \cos(\omega_0 n)$. The group delay gives a shift of the amplitude envelope $a[n]$ when $x[n]$ passes through the LTI system with phase response $\phi(e^{j\omega})$.
- Phase and group delay are equal at systems with linear phase:

$$\phi(e^{j\omega}) = -\alpha\omega, \tau_p(\omega) = -\frac{-\alpha\omega}{\omega}, \tau_g(\omega) = -\frac{-d\alpha\omega}{d\omega} \tag{11}$$

- MATLAB: `[Gd,W]=grpdelay(b,a,N)`

Part II

Other applications of DTFT

DTFT applications I

- **Frequency response** of a system given by a difference equation
- Frequency response $H(e^{j\omega})$ is the DTFT of the impulse resp. $h[n]$
- FIR systems are often given by $h[n]$, the computation of $H(e^{j\omega})$ is straightforward there
- IIR systems are usually given by their difference equation:

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] - \sum_{k=1}^{p} a[k]y[n-k] \tag{12}$$

- Question and example: How do we obtain frequency response from the difference equation?
- Due to linearity and the shift theorem, $H(e^{j\omega})$ is given by

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{\sum_{k=0}^{q} b[k]e^{-jk\omega}}{1 + \sum_{k=1}^{p} a[k]e^{-jk\omega}} \tag{13}$$

- MATLAB: `[H,W]=freqz(b,a,N)`

DTFT applications II

- **Computation of convolutions**: (for infinite sequences)
- DTFT maps convolution in the time-domain into multiplication in the frequency domain
- This fact is used to evaluate convolutions of infinite sequences
- EXAMPLE: Computation of convolution using DTFT

DTFT applications III

- **Analytical solution of difference equations** with zero initial conditions
- *In other words:* analytical computation of filtering
  1. Substitution of the input signal and transformation of the equation into the frequency-domain
  2. Expression of the output
  3. Inverse DTFT
- EXAMPLE: Analytical solution of a difference equation using DTFT

DTFT applications IV

- **Inverse system** to a system given by impulse response $h[n]$ is such a filter, whose impulse response $g[n]$ fulfills

$$h[n] * g[n] = \delta[n]. \tag{14}$$

- Frequency response $G(e^{j\omega})$ therefore fulfills

$$G(e^{j\omega}) = \frac{1}{H(e^{j\omega})} \tag{15}$$

- Not every system is practically invertible (the inverse system may be unstable)
- Inverse system to a causal system may be non-causal
- *In other words:* the existence of a causal and stable inverse system is not always guaranteed
- EXAMPLE: Inverse system and its causality

Part I

Digital filters

Filters

- **Digital filter** is an algorithm, which transforms *input discrete signal* into another *output discrete signal*.
- The process may include low-pass filtering (smoothing), band-pass filtering, interpolation, generation of derivatives etc.
- Filter is thus just another name given to a *discrete system*, when it is used in the context of signal processing.
- Filters thus have pro mathematical properties, which we defined earlier for discrete systems ... (linearity, causality, stability)
- ... and also other properties, which stem from the frequency response and concern signal processing operations.

- **Allpass filter**: - rozsiruje schopnosti inverznich filtru
- The magnitude of an *allpass filter* is constant and independent on frequency

$$\left|H(e^{j\omega})\right| = c, c \in \mathcal{R} \tag{1}$$

- **Frequency selective filters**:
- *Low-pass, high-pass, band-pass, band-stop*
- Presented responses are ideal and unachievable. In practice, these need to be closely approximated.
- *Stop band* - $\left|H(e^{j\omega})\right| = 0$ - zádrzne pasmo
- *Pass band* - $\left|H(e^{j\omega})\right| = 1$ - propustne pasmo
- *Cutoff frequency* - Frequency separating pass and stop bands

- frekvence na ose x

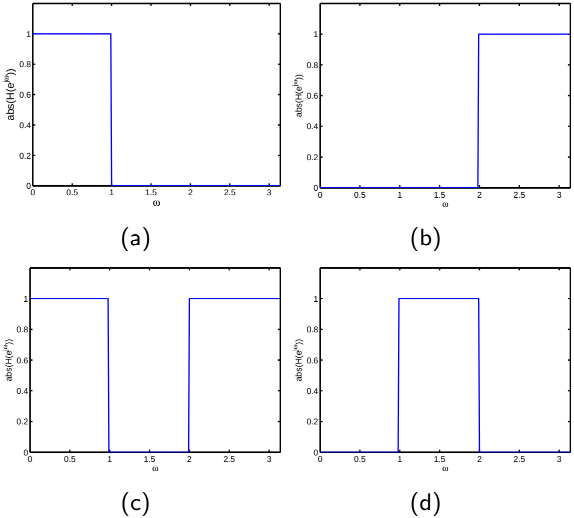


Figure: (a) Low-pass (b) High-pass (c) Band-pass (d) Band-stop

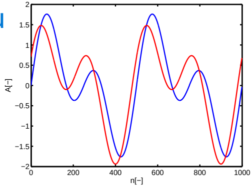
Part II

Filters with linear phase

Filters with linear phase I

- For many filtering applications, the *magnitude response* is of prime interest
- In some cases, it is important to consider the influence of filtering on the phase spectrum as well
- **Phase response** gives the change of phase of the harmonic function at specific frequency when it passes through the filter
- In the case, when various frequency components are delayed differently, the *phase distortion* arises
- This distortion modifies the shape of the signal in the time-domain, even when all frequency components should pass the filter
- This behavior is undesirable, for example when the signal should be analyzed in the time-domain (ECG/EEG)
- Systems/filters, which do not deform the phase spectrum are denoted as filters with *linear phase*

- fáze popisuje tvar signalu
v time domain
(dulezite napr. ECG)

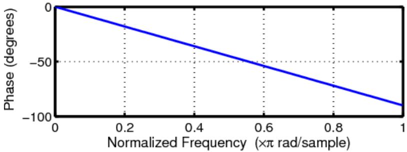


Filters with linear phase II

- Digital filter has the **linear phase** when

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega}, \quad \alpha \in \mathcal{R} \tag{2}$$

$A(e^{j\omega}) \in \mathcal{R}$ - amplitude (can be positive and negative)

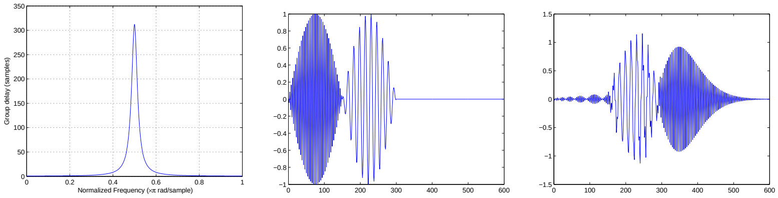


- Systems with linear phase have constant phase/group delay

$$\tau_g(\omega) = \tau_p(\omega) = \alpha. \tag{3}$$

- These filters only delay the processed signal in the time-domain and (almost) do not distort it (if the spectrum of the signal is contained in the pass-band of the filter).

All pass filter: $H(e^{j\omega}) = \left(\frac{e^{-2j\omega}+0.95^2}{1+0.95^2 e^{-2j\omega}}\right)^8$



(a) (b) (c)

(a) Group delay $\tau(\omega)$, (b) Modulated input signal $x[n]$ -Carrier frequencies $\pi/2$ a $\pi/8$.
(c) Output signal $y[n] = \text{IDTFT}(H(e^{j\omega})X(e^j\omega))$

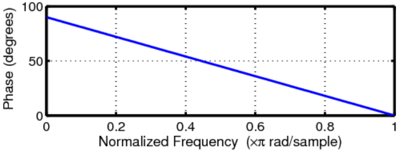
This example is inspired by a lecture given by professor Barry Van Veen (University of Wisconsin) "Characterizing Filter Phase Response", available on Youtube.

Filters with linear phase III

- Digital filter has the **generalized linear phase** when

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j(\alpha\omega-\beta)}, \quad \alpha,\beta \in \mathcal{R} \tag{4}$$

$A(e^{j\omega}) \in \mathcal{R}$ - amplitude (can be positive and negative)



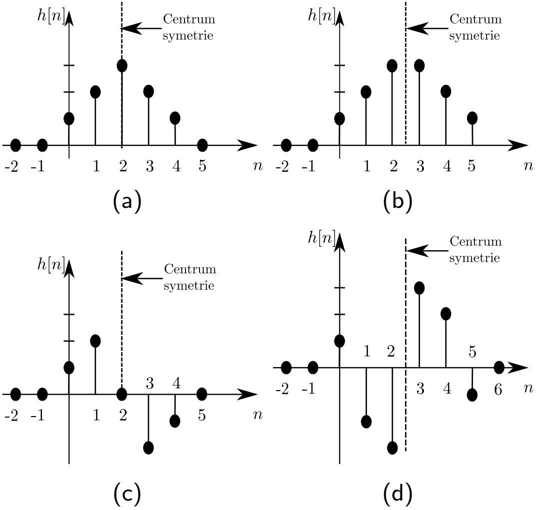
- Systems with generalized linear phase have constant group delay (not the phase delay)

Filters with linear phase IV

- A filter has the **(generalized) linear phase**, is stable, causal and has real-valued impulse response if the following conditions are fulfilled.
  1. The impulse response $h[n]$ must be finite (FIR)
  2. The impulse response $h[n]$ must feature specific form of symmetry
- Based on these requirements, there are four types of FIR filters with linear phase. Let $N+1$ be the length of $h[n]$ then
  1. Type 1 - symmetric $h[n]$, N is even number, linear phase
  2. Type 2 - symmetric $h[n]$, N is odd number, linear phase
  3. Type 3 - antisymmetric $h[n]$, N is even number, gen. linear phase
  4. Type 4 - antisymmetric $h[n]$, N is odd number, gen. linear phase
- For symmetric impulse response $h[n] = h[N-n]$ holds, whereas for the antisymmetric ones $h[n] = -h[N-n]$.
- Phase/group delays of filters of Types 1 and 2 (with linear phase) fulfill

$$\tau_g(\omega) = \tau_p(\omega) = \alpha = \frac{N}{2} \tag{5}$$

- DETAILS: Boaz Porat, *A course in digital signal processing*, 256 / chapter 8.4.3



Impulse responses for various types of (generalized) linear phase filters
(a) Type 1, (b) Type 2 (c) Type 3, (d) Type 4

Part III

Filter interconnection

Filter interconnection I

- **Serial / cascade interconnection:**
- Overall impulse response is a convolution of the partial ones
- $h[n] = h_1[n] * h_2[n]$
- Overall frequency response is a multiplication of the partial ones
- $H(e^{j\omega}) = H_1(e^{j\omega})H_2(e^{j\omega})$
- QUESTION: What holds for the magnitude and phase?

- **Parallel interconnection:**
- Overall impulse response is a sum of the partial ones
- $h[n] = h_1[n] + h_2[n]$
- Overall frequency response is a multiplication of the partial ones
- $H(e^{j\omega}) = H_1(e^{j\omega}) + H_2(e^{j\omega})$
- QUESTION: What holds for the magnitude and phase?

Filter interconnection II

- **Feedback loop:**
- Frequency response:

$$H(e^{j\omega}) = \frac{H_1(e^{j\omega})}{1 - H_1(e^{j\omega})H_2(e^{j\omega})} \tag{6}$$

- QUESTION: How is this formula derived?

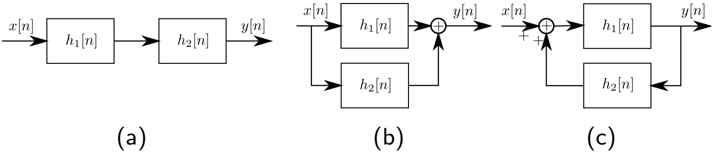


Figure: (a) Cascade (b) Parallel interconnection (c) Feedback loop

Part IV

Decibel, Signal-to-Noise Ratio

Decibel

Decibel [dB]: - zesileni na 2x energie amplitudy je o 3 dB

- Logaritmic unit expressing ratio of two values of a physical quantity (often of energy). - log lepe popisuje radove hodnoty
- It can be used as a measure of an attenuation/amplification of a signal after filtering
- It is defined as a ratio of an investigated and a reference variable

$$L_{dB} = 10 \log_{10} \left(\frac{E_{x[n]}}{E_{x_{ref}[n]}} \right) \tag{7}$$

- **Amplitude** of a periodic/harmonic signal is its maximal change (height of the peak) within a single period.
- For periodic signals, the SNR value is computed using the amplitude A as

$$L_{dB} = 20 \log_{10} \left(\frac{A_{x[n]}}{A_{x_{ref}[n]}} \right) \tag{8}$$

Signal-to-Noise Ratio

SNR:

- Quantity measuring a ratio of energy of a desired signal $s[n]$ and the undesired background noise $v[n]$ in the mixture

$$x[n] = s[n] + v[n] \tag{9}$$

- Usually given in decibels as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_{s[n]}}{E_{v[n]}}\right) \tag{10}$$

- In denoising applications another related quantity is stated, the **SNR improvement** defined by

$$\mathrm{SNR}_{\mathrm{imp}} = \mathrm{SNR}_{\mathrm{enh}} - \mathrm{SNR}_{\mathrm{orig}}, \tag{11}$$

  where $\mathrm{SNR}_{\mathrm{orig}}$ and $\mathrm{SNR}_{\mathrm{enh}}$ are SNRs prior/after the enhancement.

- Prior the computation, it is necessary to decompose the signal into the desired and the noise components, since these are usually unknown.

SNR in the context of acoustic signal denoising

Denoising:

- Removal / Suppression of undesired signal component (noise, interference) in the audio signal.
- Evaluation proceeds via *objective* / *subjective* criteria (SNR / listening tests)
- **Example** of interfering speech suppression in real acoustic conditions (beamforming, 4 microphones)

Mixture - two speakers (SNR = 0.7 dB),
Interference (SNR = -10.3 dB, interference amplification by 11dB)
Desired speaker (SNR = 8.3dB, SNR_{imp} = 7.6 dB),
Desired speaker (distortion, SNR = 18.2 dB, SNR_{imp} = 17.5 dB)

Part I

Sampling

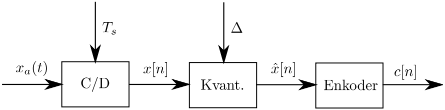
- Most of the discrete signals arises via *sampling* of some analog quantity
  EXAMPLE: audio recording, measurement of biomedical signals etc.
- Transformation of analog signals into discrete sequences is denoted as *A/D conversion* (Analog to Digital Conversion)
- A reverse process is the *D/A conversion* - Digital to Analog Conversion
- *Sampling theorem* states, when is the analog signal uniquely determined by its samples

A/D conversion

- A transformation of the analog signal $x_a(t), t \in \mathcal{R}$ to a discrete sequence $\hat{x}[n]$ and subsequent encoding
- Amplitude $x_a(t)$ is an arbitrary real number, $\hat{x}[n]$ is quantized - has finite number of amplitude levels
- **A/D conversion** typically consists of three parts:
  1. *(Ideal) sampling* - selection of values $x_a(t)$ at times equal to integer multiples of *sampling period* T_s

$$x[n] = x_a(nT_s) \tag{1}$$

  2. *Quantization* of continuous amplitudes in $x[n]$ into a discrete set of amplitude values, gives $\hat{x}[n]$
  3. *Encoding* of the discrete values $\hat{x}[n]$ into a sequence of binary code-words $c[n]$



- Sampling period $T_s[s, s/sample]$, sampling frequency $F_s[Hz, sample/s]$, frequency $F[Hz]$
- *Equidistant sampling*- multiplication of $x_a(t)$ with periodic sequence of Dirac pulses
- DETAILS around equidistant sampling
- Sampling maps frequencies of the analog signal $-\infty < \Omega < \infty$ to digital frequencies $-\pi < \omega < \pi$

$$\omega = \Omega T_s = \frac{\Omega}{F_s} = \frac{2\pi F}{F_s} \tag{2}$$

- The frequency components corresponding to ω repeat periodically with period 2π
- DETAILS to transition from analog to digital frequencies

- vzorkovací perioda = vzdalenost mezi dvma vzorky

Sampling II

- **Sampling theorem** - The signal $x_a(t)$ is *band-limited*, if it contains only components with frequency lower than Ω_0 (analog spectrum $X_a(\Omega) = 0; |\Omega| \geq \Omega_0$)
- The band-limited signal $x_a(t)$ can be reconstructed from its samples $x_a(nT_s)$ if

$$\Omega_s = \frac{2\pi}{T_s} \geq 2\Omega_0 \tag{3}$$

  Ω_0 - Nyquist frequency
- Question: What happens when the sampling theorem does not hold?
- *Aliasing* - Distortion of the digital signal arising during sampling with a low sampling frequency.
  In frequency domain, the periods of DTFT spectrum $X(e^{j\omega})$ overlap and sum together.
  In the time-domain, the signal reconstructed from the samples of $x(nT_s)$ is different than the original $x_a(t)$.
- Most signals are not bend-limited, an anti-aliasing analog filter (low-pass) must be applied, in order to avoid aliasing.

Quantization I

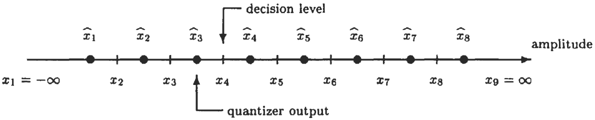
- **Quantization** - Transformation of a continuous amplitude of $x[n]$ to discrete finite set of amplitudes
- May be interpreted as a form of rounding

$$\hat{x}[n] = Q(x[n]) \tag{4}$$

- Quantization splits the continuous amplitude of $x[n]$ to L non-overlapping intervals I_k using $L+1$ decision levels $x_1, x_2, \dots, x_{L+1}$

$$I_k = [x_k, x_{k+1}], \quad k = 1, 2, \dots, L \tag{5}$$

- If $x[n]$ belongs to the interval I_k, quantization assigns to $\hat{x}[n]$ the value $\hat{x}_k$



Source: MONSON H. HAYES, Schaum's Outlines of Digital Signal Processing

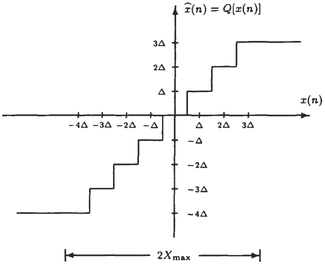
Quantization II

- **Quantization step/resolution** - the width of the interval I_k, often constant for all intervals - *linear/equidistant quantization*
- Number of decision levels is often $L = 2^B + 1$, due to subsequent binary coding, B-Number of bits
- **Quantization error** - $e[n] = x[n] - Q(x[n])$
- **SQNR[dB]** - *Signal to Quantization Noise Ratio*

$$\mathrm{SQNR} = 10 \log \frac{\sigma_x^2}{\sigma_e^2} \approx 6.02B[dB] \tag{6}$$

  SQNR thus grows by about 6dB with every added bit (doubles the number of encoding levels)
  ASSUMPTION: signal is amplified such that it covers all quantization level equally
- **In practice:** Provided that the assumption holds, SQNR is very high, due to high number of quantization levels (approx. 96dB for 16-bit quantizer)
  Thus, $x[n]$ usually can be considered equal to $\hat{x}[n]$

Quantization III, Encoding



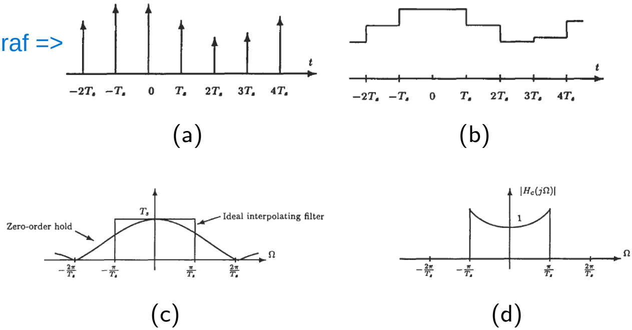
Source: MONSON H. HAYES, Schaum's Outlines of Digital Signal Processing

- **Encoder** - An algorithm/device assigning a binary word to each quantization level
- *Many numerical types:* integers, signed/unsigned, fixed/floating point etc.
- *Many encoding schemes:* e.g., *two's complement* for signed integers

D/A conversion I

- **D/A conversion** - Transformation of discrete sequence into analog signal
- If the *sampling theorem* is fulfilled, then the analog signal can be uniquely reconstructed from its samples
- The exact reconstruction is however unavailable, due to quantization errors
- In practice: if quantization is sufficiently fine (SQNR is high), these errors can be neglected
- Ideal (theoretical) D/A conversion proceeds in two steps:
  1. Continuous sequence of pulses $x_s(t)$ is generated using the samples of $x[n]$
  2. Ideal analog low-pass filter - *a reconstruction filter* is applied to $x_s(t)$
- DETAILS - Ideal D/A conversion
- In practice: ideal reconstruction filter is not realizable
- Instead: *zero-order hold* and *compensation filters* are used
- DETAILS - Real-world D/A conversion

hrebenovy graf =>

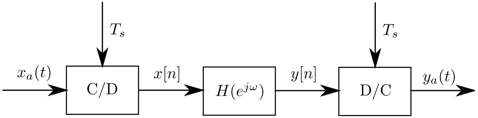


Source: MONSON H. HAYES, Schaum's Outlines of Digital Signal Processing

(a) Continuous sequence of pulses $x_s(t)$ - hrebenovy graf
(b) Zero-Order Hold signal
(c) Analog frequency response: Ideal low-pass, zero-oder hold
(d) Analog frequency response: Ideal compensation filter

D/A conversion III

- A/D and D/A transducers are often used when an analog signal should be processed by a discrete system (system control, music processing etc.)
- This scenario assumes:
  - In theory: signal is not quantized, in practice: quantization levels are sufficiently fine
  - in theory: ideal low-pass reconstruction filter is used, in practice: zero-order hold and compensation filter instead
- When all assumptions (or their practical approximations) hold, the overall cascade can be considered as continuous system, and any influence of sampling can be neglected



Part II

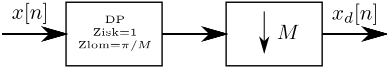
Sampling - Sample rate conversion

Sample rate conversion

- A frequent task in signal processing
- Two possible implementations:
  1. In theory: D/A conversion and sampling via a different sampling frequency F_s (infeasible)
  2. In practice: *resampling* directly in the time-domain
- The types:
  1. Decrease of F_s by an integer factor
  2. Increase of F_s by an integer factor
  3. Change of F_s by a rational factor

Decrease of sample rate by an integer factor

- **Down-sampling**
- When down-sampling M-times, the down-sampled signal $x_d[n]$ contains every Mth sample of the original $x[n]$
- BEWARE - down-sampling generally leads to aliasing!
- DETAILS - Aliasing by down-sampling
- *Aliasing prevention:* Filtration of $x[n]$ with a low-pass filter with cutt-off frequency $\omega_c = \pi/M$
- Cascade of low-pass filtering and down-sampling is called *decimation*

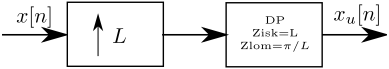


Increase of sample rate by an integer factor

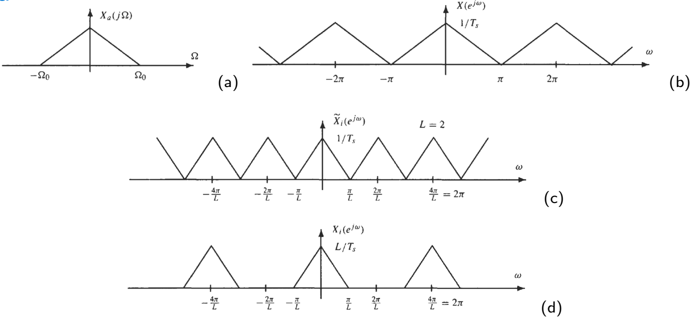
- **Up-sampling**
- When up-sampling L-times, the up-sampled signal $x_u[n]$ contains the samples of $x[n]$ with $L-1$ zeros between every two samples

$$x_u[n] = \begin{cases} x[n/L] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & \textit{elsewhere} \end{cases} \tag{7}$$

- Approximation of zero samples is performed by low-pass filtering with cut-off frequency π/L and gain L.
- Cascade of up-sampling and low-pass filtering is called interpolation
- DETAILS - Interpolation and spectrum



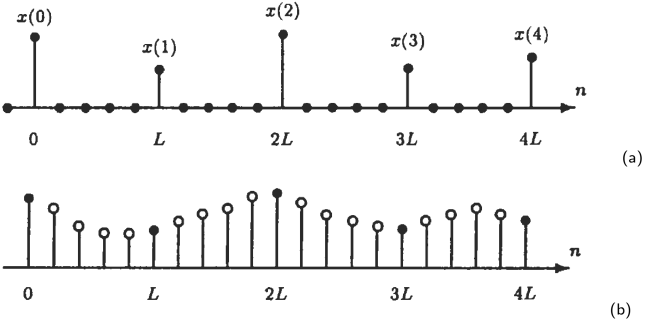
- prevzorkování je vynasobeni analog. signalu hrebenovy grafem => vede k "namnozeni" trojuhelniku



Source: MONSON H. HAYES, Schaum's Outlines of Digital Signal Processing

(a) Spectrum of continuous band-limited signal
(b) Spectrum of sampled signal
(c) Spectrum of upsampled signal
(d) Spectrum of interpolated signal (up-sampled and low-pass filtered)

Interpolation in the time-domain



Source: MONSON H. HAYES, Schaum's Outlines of Digital Signal Processing

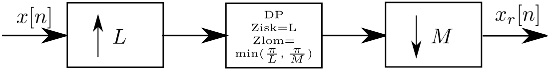
(a) Up-sampled signal $x_u[n]$
(b) Interpolated signal (up-sample and low-pass filtered)

Change of the sampling frequency by rational factor

- The sampling rate conversion by rational-factor $\frac{L}{M}$ is performed by interpolation L-times followed by decimation M-times.
- The cascade can be replaced by a single low-pass filter with cut-off frequency

$$\omega_c = \min\left\{\frac{\pi}{M}, \frac{\pi}{L}\right\} \tag{8}$$

and gain L.



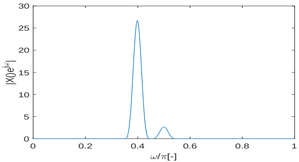
- EXAMPLE: Change of the sampling frequency by a rational factor
- MATLAB: `resample(x,L,M);`

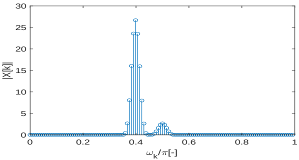
Part III

Spectral analysis - signals with finite duration

Spectral analysis - motivation

- We have discussed the analysis of the signals in the frequency domain via DTFT
- Here, the digital frequency $\omega \in \mathcal{R}$; even when the analyzed signal is of finite duration, its DTFT spectrum is comprised (in theory) of infinitely many frequency components
  - This is advantageous for system analysis (the response to any frequency is known)
  - Unsuitable for analysis of common signals with short duration
- **Discrete Fourier Transform (DFT)**
  - Form of Fourier transform suitable for finite of periodical signals
  - Its output is a "more compact" spectrum (sequence of the same length as the original signal)
  - It retains most (but not all) advantageous properties of the DTFT (invertibility, linearity)





Part IV

Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT)

- DTFT allows to transform the discrete sequence $x[n]$ to continuous function of digital frequency ω, i.e., $X(e^{j\omega})$.
- Considering discrete signal $x[n]$, the function $X(e^{j\omega})$ is DTFT spectrum
- Unique inverse transform is possible, provided that values of $X(e^{j\omega})$ are known for all frequencies $\omega \in [0, 2\pi)$
- For finite-length $x[n]$ ($x[n] \neq 0, 0 < n < N-1$), $x[n]$ can be reconstructed using N suitably selected frequency points.
- DTFT is defined as:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega} \tag{9}$$

- Provided that $x[n]$ has finite duration then only N elements of the sum in non-zero

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n]e^{-jn\omega} \tag{10}$$

Discrete Fourier Transform II

- By uniformly sampling one period of $X(e^{j\omega}), \omega \in [0, 2\pi)$ (frequency resolution being $2\pi/N$) we use frequencies:

$$\omega[k] = \frac{2\pi k}{N}, \quad 0 \leq k \leq N-1, \tag{11}$$

and the original $x[n]$ can be uniquely reconstructed using these points.

- **Discrete Fourier Transform** is thus defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad 0 \leq k \leq N-1 \tag{12}$$

- Sequence of complex-valued signals $X[k]$ is called N-point DFT of discrete sequence $x[n]$
- If the formula (12) is evaluated for all k (not just $0 \leq k < N$), an infinite periodic DFT image with period N is obtained (denoted by $\tilde{X}[k]$).

Discrete Fourier Transform III

- **Inverse Discrete Fourier Transform** is defined by:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi nk/N}, \quad 0 \leq n \leq N-1 \qquad (13)$$

- Again, if the formula (13) is evaluated for all n, an infinite periodic extension of $x[n]$ with period N is obtained (denoted by $\tilde{x}[n]$)
- Equations (12) and (13) form the DFT pair

$$x[n] \overset{DFT}{\Longleftrightarrow} X[k] \qquad (14)$$

- By application of the N-point DFT to a digital signal, a N-point **discrete complex spectrum** $X[k]$ is obtained, which corresponds to sampling of the DTFT spectrum at frequencies $\frac{2\pi k}{N}, \; k = 0 \ldots N-1$
- DFT is advantageous due to simplicity of the computation on digital computers
- EXAMPLE: Computation of 4-point DFT.

Part I

Discrete Fourier Transform (DFT)

Comparison of DFT and DTFT

- **DTFT:** *infinite, aperiodic, discrete signals* $x[n]$
- DTFT output: *continuous, complex, periodic spectrum* $X(e^{j\omega})$ (period 2π)
- **DFT:** *infinite, periodic, discrete signals* $\tilde{x}[n]$
- In practice: the computation is performed on a single period $x[n]$ (of duration N samples) of the periodic signal $\tilde{x}[n]$
- DFT output: *discete, complex, periodic* spectrum $\tilde{X}[k]$ (period N)
- In practice: only single period $X[k]$ (N-point DFT) of the periodic spectrum $\tilde{X}[k]$ is computed
- The values of the *DFT spectrum* can be obtained by sampling of DTFT spectrum at frequencies $\omega = \frac{2\pi k}{N}, k = 0, 1, \dots N-1$
- The distance between frequency samples $\Delta\omega = \frac{2\pi}{N}$ is called *frequency resolution*

DFT properties I

- **Linearity**: Let signals $x_1[n]$ and $x_2[n]$ have spectrum $X_1[k]$ and $X_2[k]$. Then it holds

$$ax_1[n] + bx_2[n] \overset{DFT}{\Longleftrightarrow} aX_1[k] + bX_2[k] \qquad (1)$$

- Sequences must have equal duration, if not, the shorter one is zero-padded
- **Symmetry**: If $x[n] \in \mathcal{R}$, then $X[k]$ is conjugate symmetric:

$$X[k] = X^*[-k] = X^*[N-k]_N \qquad (2)$$

- If $x[n]$ is imaginary, then $X[k]$ is conjugate antisymmetric:

$$X[k] = -X^*[-k] = -X^*[N-k]_N \qquad (3)$$

- EXAMPLE: DFT symmetry

- **Simplified notation**: - zkratky

$$W_N \overset{def.}{=} e^{-j2\pi/N} \tag{4}$$

$$W_N^{nk} \overset{def.}{=} e^{-j2\pi kn/N} \tag{5}$$

- **Circular shift**: Circular shift by n_0 samples is defined by

$$(x[n-n_0])_N R_N[n] = \tilde{x}[n-n_0] R_N[n] \tag{6}$$

where $R_N[n]$ is rectangular window of length N

- EXAMPLE: Circular shift - posun o okno
- Circular shift in the time domain causes in the frequency domain a change of the phase spectrum

$$(x[n-n_0])_N R_N[n] \overset{DFT}{\Longleftrightarrow} W_N^{n_0 k} X[k] \tag{7}$$

DFT properties III

- **Circular convolution:** Let $x[n]$ and $h[n]$ be two finite sequences with N-point DFTs $X[k]$ and $H[k]$, then sequence with DFT equal to $Y[k] = H[k]X[k]$ is given by a formula

$$y[n] = x[n] \circledN h[n] = \left[\sum_{k=0}^{N-1} h[k]\tilde{x}[n-k]\right] R_N[n] \qquad (8)$$

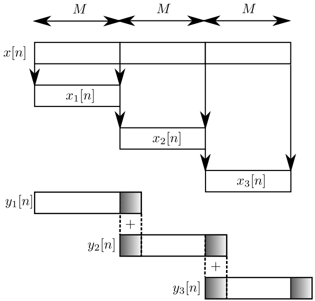
- This is a convolution of $h[n]$ with periodical $\tilde{x}[n]$, evaluated using only single period of $\tilde{x}[n]$
- *Circular convolution* is generally not equal to *linear convolution* (filtering), even lengths of these two operations differ.
- Using suitable zero-padding, both operations coincide (give the same result).
- Then, the circular convolution can be used for fact computation of linear convolution (using Fast Fourier Transform - FFT)

Computation of linear convolution via the circular one

- DFT and circular convolution can be used for effective computation of linear convolution
- Having two finite sequences $h[n]$ and $x[n]$ with lengths N_1 and N_2, respectively, the linear convolution can be computed as follows:
  1. Zero-padding of $h[n]$ and $x[n]$ to length $N \geq N_1 + N_2 - 1$
  2. Computation of N-point DFT of sequences $h[n]$ and $x[n]$
  3. Multiplication $Y[k] = H[k]X[k]$
  4. Inverse DFT of $Y[k]$
- This procedure becomes *much less computationally demanding* then the definition formula of convolution, when FFT is used to compute the DFT.
- Unlike the definition formula, this procedure is not suitable for very long sequences $x[n]$
  - It requires the knowledge of the whole $x[n]$ (which disables real-time processing)
  - $h[n]$ is usually much shorter than $x[n]$ (a lot of zero-padding)
- These negatives are mitigated using *block-wise computation of convolution*

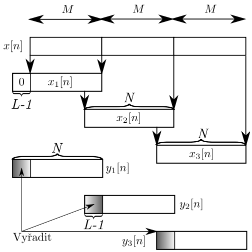
Overlap-Add (Block-wise processing)

- **Overlap-Add** (OA,OLA) is an efficient approach to compute convolution of a *long signal* $x[n]$ with impulse response of a FIR filter $h[n]$ (length L)
  - Signal $x[n]$ is split to non-overlapping sequences $x_i[n]$ of length M
  - Output signal $y[n]$ can be expressed as a sum of partial convolutions $y_i[n] = x_i[n] * h[n]$
  - The partial convolutions $y_i[n]$ have lengths $N = L + M - 1$ and are added with shift M ($L - 1$ samples of $y_i[n]$ and $y_{i-1}[n]$ thus overlap)
  - Computation of $y_i[n]$ is performed via zero-padded circular convolution using FFT



Overlap-Save (Block-wise processing)

- **Overlap-Save:** is an alternative to Overlap-Add, it efficiently computes convolution of a *long signal* $x[n]$ with impulse response of a FIR filter $h[n]$ (length L)
- $x[n]$ is split (with overlap of $L-1$ samples) into subsequences $x_i[n]$ of length N
- Overlap-Save computes the classing convolution via suitable concatenation of parts of circular convolutions involving subsequences $x_i[n]$.
- PRINCIPLE: Considering the circular convolution $x_1[n] \textcircled{N} h[n]$, the first $L-1$ samples differ from $x_1[n] * h[n]$, the remaining $M = N - L + 1$ samples are equal.
- These M samples constitute one interval of the output $y[n]$

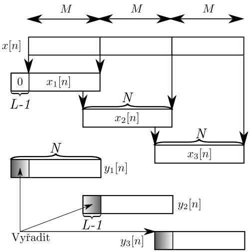


Overlap-Save II - Algorithm

1. Formation of the sequence $x_1[n]$

$$x_1[n] = \begin{cases} 0 & 0 \leq n < L-1 \\ x[n-L+1] & L-1 \leq n \leq N-1 \end{cases} \tag{9}$$

2. Computation of $x_1[n] \circledN h[n]$ using DFT.
   $L-1$ samples differ from *linear convolution*.
   Last $M = N - L + 1$ values of $x_1[n] \circledN h[n]$ are the first samples of $y[n]$



3. Formation of subsequence $x_2[n]$, where first $L-1$ samples overlap with the last samples of $x_1[n]$
4. Computation of $x_2[n] \circledN h[n]$ using DFT.
   $L-1$ samples differ from the linear convolution.
   Last $M = N - L + 1$ values form the second interval of $y[n]$
5. Steps 3. and 4. are repeated until the whole linear convolution is evaluated

Part II

Fast Fourier Transform (FFT)

Fast Fourier Transform

- **Fast Fourier Transform** is a group of algorithms allowing optimized computation of DFT and IDFT
- DFT transforms finite (or infinite periodic) sequence of time-domain samples into finite sequence of frequency components
- Computational complexity od DFT computed by definition is $O(N^2)$
- FFT is able to compute the same result in $O(N \log(N))$ operations
- The difference in computational complexity becomes apparent for growing N
- Due to FFT, the DFT algorithm is used into many scientific areas (signal processing, image processing, solution of differential equations etc.)

Fast Fourier Transform II

- Many FFT algorithms stem from factorization of the sample number N
- The N-point DFT of $x[n]$ is computed via several transforms applied to subsequences of $x[n]$
- However, even implementations suitable for prime N have been discovered
- The FFT algorithm can easily compute the IDFT (which differs by a sign in the exponent and normalization)
- **Comparison of FFT and DFT**:
- Evaluation of DFT by definition requires N^2 complex multiplications and $N(N-1)$ complex summations
- The most famous FFT version radix-2 Cooley-Tukey is suitable for N equals power of 2
- It requires $(N/2)\log_2(N)$ complex multiplications and $N\log_2(N)$ complex summations

Fast Fourier Transform III

- **Radix-2 Cooley-Tukey FFT:** Algorithm designed for sequences of length $N = 2^k, \; k \in \mathcal{Z}$
- Computational savings are achieved due to periodicity of the complex exponentials and the possibility to compute N-point DFT using two $N/2$-point DFTs
- The algorithm is recursive but can be computed non-recursively, if the input samples are suitably permuted
- DETAILS: Radix-2 Cooley-Tukey FFT

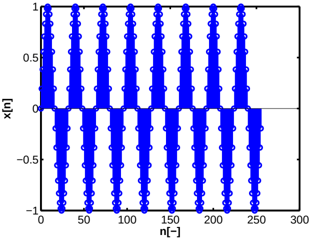
Part I

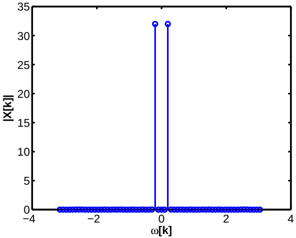
Practical spectral analysis, windowing

Practical spectral analysis I

- Spectral analysis: search for a regular inner structure / periodicity in a general signal - 3 typy signalu dle periodicity
- Complications: finite signal length, potential non-stationarity, noise..

1) - **Signals with harmonic structure** (energy focused into narrow bands, ideally a sparse spectrum)
- EXAMPLE: Signals originated by rotating machinery, musical signals, alternating current ...
- *Harmonic analysis* or frequency estimation, aims at "'accurate"' determination of several frequency components

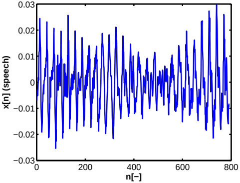


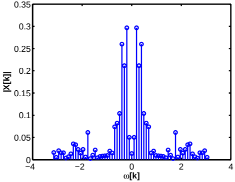


Practical spectral analysis II

2)

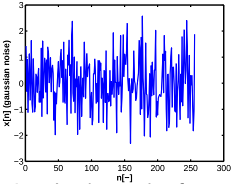
- (Locally) **(quasi)-periodic signals** (they have “certain” harmonic structure, spectrum is not sparse, several bands with significant energy)
- EXAMPLE: Speech signal (vowel - a,e,i,o,u)

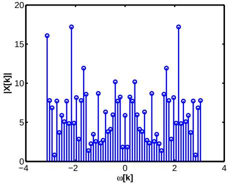




3)

- **Non-periodic signals** (no harmonic structure, energy spread throughout the spectrum, wide-band signals)
- EXAMPLE: White noise, speech signal (fricatives - s,z)





- *Spectral analysis* - Analyzes the frequency bands, the“shape” of the spectrum, the distribution of energy with respect to frequency (e.g., computation of features for speech recognition, detection of formants etc.).

Practical spectral analysis III

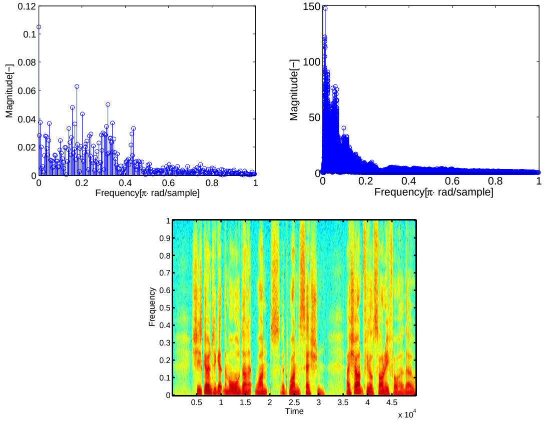
- spektralni analyza => zkoumame intervaly
- harmonicka analya => zkoumame cary magnitud

Factors complicating practical spectral analysis:

- Finite (short) signal length - prevents accurate detection of frequency components.
- (Potential) non-stationarity - spectrum of the signal evolves in time. Signal should not be analyzed as a whole over the changes. Instead, the signal is analyzed in short intervals, where it is approximately stationary - **short-time spectral analysis**. - reseni nestacionarity
- Presence of various unwanted noise components (quantization, sensor, environment noise).

Short-time spectral analysis

- Variant of spectral analysis for non-stationary signals (whose spectrum changes in time)
- In this case, the spectrum should not be computed using the whole signal:
  - The computed spectrum has extraordinary spectral resolution ($\Delta\omega = 2\pi/N$)...
  - ...but practically no time resolution
  - Computational burden is unnecessarily large
- More useful is a sequence of short DFTs, which provides a compromise between time and spectral resolution
- *Short Time Spectral Analysis + Windowing*
- Computation using *Short Time Fourier Transform - STFT*
- EXAMPLE: Analysis of music recording



(a) Short-time DFT spectrum ($N = 512$, $\Delta\omega \approx 3 \cdot 10^{-3}\pi$, $\Delta f \approx 40$Hz), (b) DFT spectrum ($N = 10^5$, $\Delta\omega = 2 \cdot 10^{-5}\pi$, $\Delta f = 0.2$Hz), (c) Spectrogram

- Time-resolution of spectrogram can be improved by *overlapping* of segments for DFT computation.

Spectral leakage - multiplication by rectangular window

- Let us have infinite signal $y[n]$, from which we select short data segment $x[n]$ such, that

$$x[n] = \begin{cases} y[n], & 0 \leq n \leq N-1 \\ 0, & \text{elsewhere} \end{cases} \tag{1}$$

- This operation corresponds to multiplication of $y[n]$ with rectangular window $w_r[n]$
- Multiplication in the time-domain corresponds to the convolution in the frequency domain
- The relationship between DTFT spectrum of long signal $y[n]$ and signal $x[n]$ weighted by the rectangular window is therefore

$$X(e^{j\omega}) = \frac{1}{2\pi}\{Y(e^{j\omega}) * W_r(e^{j\omega})\} \tag{2}$$

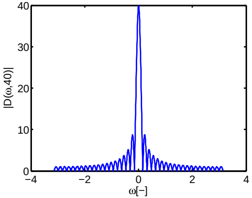
- Function $W_r(e^{j\omega})$ is DTFT$(w_r[n])$ given by

$$W_r(e^{j\omega}) = \frac{sin(0.5\omega N)}{sin(0.5\omega)}e^{-j0.5\omega(N-1)} = D(\omega, N)e^{-j0.5\omega(N-1)} \tag{3}$$

Multiplication by rectangular window II

Dirichlet kernel - Magnitude of DTFT$(w_r[n])$ - $D(\omega, N)$

- Maximum value N occurs at frequency $\omega = 0$
- Closest zeros occur at frequencies $\pm 2\pi/N$
  Frequency interval between zeros is denoted as *main lobe*
- Another zeros occur at frequencies
  $\omega = 2m\pi/N, m = \pm 2, \pm 3, \ldots$
  Frequency intervals between these zeros are denotes as *side lobes*
- A lobe with largest magnitude occurs at frequency $\omega = \pm 3\pi/N$ and the ratio between its magnitude and the magnitude of the main lobe is -13.5dB.



- lalokové spektrum

Multiplication by rectangular window III

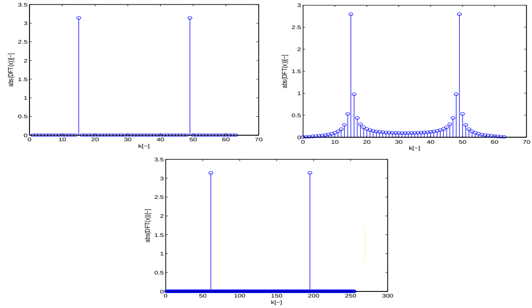
What is the difference between spectrum $X(e^{j\omega})$ of signal $x[n]$ (signal multiplied by rectangular window) from spectrum $Y(e^{j\omega})$ of the original signal $y[n]$?

- There appears **smearing of spectrum**
  - The loss of frequency resolution - if two frequency components in $Y(e^{j\omega})$ are distant less than $2\pi/N$ (width of the side-lobe), then they merge.
- There appears **masking of weak frequency components**
  - If there is one dominant component within the spectrum and some weak components, then the side-lobe of the dominant component masks the main-lobes of the weak ones.
  - This effect is the most significant, when the components differby and odd multiple of π/N

In other words: Selection of signal segments via rectangular window has undesirable side effects on the spectrum of the original signal and may significantly distort results of short-time spectral analysis in some cases.

Spectral leakage II

- Spectral leakage (smearing) occurs **always**, when the signal $y[n]$ is windowed and the (DTFT/DFT) spectrum is computed from shortened sequence $x[n]$
- EXAMPLE: How is it possible that if exactly one period of harmonic signal is selected then its DFT spectrum appears free of leakage?



$$x_1[n] = sin(\frac{2\pi 15}{64}n),\ N_1 = 64;\ x_2[n] = sin(\frac{2\pi 15{,}25}{64}n), N_1 = 64;$$
$$x_2[n] = sin(\frac{2\pi 15{,}25}{64}n),\ N_2 = 256;$$

Windowing

Undesired effects of windowing by rectangular window $w_r[n]$ can be partly mitigated via selection of a more suitable window $w[n]$

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

The desired sequence $w[n]$ is not arbitrary, it must fulfill the following criteria:

- Sequence $w[n]$ has final duration
- Window length N_w is the same as the length of the segment to be analyzed
- Sequence $w[n]$ should be non-negative

Moreover, following properties are of importance in the frequency domain:

- The main-lobe should have minimal width
- Side-lobes should have minimal magnitude

!

DTFT sequence $w[n]$ denoted as $W(e^{j\omega})$ is called *kernel function*

Windowing II

- **Ideal kernel function** $W(e^{j\omega})$**:** approaches $\delta(\omega)$, then convolution

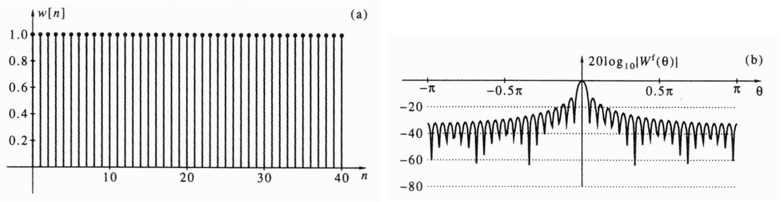
$$X(e^{j\omega}) = \frac{1}{2\pi}\{Y(e^{j\omega}) * W(e^{j\omega})\} \tag{4}$$

  does not smear spectrum $Y(e^{j\omega})$ too significantly.
- Unfortunately, the window corresponding to kernel function $W(e^{j\omega}) = 2\pi\delta(\omega)$ is $w[n] = 1$, i.e., it is infinite (no windowing occurs).

- **Selection of suitable window:** is a compromise between ...
- ... narrow main-lobe ...
- ... and side-lobes with low magnitude
- The narrower the main-lobe, the higher the magnitude of the side-lobes

Rectangular window

- **Rectangular window:** Has the narrowest *main-lobe* from all windows: $4\pi/N$
- *Magnitude of side-lobes* is however the largest: $-13.5dB$, which is highly impractical for spectral analysis - weak frequency components are masked



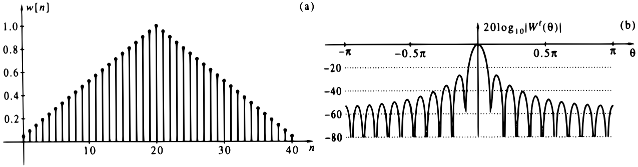
(a) Rectangular window $w_r[n]$, (b) DTFT spectrum $W_r(e^{j\omega})$ magnitude

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

- Derived from squaring of the kernel function $W_r(e^{j\omega})$
- Result in two-times lower side-lobes (in dB)
- Squaring $W_r(e^j\omega)$ in frequency-domain corresponds to $w_r[n] * w_r[n]$ in time-domain
  (length $w_r[n]$ is $(N+1)/2$ for $w_t[n]$ of length N)

$$w_t[n] = \frac{2}{N+1}\{w_r[n] * w_r[n]\} = 1 - \frac{|2n - N + 1|}{N+1} \qquad (5)$$

- *Main-lobe width*: $8\pi/(N+1)$
- *Side-lobe magnitude*: $-27dB$



(a) Triangular window $w_t[n]$, (b) DTFT spectrum $W_t(e^{j\omega})$ magnitude

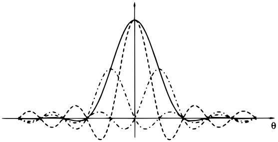
SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Hann window

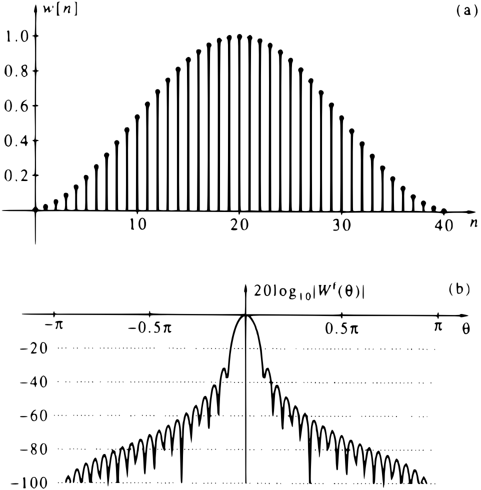
- Derived from superposition of three Dirichlet kenrels shifted in frequency ($\Delta\omega = \pm 2\pi/(N-1)$), which partly cancels its side-lobes
- Magnitude of central kernel is 0.5, magnitude of the two-shifted kernels are 0.25

$$w_{hn}[n] = 0.5\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right], \quad 0 \leq n \leq N-1 \qquad (6)$$

- By mistake denoted as Hanning
- *Main-lobe width*: $8\pi/(N)$
- *Side-lobe magnitude*: $-32dB$
- The boundary samples are equal to 0 (deletes samples $y[0]$ a $y[N-1]$)



SOURCE: BOAZ PORAT, A Course in Digital Signal Processing



(a) Hann window $w_{hn}[n]$, (b) DTFT spectrum $W_{hn}(e^{j\omega})$ magnitude

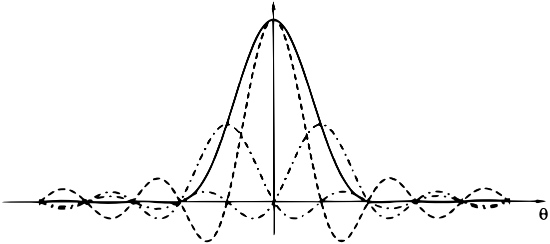
SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Hamming window

- **Hamming window** is obtained by modification of magnitudes of Dirichlet kernels summed to obtained the Hann window

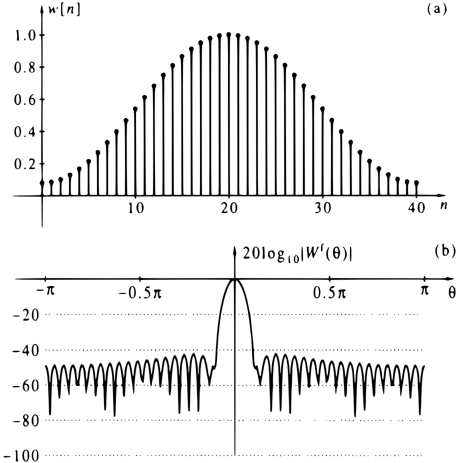
$$w_{hm}[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \tag{7}$$

- Largest side-lobe is not the closest to the main lobe
- *Main-lobe width*: $8\pi/(N)$
- *Side-lobe magnitude*: $-43dB$



SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Hamming window II



(a) Hamming window $w_{hm}[n]$, (b) DTFT spectrum $W_{hm}(e^{j\omega})$ magnitude

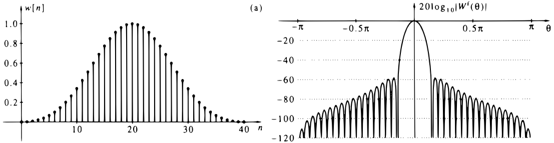
SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Blackmann window

- **Blackmann window** stems from superposition of five Dirichlet kernels shifted in frequency ($\Delta\omega = \pm 2\pi/(N-1)$)

$$w_b[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \qquad (8)$$

- *Main-lobe width*: $12\pi/(N)$
- *Side-lobe magnitude*: $-57dB$
- The boundary samples of Blackmann window are equal to 0 (deletes samples $y[0]$ a $y[N-1]$)



(a) Blackmann window $w_b[n]$, (b) DTFT spectrum $W_b(e^{j\omega})$ magnitude

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Kaiser window

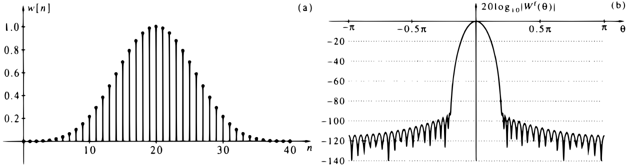
- Previous windows are considered classic - these are based on intuition and qualified guesses
- Kaiser window is an example of modern window, which is based on optimality criterion
- **Kaiser criterion:** Minimize the width of the main-lobe, provided that the length of the window is fixed and the energy of side-lobes does not exceed given percentage of total energy
- **Kaiser window** is given by

$$w_k[n] = \frac{I_0\left[\alpha\sqrt{1-\left(\frac{|2n-N+1|}{N-1}\right)^2}\right]}{I_0[\alpha]}, \quad 0 \leq n \leq N-1 \qquad (9)$$

where $I_0(x) = \sum_{k=0}^{\infty}\left(\frac{x^k}{2^k k!}\right)^2$ is modified Bessel function of order 0, $\alpha \in R$ - a free parameter influencing the main-lobe/side-lobe compromise

Kaiser window II

- **Parameter** α of the Kaiser window influences the width of the main-lobe and magnitude of the side-lobes
- For growing α, the main-lobe width is growing and the magnitude of the side-lobe diminishes
- *Example of Kaiser window:* $N = 41, \alpha = 12$
- *Main-lobe width*: $16\pi/(N)$
- *Side-lobe magnitude*: $-90dB$



(a) Kaiser window $w_k[n]$, (b) DTFT spectrum $W_k(e^{j\omega})$ magnitude

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Part II

Harmonic analysis

Measuring frequency of periodic signals

- Measuring frequency of periodic signals, especially harmonic ones, is a very important taks in digital signal processing
- Fourier analysis is a natural tool for this task
- In practice, signals are measured only within some finite time interval
- Spectrum of such signals can then be evaluated on some discrete finite set of frequencies (using DFT)

Measuring frequency for a set of harmonic functions

We seek argument of maximum of magnitude spectrum computed via:

1. We select a window, which reflects (by side-lobe magnitude) the expected ratio of the weakest and the largest frequency components
2. We multiply signal $y[n]$ by a window of selected length - the length is selected according to the stationarity of the analyzed signal as a compromise between frequency and time resolution
3. We compute $Y(e^{j\omega})$, in practice its sampled variant (DFT spectrum) $Y[k]$ (using FFT)
4. If the window is suitable and the *conditions of distinguish-ability* hold, the the sought frequencies are *close* to the local maxima of $Y[k]$.
   - Conditions of distinguish-ability determine, when two frequency components in a short signal can be distinguished from each other

   The inaccuracy of the detected maximum is caused by
   - limited frequency resolution (mitigated by concatenation of zeros)
   - frequency bias - summation of side lobes of various components; it shifts the local maximum of the magnitude spectrum

Suitable a priory information in this task is:

1. Frequency distance between the frequency components
2. Ratio of magnitudes of the respective frequency components
3. Distance of the frequencies ω_k from 0 and π

Measuring frequency for a single complex exponential

- Let us have continuous signal $y(t) = Ae^{j(\Omega t+\phi_0)}$ and let us measure the frequency Ω_0
- Let us sample the signal with sample period T_s such that $-\pi < \Omega_0 T_s < \pi$
- We obtain a signal $y[n] = Ae^{j(\omega_0 n+\phi_0)}, \quad 0 \leq n \leq N-1$ and $\Omega_0 T_s = \omega_0$
- Dirichlet kernel has a single maximum in point $\omega = 0$, therefore it is in theory possible to find ω_0 exactly as a frequency, where magnitude spectrum $|Y(e^{j\omega})|$ is maximal
- CAREFUL - in practice it is not possible to find the global maximum exactly, we evaluate $|Y(e^{j\omega})|$ only for some finite number of points using DFT
- If it is necessary, it is possible (for enhancement of the frequency resolution) to concatenate the original sequence $y[n]$ with a vector of zeros

Measuring frequency for a two complex exponentials

- Let us have continuous signal given by $y(t) = A_1 e^{j(\Omega_1 t + \phi_1)} + A_2 e^{j(\Omega_2 t + \phi_2)}$ with the task to measure Ω_1, Ω_2
- Let us sample the signal with sampling period T_s such that $-\pi < \Omega_{1,2} T_s < \pi$
- We obtain signal $(\Omega_1 T_s = \omega_1, \Omega_2 T_s = \omega_2)$ $y[n] = A_1 e^{j(\omega_1 n + \phi_1)} + A_2 e^{j(\omega_2 n + \phi_2)}, \quad 0 \leq n \leq N-1$
- Let us search first for ω_1, for $Y(e^{j\omega})$ in point $\omega = \omega_1$ holds

$$Y(e^{j\omega})|_{\omega=\omega_1} = NA_1 e^{j\phi_1} + A_2 e^{-j(0.5(\omega_1-\omega_2)(N-1)-\phi_2)} D(\omega_1 - \omega_2, N)$$

- If $A_2 \neq 0$ and

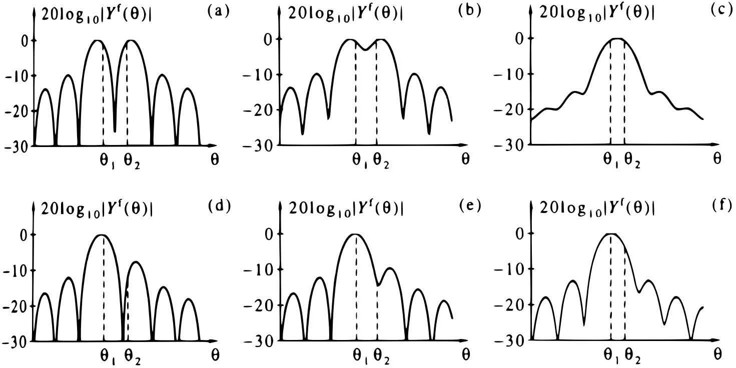
$$|A_2 D(\omega_1 - \omega_2, N)| << NA_1 \tag{10}$$

  than the *local* maximum $Y(e^{j\omega})$ approaching ω_1 is well distinguishable.
- The condition (10) holds, if $|\omega_2 - \omega_1| \geq 2\pi/N$ and if A_2 "not much larger" than A_1

Similar conditions hold symmetrically for ω_2

Measuring frequency for a two complex exponentials II

EXAMPLE: Measuring frequency for a two complex exponentials:
(a,b,c): $A_1 = A_2$, (d,e,f): $A_2 = 0.25A_1$



(a) $\omega_1 - \omega_2 = 2\pi/N$, (b) $\omega_1 - \omega_2 = 1.5\pi/N$, (c) $\omega_1 - \omega_2 = \pi/N$
(d) $\omega_1 - \omega_2 = 2\pi/N$, (e) $\omega_1 - \omega_2 = 1.5\pi/N$, (f) $\omega_1 - \omega_2 = \pi/N$

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Measuring frequency for a two complex exponentials III

- Condition (10) can be hard to fulfill using rectangular window (large magnitude of side-lobes)
- This problem can be partly mitigated using windowing
- For the value of spectrum $Y(e^{j\omega})$ on frequency ω_1 using window $w[n]$ of length N (with DTFT denoted as $W(e^{j\omega})$) it holds

$$Y(e^{j\omega})|_{\omega=\omega_1} = A_1 e^{j\phi_1} W(e^{j0}) + A_2 e^{j\phi_2} W(e^{j(\omega_1-\omega_2)})$$

- The condition (10) using window $w[n]$ evolves into

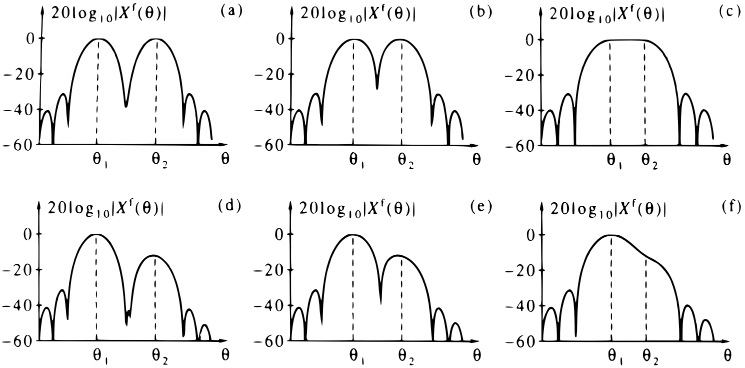
$$|A_2 W(e^{j(\omega_1-\omega_2)})| << A_1 \sum_{n=0}^{N-1} w[n] \qquad (11)$$

where $W(e^{j0}) = \sum_{n=0}^{N-1} w[n]$

- Condition (11) holds:
  1. if $|\omega_1 - \omega_2|$ is larger than $\frac{1}{2}$ of the main-lobe of $W(e^{j\omega})$
  2. if $20log_{10}(A1/A2)$ larger than the magnitude of the side-lobes

EXAMPLE: Measuring frequency for a two complex exponentials: (windowing by Hann window)

(a,b,c): $A_1 = A_2$, (d,e,f): $A_2 = 0.25A_1$



(a) $\omega_1 - \omega_2 = 8\pi/N$, (b) $\omega_1 - \omega_2 = 6\pi/N$, (c) $\omega_1 - \omega_2 = 4\pi/N$

(d) $\omega_1 - \omega_2 = 8\pi/N$, (e) $\omega_1 - \omega_2 = 6\pi/N$, (f) $\omega_1 - \omega_2 = 4\pi/N$

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Measuring frequency for a set of harmonic functions

TASK: We want to learn frequencies of M real-valued harmonic functions

- Real-valued harmonic functions exhibit both-sided symmetric spectrum

For well distinguishable components it must hold:

1. All components $\omega_k, k = 1 \dots M$ must be distant in the spectrum at least $2\pi/N$
2. No component ω_k is lower than π/N and larger than $\pi(1 - 1/N)$
3. All amplitudes $A_k, k \neq m, k = 1 \dots M$ are lower or "not much larger" than A_m

Again: side-lobes of the Dirichlet kernel will mask weak frequency components - to mitigate, windowing can be utilized

Measuring frequency for a set of harmonic functions II

Usage of window $w[n]$ changes the conditions of distinguish-ability for two frequency components as follows:

1. All components $\omega_k, k = 1 \ldots M$ are mutually distant at least half of the main-lobe $W(e^{j\omega})$
2. No-frequency component ω_k is lower than half of the main-lobe $W(e^{j\omega})$ and larger than π minus half of main-lobe $W(e^{j\omega})$
3. The ratio of logarithmic magnitudes $20 \log_{10} A_k$ cannot be larger than the magnite of the side-lobe of $W(e^{j\omega})$

Influence of noise on harmonic analysis

- Some sort of noise is up to some extend present in all measured signals
- Harmonic analysis in the noisy case is performed as in the noiseless case, up to following differences:
- The noise further **masks weak frequency components** (alongside masking due to window side-lobes)
- *Signal detection:* Distinguishing of the weak harmonic components in the presence of many spurious noise peaks
- The noise **shifts maxims** of the DTFT/DFt spectrum
- *Frequency estimation:* Found maxims are identified with an error corresponding to the random nature of the noise

Influence of noise on harmonic analysis II

- The influence of noise on signal detection and frequency estimation can be quantified for white noise to some extend
- The analysis is accurate only for signals containing only "a few" harmonic components
- For approximative quantification the following "'rule-of-thumb"' is used
- Frequency component can be detected in the presence of noise, if:
  1. The *conditions of distinguishability* (see slide 32)
  2. The following *enequality* holds

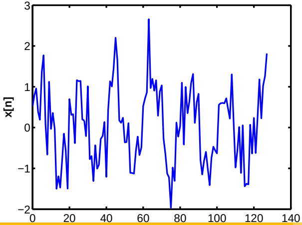
$$\frac{N \cdot A^2 \cdot \mathrm{PG}}{P_v} \geq 100 \tag{12}$$

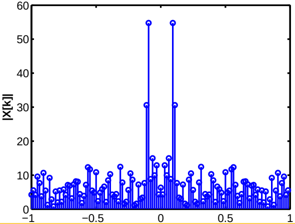
  - N - Window length
  - A - Harmonig component magnitude
  - PG - *Processing gain* - Parameter characteristic for a specific window, determines amplification of harmonic signal with respect to noise during windowing (the higher the better)
  - P_v - Energy of white noise (zero value for $\omega = 0$)

Influence of noise on harmonic analysis III

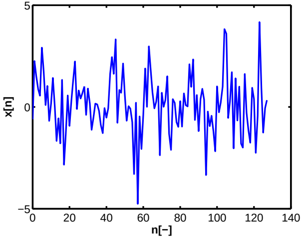
Window	Recangular	Bartlett	Hann	Hamming	Blackman	Kaiser($\alpha = 12$)
PG	1	0.74	0.67	0.73	0.58	0.50

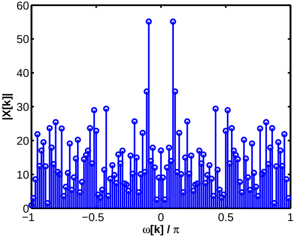
- DETAILS: Boaz Porat, *A course in digital signal processing*, 185 / chapter 6.5
- EXAMPLE: Detection of harmonic signal in the presence of noise
- *Signal*: $x[n] = \sin(0.1\pi \cdot n) + v[n]$, $v[n]$ - white noise
- P_{lim}: Limit to noise energy, which allows to detect the harmonic component as in (12)
- **Scenario 1**: Noise energy $P_v = 0.25 P_{lim}$



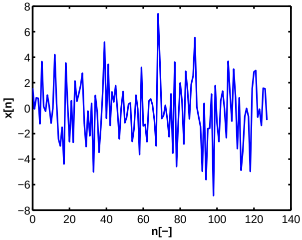


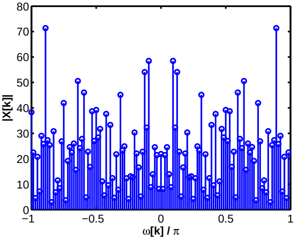
- **Scenario 2**: Noise energy $P_v = P_{lim}$





- **Scenario 3**: Noise energy $P_v = 4P_{lim}$





Part I

Application of Digital Signal Processing

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)

Too many areas to cover in a single lecture. We will focus on:

- Processing of musical signals
- Processing of biomedical signals (ECG)
- Speech enhancement - denoising
    - Spectral thresholding
    - Spectral subtraction

Part II

Processing of musical signals

- Audio signals generated by musical instruments have rather simple mathematical model

$$x(t) = a(t) \sum_{m=1}^{\infty} c_m \cos(2\pi m f_0 t + \phi_m) \quad (1)$$

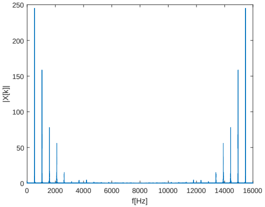
where f_0 is a pitch, c_m is an amplitude (loudness) and ϕ_m is a phase of the mth harmonic component. $a(t)$ is the signal "envelope" (time dynamics); low-frequency signal modulating the sound amplitude.

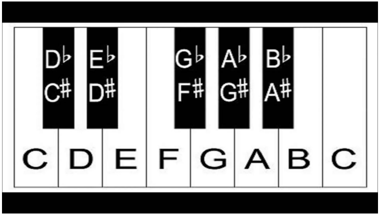
- DETAILS: Basic musical theory
- **Example:** The same tone played on various instruments

Violoncello, *Classical guitar*
Flute, *French horn*

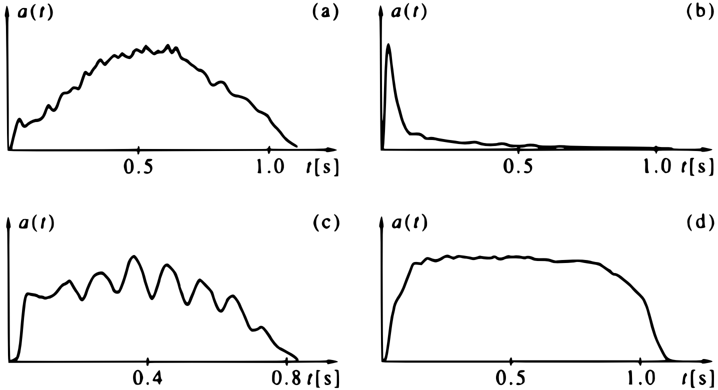
Basic musical theory

- European music has twelve halftones in an octave (c-c'), which can be well imagined through piano keys.
- Each tone has a pitch given by its "fundamental frequency" and higher harmonics which determine a color/timbre of the tone (this differs tones played on various instruments).
- If the tone has fundamental frequency f_0 (e.g., a' has frequency 440 Hz), then a tone higher by one halftone has frequency $f_0 \cdot 2^{1/12}$ (that is a$\#$' has frequency 466 Hz).
- Tones which differ by an octave have double frequency (geometric series with ratio $2^{1/12}$)



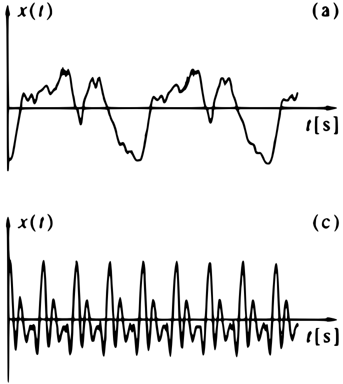


Left: DFT magnitude spectrum of tone c" on flute (basic frequency approximately 523Hz)
Right: Piano keys (SOURCE: `piano-keyboard-guide.com`)



Tone Dynamics: (a) Violoncello, (b) Classical guitar, (c) Flute, (d) French horn

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing



Time course: (a) Violoncello (220 Hz), (b) Classical guitar (440 Hz), (c) Flute (880 Hz), (d) French horn (440 Hz)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Additive synthesis:

- The very basic synthesis uses the mathematical model in (1)
- To create a tone of a specific instrument, the knowledge of its time dynamics and harmonic structure is needed
- Advantage: Tone modification can be easily achieved by a change of several model parameters
- Disadvantage: Imperfect realism, the model (1) does not take all sound phenomena into consideration

Examples:

Generated melody - French horn, *Recorded melody - French horn*, *Generated melody - Trumpet*

Sample-based synthesis

- Basic elements are recorded samples of true musical instruments
- Acords are created by summation of partial samples
- Advantage: The generated tones are very realistic
- Disadvantage: Realistic tone modification requires "complicated" DSP techniques
- Disadvantage: Memory requirements

Examples:

Violoncello, *French horn*,
Violoncello + *French horn*

Sound effects using DSP:

- DSP is used to generate *sound effects* - artificially created or modified sounds, which are used in the movies, electronic music, computer games and life performances
- To create the sound effects, one can use linear (adaptive) filtering or various nonlinear (irreversible) mothods
- More about creation the creation of sound effects can be learned in subject **Digital Audio Engineering** (DAI, in Czech), which is tutored by *Prof. Ing. Zbyněk Koldovský, PhD.*

Example: Digital sound effects applied to the sound of electric guitar

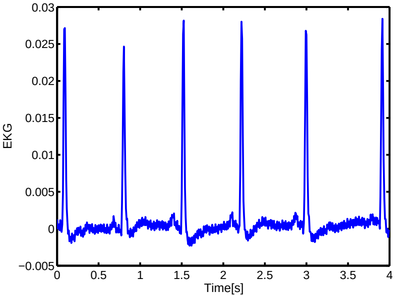
Original sound	*Vibrato*
Delay	*Chorus*
Tremolo	*Wah-wah*

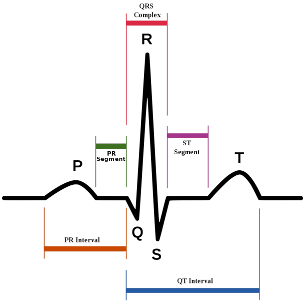
Part III

Processing of biomedical signals

Processing of ECG

- A lot of different biomedical signals exist, their analysis is differs for each specific signal
- **Electrocardiogram** is a electrical signal reflecting the heart activity
- ECG of a healthy heart has a characteristic shape - QRS complex
- The time-domain analysis of the (shape of) QRS complex allows the physicians to state diagnosis
- The sampling frequency of ECG is usually $Fs = 500Hz$ and more
- ECG can contain many unwanted signals called *artifacts*
- *Narrow-band noise:* isoline drift (slow pacient movements, breathing), voltage
- *Wide-band noise:* Myopotencials (muscle movements), sudden isoline changes (insuficient electrode contact)
- Artifact removal by fitration is performed only in the case, when the signal cannot be remeasured correctly
- **Fidelity criteria** do not allow linear filtering of the voltage noise in the strictest case
- Often, the filtering can be performed by (adaptive) FIR filters with linear phase. If the IIR filter is applied, its nonlinear phase needs to be compensated.





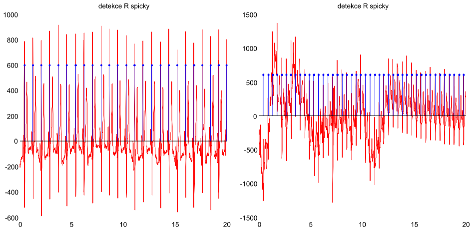
ECG and its stages

SOURCE (right image): en.wikipedia.org

Example: Detection of the R pitch within the ECG signal

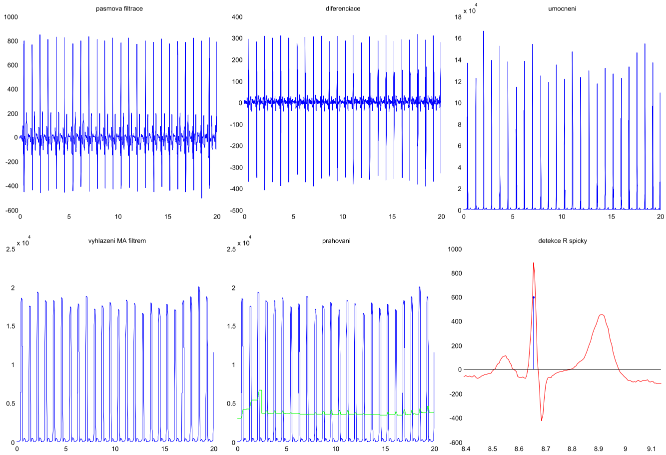
1. Bandpass filtering, pass-band 2-30Hz
2. Differentiation: $y[n] = x[n] - x[n-1]$,
   $y[n] = 0.5(x[n] - x[n-2])$
3. Squaring - emphasizes large and attenuates small values
4. Moving average filtering/smoothing
5. Thresholding
6. Detection of R pitches

More about biomedical signal processing can be learned in subject **Modern methods of signal processing** (MMZ, in Czech), which is lectured by *Prof. Ing. Zbyněk Koldovský, PhD.*



SOURCE: doc. Ing. ROMAN ČMEJLA CSc., Biomedical Signals, lectures

Example: Detection of the R pitch within the ECG signal



SOURCE: doc. Ing. ROMAN ČMEJLA CSc., Biomedical Signals, lectures

Part IV

Speech Enhancement

Speech Enhancement

Broad set of algorithms, which aims to increase speech quality and to remove various distortions within the speech signal.

- **Quality:** clarity, subjective intelligibility, compatibility with a subsequent signal processing technique
- **Distortion:** Aditive noise, reverberation, nonlinear distortions (clipping), influence of codecs/recording device/sensors
- **Evaluation:** Objective (computable criteria, recognition accuracy), subjective (listening tests)

Speech denoising:

- Identification and removal (suppression) of non-desired components within speech.
- Which component is non-desired (i.e., noise) depends on the situation.
- According to available information about noise signals, a denoising algorithm is chosen.

Assumption:

- *Stationary, always active:* Spectral subtraction
- *Directional, unknown direction:* Blind Source Separation
- *Directional, known direction:* Target cancellation filters, beamforming
- *Extensive sample database:* Machine learning principles

Part V

Spectral thresholding, Spectral subtraction

Denoising by spectral thresholding I

- Simple method for removal of additive background noise, provided that the SNR is high enough for the most frequency components of speech
- **Assumption:** Noise $v[n]$ has lower instantaneous power than speech $s[n]$, i.e., frequency bins with the lowest power corresponds to noise
- Based on STFT and additive model of the noisy signal ($x[n] = s[n] + v[n]$)
- **Disadvantage:** The assumption hold to a limited extend for a real-world signal, the method removes also the weak components of speech
- One of methods called jointly Spectral Masking

Algorithm:

1. Computation of the Short-time Fourier Transform (STFT): $X[m,k] = \mathrm{STFT}(x[n])$ (m-frame index, k-frequency bin index)
2. Thresholding/masking: $\hat{S}[m,k] = W[m,k] \cdot X[m,k]$, where $W[m,k]$ is a spectral mask (here a binary thresholding function)

$$W[m,k] = \begin{cases} 0, & |X[m,k]| \leq T \\ 1, & |X[m,k]| > T \end{cases} \tag{2}$$

   and T is a suitable threshold.
3. Reconstruction using an inverse STFT: $\hat{s}[n] = \mathrm{ISTFT}(\hat{S}[m,k])$.

Perfect signal reconstruction using ISTFT I

- Processing of signal $s[n]$ via STFT: (Short Time Fourier Transform)
  1. Signal segmentation into frames of length M weighted by a window $w[n]$, i.e.,

     $$s_m[n] = s[n] \cdot w[n - mR], \tag{3}$$

     where R is a shift of the time frames.
  2. Computation of DFT for each segment $S[m, k] = DFT(s_m[n])$.
- It is possible to reconstruct $s[n]$ perfectly from its STFT image $S[m, k]$, provided that the *shape* and the *shift* of $w[n]$ is suitable.
- This is possible to inversibility of DFT, if $s[n]$ can be reconstructed from weighted frames $s_m[n]$.
- **Condition:** (for the shape of $w[n]$)

$$\begin{aligned} s[n] &= \sum_{m=-\infty}^{\infty} s_m[n] = \sum_{m=-\infty}^{\infty} s[n]w[n - mR] \\ &= s[n] \sum_{m=-\infty}^{\infty} w[n - mR]. \end{aligned} \tag{4}$$

Perfect signal reconstruction using ISTFT II

- Using equation (4) we derive the condition

$$\sum_{m=-\infty}^{\infty} w[n - mR] = \text{const.}, \qquad (5)$$

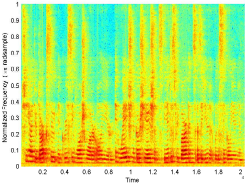
i.e., the sum of overlapping shifted windows must be equal to a constant.

- Reconstructed signal is then similar to the original, up to a scaling.
- The conventional symmetric windows must be slightly modified to fulfill (5), .e.g, last sample must be set to zero.
- MATLAB: Windows for perfect reconstruction can be obtained using parameter `'periodic'`, e.g., `w = hamming(512,'periodic')`
- For different windows is the allowed shift R in (5) different:

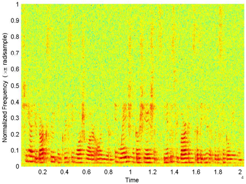
Window	Shift $R \in \mathcal{Z}$
Rectangle	M/p
Bartlett	M/p
Hann	$(M/2)/p$
Hamming	$(M/2)/p$
Blackmann	1

($p \in \mathcal{Z}$ - a free parameter)

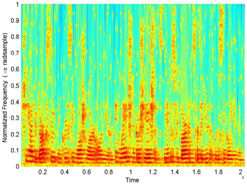
Example: Removal of Gaussian Noise



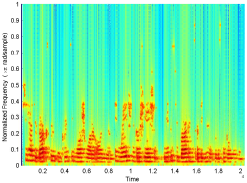
(a)



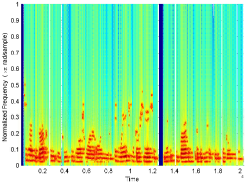
(b)



(c)



(d)



(e)

(a) Clean Speech, (b) Mixture with gaussian noise (SNR=10dB), (c) Enhanced ($T = 0.008 \cdot max(|X[m,k]|)$) *(d) Enhanced* ($T = 0.02 \cdot max(|X[m,k]|)$) *(e) Enhanced* ($T = 0.035 \cdot max(|X[m,k]|)$)

Denoising by spectral subtraction I

- A method for removal of stationary additive noise
- **Assumption:** Noise $v[n]$ has stationary spectrum $V[k]$ and is active separately during some time instant (i.e., target speech $s[n]$ is not active, which needs to be detected - Voice activity detection)
- Based on STFT and additive model of the noisy speech ($x[n] = s[n] + v[n]$)
- **Disadvantage:** Changes in noise spectrum need to be detected or the spectrum must be periodically updated (i.e., $V[m, k]$ is time-variant)

Algorithm:

0. Estimation of magnitude spectrum $V[k]$ (smoothing of $|\mathrm{STFT}(v[n]))$ (estimation in interval, where $x[n] = v[n]$, i.e., $s[n] = 0$).
1. Computation of STFT: $X[m, k] = \mathrm{STFT}(x[n])$
2. Computation of magnitude spectrum $|X[m, k]|$ and phase $\phi[m, k]$:

$$\phi[m, k] = \tan^{-1} \frac{\mathrm{Im}(X[m, k])}{\mathrm{Re}(X[m, k])} \tag{6}$$

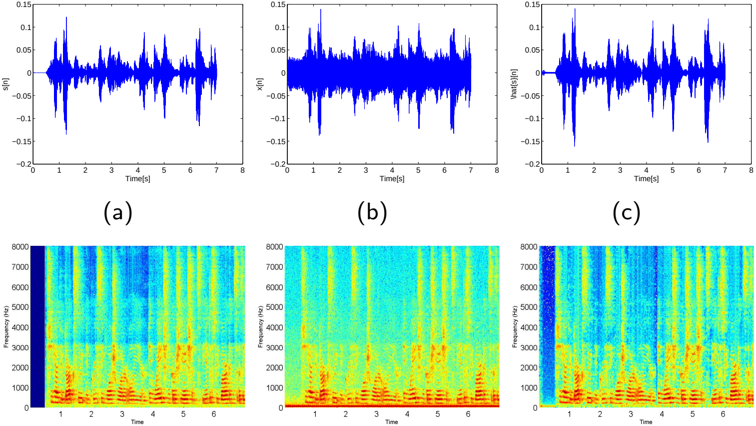
3. Estimation of speech magnitude spectrum $|\hat{S}[m, k]|$:

$$|\hat{S}[m, k]| = \begin{cases} |X[m, k]| - |V[k]|, & |X[m, k]| - |V[k]| > 0 \\ 0, & else \end{cases} \tag{7}$$

4. Reconstruction $\hat{s}[n] = \mathrm{ISTFT}(|\hat{S}[m, k]| \cdot e^{j\phi[m,k]})$

Denoising by spectral subtraction II

Example: Fan noise subtraction



(a) Clean speech,
(b) Noisy signal with a fan noise (SNR=5dB),
(c) After subtraction

Part I

Z-transform

Z-transform - motivation

- The utilization of DTFT and DFT is very useful for analysis of signals (spectral analysis)
- However, it cannot analyze easily some important properties of systems/filters
  - Stability cannot be simply determined
  - Causality cannot be easily work with (important for realizable filters, inverse systems computation)
- For analysis of these properties, a generalization of DTFT denoted by **Z-transform** is used

Z-transform I

- A transform generalizing DTFT for signals, which are not absolutely summable
- Applications in signal processing:
  1. LTI system analysis (stability, causality)
  2. Filter design
  3. Solving of difference equations with initial conditions
- DTFT is defined by:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega} \tag{1}$$

for sequences $x[n]$, which are absolutely summable, i.e.,

$$\sum_{n=-\infty}^{\infty} |x[n]| = S < \infty \tag{2}$$

- For many common sequences, the DTFT does not exist/converge, e.g.,

$$x[n] = u[n] \tag{3}$$

Z-transform II

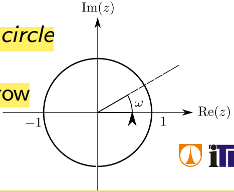
- **Z-transform** of discrete sequence $x[n]$ is defined by:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \tag{4}$$

$z = re^{j\omega}, z \in \mathcal{C}, r \in \mathcal{R},\ r > 0$

- The z-transform output is a complex function of a complex variable. Its properties are commonly described in *the z-plane*.
- The transform is denoted by $x[n] \overset{Z}{\Longleftrightarrow} X(z)$
  $X(z)$ - Output of the z-transform applied to $x[n]$
- QUESTION: What is the relation of Z-transform and DTFT?

- **Unit circle:**
- DTFT output can be obtained from the Z-transform output by substitution $z = e^{j\omega}$
- DTFT output thus consists of point located on *the unit circle* in the z-plane
- Point $z = 1$ corresponds to $\omega = 0$ and the frequencies grow counter-clockwise (e.g., $z = j$ corresponds to $\omega = \pi/2$)



Region of Convergence - ROC:

- the area in the z-plane, where the z-transform converges (the series (4) has a finite sum)
- EXAMPLE: Computation of z-transform and the ROC
- *Beware:* Z-transform output is uniquely determined, only when the ROC is known
- *ROC properties:*
  - Region of convergence is an annulus in the form $\alpha < |z| < \beta$
  - When $x[n]$ has *finite duration* the ROC consists of the whole z-plane except $z = 0$ and $z = \infty$.
    $z = \infty$ belongs to the ROC if $x[n]$ is right-sided
    $z = 0$ belongs to the ROC if $x[n]$ is left-sided
  - *Right-sided sequence:* ROC in the form $|z| > \alpha$
  - *Left-sided sequence:* ROC in the form $|z| < \beta$

Z-transform IV

- Z-transform output is a complex function of a complex independent variable
- It is visualized as a magnitude and a phase part (both have 3D graph - a real function of a complex variable)

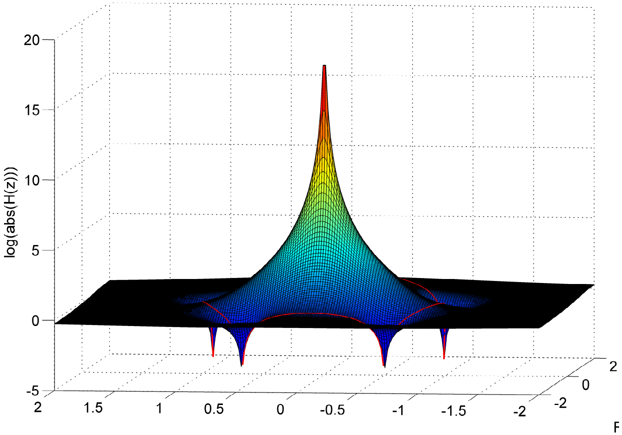
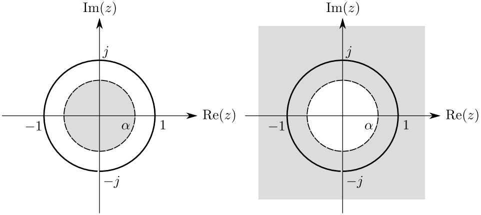


Table of common z-transform pairs



Sequence	Z-transform	ROC
$\delta[n]$	1	z-plane
$\alpha^n u[n]$	$\frac{1}{1-\alpha z^{-1}}$	$\|z\| > \alpha$
$-\alpha^n u[-n-1]$	$\frac{1}{1-\alpha z^{-1}}$	$\|z\| < \alpha$
$n\alpha^n u[n]$	$\frac{\alpha z^{-1}}{(1-\alpha z^{-1})^2}$	$\|z\| > \alpha$
$-n\alpha^n u[-n-1]$	$\frac{\alpha z^{-1}}{(1-\alpha z^{-1})^2}$	$\|z\| < \alpha$
$\cos[n\omega_0]u[n]$	$\frac{1-(\cos(\omega_0)z^{-1})}{1-2(\cos(\omega_0)z^{-1})+z^{-2}}$	$\|z\| > 1$
$\sin[n\omega_0]u[n]$	$\frac{\sin(\omega_0)z^{-1}}{1-2(\cos(\omega_0)z^{-1})+z^{-2}}$	$\|z\| > 1$

Z-transform properties I

- **Linearity:**
- Let $X_1(z)$ with ROC R_1 be z-transform output of $x_1[n]$ and $X_2(z)$ with ROC R_2 be z-transform output of $x_2[n]$, then:
- $y[n] = ax_1[n] + bx_2[n] \overset{Z}{\Longleftrightarrow} Y(z) = aX_1(z) + bX_2(z)$
- R_z contains intersection $R_1 \cap R_2$, but can be even larger
- EXAMPLE: ROC of a sum of two sequences

- **Shifting property:**
- Shifting of a sequence by n_0 samples leads to
- $x[n - n_0] \overset{Z}{\Longleftrightarrow} z^{-n_0} X(z)$
- ROC remains unchanged
  (with the exception of points 0 or ∞)

Z-transform properties II

- **Time-reversal:** of sequence $x[n]$ leads to
- $x[-n] \overset{Z}{\Longleftrightarrow} X(z^{-1})$
- When the original ROC is $R_x = (\alpha < |z| < \beta)$, then ROC corresponding to the shifted sequence is $1/R_x$, i.e., $(1/\beta < |z| < 1/\alpha)$

- **Multiplication by an exponential:** leads to scaling of the z-plane
- $\gamma^n x[n] \overset{Z}{\Longleftrightarrow} X(\gamma^{-1} z)$
- The time-domain multiplication scales also the ROC
- Let ROC of the original sequence be $R_x = (\alpha < |z| < \beta)$, then ROC of the multiplied sequence R_y takes the form $(|\gamma| \, \alpha < |z| < |\gamma| \, \beta)$

Z-transform properties III

- **Convolutional theorem:**
- Important property, convolution of two signals in the time-domain maps as multiplication of z-transform outputs in the z-domain
- $y[n] = h[n] * x[n] \overset{Z}{\Longleftrightarrow} Y(z) = H(z)X(z)$
- R_z contains intersection $R_h \cap R_x$, but can be even larger
- EXAMPLE: Convolution of two sequences using z-transform

- **Initial value theorem:**
- If $x[n]$ is a right-sided sequence then $x[0]$ can be found using:
- $x[0] = \lim_{z\to\infty} X(z)$
- QUESTION: Why is it so?

Part II

Description of the LTI systems using z-transform

Transfer (system) function

- Response of the LTI system given by *impulse response* $h[n]$ to the input $x[n]$ is given by convolution

$$y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \qquad (5)$$

- Application od DTFT to (5) gives the relation of spectra of the input and output signals via the *frequency response*

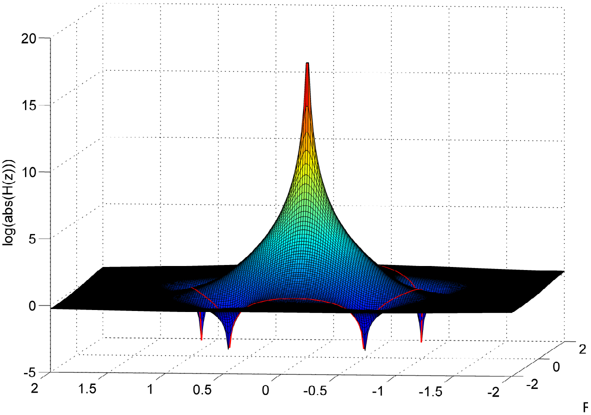
$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \qquad (6)$$

- Similarly, the application of z-transform to (5) gives the relation between z-transform outputs of the input/output signals via a *transfer function* $H(z)$

$$Y(z) = H(z)X(z) \qquad (7)$$

- Transfer function is very important from the perspective of LTI system analysis (stability, causality, inverse systems etc.)

- **Transfer function** is a complex-valued function of a complex independent variable
- It is visualized as a magnitude part and a phase part (both by a 3D graph - a real-valued function of a complex independent variable)



Transfer (system) function III

- **Transfer function** is thus given as a z-transform if the *impulse response* of the LTI system

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} \tag{8}$$

- *Frequency response* is obtained by evaluation of $H(z)$ on the unit circle

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} \tag{9}$$

- For the LTI system described via a recursive *difference equation* (i.e., an IIR system)

$$y[n] + \sum_{k=1}^{p} a[k]y[n-k] = \sum_{k=0}^{q} b[k]x[n-k] \tag{10}$$

the transfer function is a rational function in the (normalized) form

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{1 + \sum_{k=1}^{p} a[k]z^{-k}}. \tag{11}$$

Transfer (system) function IV

- **Zeros and poles:**
- LTI system with an infinite impulse response can be described using the transfer function in the form

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{\sum_{k=0}^{p} a[k]z^{-k}} \tag{12}$$

- *Alternatively:* The numerator and the denominator can be factorized into the product form

$$H(z) = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})} \tag{13}$$

- Roots of the numerator - *zeros* (β_k), roots of the denominator - *poles* (α_k)

- **Pole-zero plot** - a diagram depicting the location of all zeros (o) and poles (x) in the z-plane
- MATLAB: `zplane(B,A)`
- **Region of converge** is also often shown in the pole-zero plot
- For a rational $H(z)$ the ROC does not contain any poles

Transfer (system) function V

- In the z-domain, the LTI system is uniquely given (up to a gain C) by its zeros and poles
- The term $(1 - \beta_k z^{-1}) = \frac{z-\beta_k}{z}$ contributes to the transfer function the zero β_k and the pole in 0
- The term $\frac{1}{(1-\alpha_k z^{-1})} = \frac{z}{z-\alpha_k}$ contributes to the transfer function the pole α_k and the zero in 0
- Zeros and poles located at the same point cancel each other
- In the case of a real-valued impulse response, the complex zeros/poles form conjugate pairs (e.g., $\beta_1 = z_0$ and $\beta_2 = z_0^*$)

Part III

Inverse z-transform

- Inverse z-transform is used to recover a sequence $x[n]$ from its z-transform $X(z)$
- It is essential for many z-transform related tasks, such as difference equation solving, analytic computation of the convolution etc.
- There are three possible approaches:
  1. Partial fraction expansion
  2. Conversion to power series (polynomial long division)
  3. (Contour integration via Cauchy's integral theorem)

Partial fraction expansion

- Utilized for z-transforms given in the form of rational function

$$X(z) = C \frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})} \tag{14}$$

- If $p > q$ and all poles are simple $(\alpha_i \neq \alpha_k \text{for } i \neq k)$

$$X(z) = \sum_{k=1}^{p} \frac{A_k}{1-\alpha_k z^{-1}} \tag{15}$$

- $A_k \in \mathcal{R}$ are constants computed via

$$A_k = [(1-\alpha_k z^{-1})X(z)]_{z=\alpha_k} \tag{16}$$

- If $p > q$, then long polynomial division of numerator and denominator is performed
- DETAILS: How to proceed, when poles are of a higher order?
- EXAMPLE: Inverse z-transform of a rational $X(z)$
- MATLAB: `residuez(B,A)`

Part I

Inverse Z-transform (continued)

- Inverse z-transform is used to recover a sequence $x[n]$ from its z-transform $X(z)$
- It is essential for many z-transform related tasks, such as difference equation solving, analytic computation of the convolution etc.
- There are three possible approaches:
    1. Partial fraction expansion
    2. Conversion to power series (polynomial long division)
    3. (Contour integration via Cauchy's integral theorem)

Partial fraction expansion (repetition)

- Utilized for z-transforms given in the form of rational function

$$X(z) = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})} \tag{1}$$

- If $p > q$ and all poles are simple ($\alpha_i \neq \alpha_k$ for $i \neq k$)

$$X(z) = \sum_{k=1}^{p} \frac{A_k}{1-\alpha_k z^{-1}} \tag{2}$$

- $A_k \in \mathcal{R}$ are constants computed via

$$A_k = [(1-\alpha_k z^{-1})X(z)]_{z=\alpha_k} \tag{3}$$

- If $p > q$, then long polynomial division of numerator and denominator is performed
- DETAILS: How to proceed, when poles are of a higher order?
- EXAMPLE: Inverse z-transform of a rational $X(z)$
- MATLAB: `residuez(B,A)`

Polynomial long division

- Z-transform is defined by a power series

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = \cdots + x[-1]z^{1} + x[0] + x[1]z^{-1} + \ldots \tag{4}$$

- If the series is *finite*, the samples of $x[n]$ are simply selected
- If the series is *infinite*, it is usually given by a rational function

$$X(z) = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{1 + \sum_{k=1}^{p} a[k]z^{-k}} \tag{5}$$

- By polynomial division of numerator and denominator, the samples of $x[n]$ are obtained
- Suitable for computer-based computation
- EXAMPLE: Inverse z-transform using polynomial division

Solving of difference equations with initial conditions

- Solving of difference equations in time-domain requires experience (we skipped it due to this in our lectures)
- DTFT introduces a rather simple algorithm for this purpose, but requires zero initial conditions
- **Unilateral z-transform:** introduces similar procedure as DTFT, generalized for arbitrary initial conditions
- Unilateral z-transform is a variant of z-transform defined for right-sided sequences

$$X_1(z) = \sum_{n=0}^{\infty} x[n]z^{-n} = x[0] + x[1]z^{-1} + x[2]z^{-2} + \dots \quad (6)$$

- **Shift theorem:** differs from its bi-lateral variant and is given by

$$x[n-1] \overset{Z}{\Longleftrightarrow} z^{-1}X_1(z) + x[-1] \quad (7)$$

- EXAMPLE: Solving of difference equation with non-zero initial conditions using unilateral z-transform

Part II

Transform analysis of LTI systems

Transfer/system function (repetition)

- **Transfer function** $H(z)$ is a z-transform of the impulse response $h[n]$:

$$y[n] = h[n] * x[n] \xrightarrow{Z} Y(z) = H(z)X(z) \tag{8}$$

- Because impulse response is a unique description of an LTI system, so is the transfer function
- For a general LTI system (IIR) given by a difference equation

$$y[n] + \sum_{k=1}^{p} a[k]y[n-k] = \sum_{k=0}^{q} b[k]x[n-k] \tag{9}$$

the transfer function is a rational function of the form

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{1 + \sum_{k=1}^{p} a[k]z^{-k}} = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})}, \tag{10}$$

where roots of the numerator are *zeros* (β_k) a roots of the denominator are *poles* (α_k).

Tranfer function: causality, stability

- **Stability:** LTI system is stable if ROC of its transfer function contains unit circle
- DETAILS: Why is it so?

- **Causality**: Impulse response of a causal system is a right-sided sequence, ROC of the corresponding transfer function is thus of the form $|z| > \alpha$. The poles of the transfer function cannot lie within the ROC
- Consequently, all poles of a causal system must lie inside or on a circle $|z| \leq \alpha$

- **Realizable system:** is both *stable* and *causal*
- Transfer function has ROC of the form $|z| > \alpha, \quad 0 \leq \alpha < 1$, poles then *must* lie inside a unit circle

- If an LTI system has transfer function $H(z)$, then its **inverse system** $G(z)$ is given by

$$G(z) = \frac{1}{H(z)} \tag{11}$$

- ROC of the inverse system $G(z)$ must have an overlap with the ROC of $H(z)$
- EXAMPLE: Inverse system computation and its properties

- Lets have an LTI system with rational transfer function

$$H(z) = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})}, \tag{12}$$

- Let us assume only first-order poles. If $p > q$, then $H(z)$ can be expanded into

$$H(z) = \sum_{k=1}^{p} \frac{A_k}{1-\alpha_k z^{-1}}. \tag{13}$$

If the system is causal, then the *impulse response* is of the form

$$h[n] = \sum_{k=1}^{p} A_k(\alpha_k)^n u[n] \tag{14}$$

- EXAMPLE: Transfer function and impulse response

- If $p \leq q$, then $H(z)$ is of the form

$$H(z) = \sum_{k=0}^{q-p} B_k z^{-k} + \sum_{k=1}^{p} \frac{A_k}{1 - \alpha_k z^{-1}}, \tag{15}$$

and (if the system is causal) the *impulse response* is of the form

$$h[n] = \sum_{k=0}^{q-p} B_k \delta[n-k] + \sum_{k=1}^{p} A_k (\alpha_k)^n u[n]. \tag{16}$$

- If $H(z)$ has only zeros

$$H(z) = \prod_{k=1}^{q} (1 - \beta_k z^{-1}), \tag{17}$$

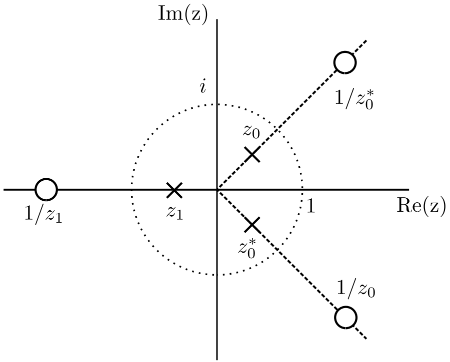
the impulse response is of a finite length and is given by

$$h[n] = \sum_{k=0}^{q} B_k \delta[n-k]. \tag{18}$$

Allpass filters I

- **Allpass filter** is a system with constant magnitude response
$$|H(e^{j\omega})| = 1. \tag{19}$$
- Allpass filters can be used for *equalization of a group delay*. This is a compensation of phase nonlinearities of other filters, such that the magnitude response of the original filter is not changed.
- The constraint placed on the magnitude response determines the location of *zeros and poles* in the z-plane. These occur in complex conjugate pairs of the form
$$H(z) = \prod_{k=1}^{N} \frac{z^{-1} - a_k^*}{1 - a_k z^{-1}}. \tag{20}$$
- Stable/causal allpass filter has *non-negative* group delay
- Stable/causal allpass filter has all poles inside and zeros outside the unit circle
- An inverse system to an allpass filter is also an allpass filter
- EXAMPLE: Equalization of the phase response of a filter

Allpass filters II



Constraints to a location of zeros and poles of a realizable allpass filter

Minimum phase filters I

- Let us have a system with a transfer function

$$H(z) = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})}. \tag{21}$$

- This system is realizable, if the poles α_k are located inside a unit circle
- The location of zeros in the z-plane may be arbitrary
- System $H_{min}(z)$ is said to have a *minimum phase*, if it has a realizable inverse system (i.e., all its *zeros* lie inside the unit circle)
- Any realizable system, which does not have zeros on the unit circle, can be transformed into a system with minimum phase
- This transformation is useful when:
  1. the *existence of an inverse filter* must be ensured
  2. the system $H(z)$ is required to have a *minimum group delay* τ_g (given a specific magnitude response)

- The transfer function $H(z)$ of any realizable system can be written in the form

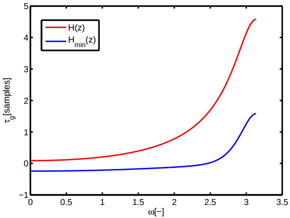
$$H(z) = H_{min}(z) \cdot H_{apr}(z), \tag{22}$$

where $H_{min}(z)$ is a minimum phase system and $H_{apr}(z)$ is a realizable allpass filter.

- Due to this factorization, $H_{min}(z)$ exhibits a minimum group delay because it holds that

$$\tau(\omega) = \tau_{min}(\omega) + \tau_{apr}(\omega). \tag{23}$$

- To obtain $H_{min}(z)$, it is necessary to "suitably move" zeros located outside the unit circle into the unit circle
- This can be done using a cascade of $H(z)$ and an inverse filter to $H_{apr}(z)$ (it is a non-causal allpass filter with zeros inside and poles outside the unit circle)
- The poles of this non-causal allpass filter cancel with zeros of $H(z)$, its zeros (located inside the unit circle) remain unchanged



Part I

Introduction to digital filtering

Digital filtering:

- Change of magnitude/phase of the spectrum
- **Magnitude filtration:** (more common) Change of the magnitude spectrum, while retaining the phase spectrum
- Delay is the only allowed distortion, (in the ideal case) does not deform the signal in the time-domain (if the signal spectrum lies in the pass-band of the filter)
- **Phase filtration:** Equalization of the phase spectrum, while maintaining the magnitude spectrum, performed by all-pass filters
- **Application:**
  - *Noise reduction:* bio-signals - ECG/EEG, digitization of analog recordings
  - *Frequency band emphasizing:* Equalization of audio signals, line detectors (high frequencies) in image data
  - *Band limitation:* Sampling (aliasing prevention), radio/tv transmission
  - *Zeroing of specific frequency components:* Suppression of DC component, suppression of power-line frequency
  - Special operations: Differentiations, Integrations, Hilbert transform

Analog vs digital filtering

Analog filtering:

- Works on continuous signals, in electronic circuits from amplifiers, resistors, capacitors ...
- In theory, unlimited frequency band, in practice limited by utilized technology
- Sensitive to noise, quality of components, limited flexibility and repeatability

Digital filtering:

- Operates with discrete signals, implemented in computers or specialized hardware (signal processors)
- Limited frequency band, equal to half of the sampling frequency
- Arbitrary level of accuracy, highly linear and flexible (adaptive filtering), perfectly repeatable.
- Allows operations, which cannot be (or with high difficulty) performed in analog domain (pure delay, adaptive filtering)
- Requires pre/post-processing (A/D-D/A conversion), anti-aliasing.

Part II

Design of frequency selective filters

Design of frequency selective filters

- Stems from spectral analysis and analysis of LTI systems via Z-transform
- Digital LTI filters are often described by the system function in the form

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{\sum_{k=0}^{p} a[k]z^{-k}} = C\frac{\prod_{k=1}^{q}(1-\beta_k z^{-1})}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})}. \quad (1)$$

- Filter having $p \geq 1$ are denoted as *IIR*, these have infinite impulse response $h[n]$ because of the feedback
- Filters having $p = 0$ are denoted as *FIR*, these have finite impulse response $h[n]$, the non-zero samples of $h[n]$ are equal to coefficients $b[k]$
- RCSR - Real, causal, stable, rational

Filter specifications (of magnitude)

Filter specifications:

- Define requirements on magnitude properties of the filter
- *Low-pass, high-pass, band-pass, band-stop, multi-band filters*
- Real-valued filter coefficients desired, magnitude of $H(e^{j\omega})$ is thus even and phase odd
- Specifications are thus necessary only in the band $0 < \omega < \pi$

Filter specifications - parameters:

- Pass band / Stop band / Transition band
- Cutoff frequency for pass-band ω_p / stop-band ω_s
- Positive δ^+/ negative δ^- tolerance in the pass-band
- Desired magnitude in the pass-band is 1
- Tolerance in the stop-band δ_s
- Desired magnitude in the stop-band is 0

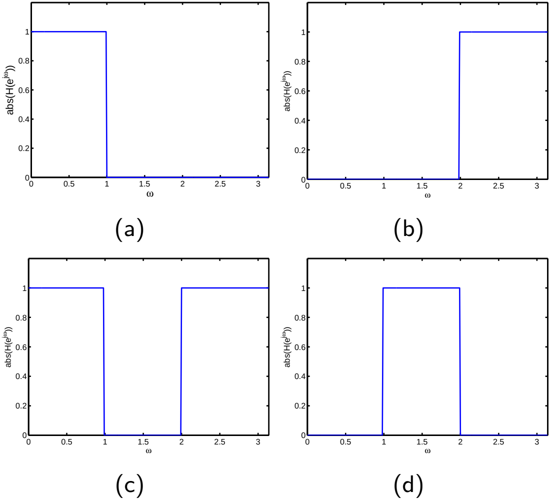


Figure: (a) Low-pass (b) High-pass (c) Band-pass (d) Band-stop

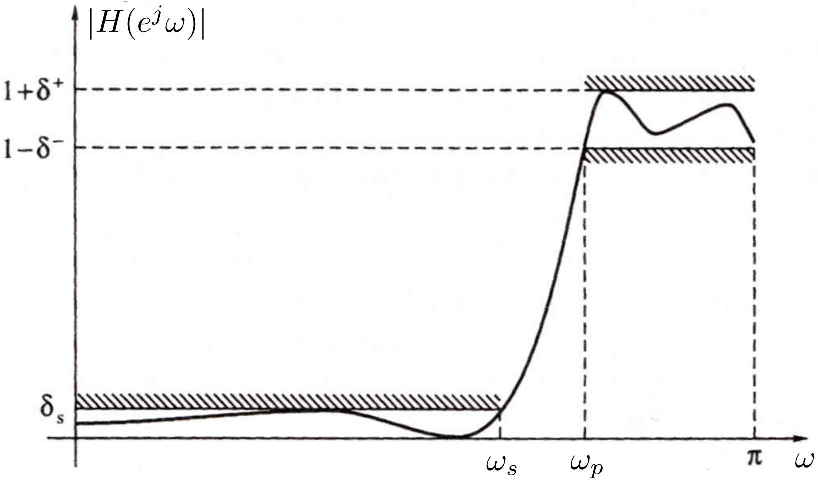
Filter specifications (of magnitude) II

Filter specifications - parameters: (continued)

- Pass band ripple: $\max(\delta^+, \delta^-)$
- Pass band ripple in dB: $A_p = max\{20\log_{10}(1+\delta^+), -20\log_{10}(1-\delta^-)\}$
- Tolerances δ are usually stated differently for FIR and IIR filters
- **IIR:** $\delta^+ = 0, \delta^- = \delta_p$ - Maximal magnitude of IIR is 1
- **FIR:** $\delta^+ = \delta^- = \delta_p$ - Middle magnitude of FIR is 1
- Filter specifications for IIR filters are sometimes (for historical reasons from analog filters) given as magnitude squared
- Attenuation in the stop-band in dB: $A_s = -20\log_{10}(\delta_s)$

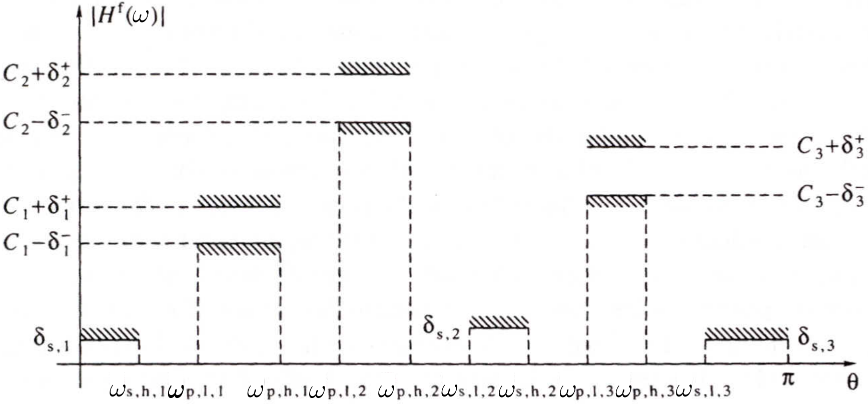
Multi-band filters:

- Allow various gain or attenuation for several frequency bands
- **Piece-wise constant multi-band filter:**
- Splits the frequency interval $[0, \pi]$ into a finite number of bands; some of these are pass-/stop-bands, the rest are transitional bands
- Each pass-/stop-band has its own tolerances δ and amplifications A_p, A_s



Filter specifications for high-pass filter

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing



Filter specifications for multi-band filter

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Phase response of a filter

- **Phase characteristic** states the change in phase of a given frequency component at the output of the filter
- **Phase distortion** appears, if the components present in the signal are shifted by varying phase delay
- This distortion influences the shape of the signal in the time-domain
- We will focus on filters, which (almost) do not distort the signal in the time-domain. These are denoted as **filters with linear phase**
- Digital filter exhibits linear phase, if

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega}, \quad \alpha \in \mathcal{R} \qquad (2)$$

$A(e^{j\omega}) \in \mathcal{R}$ - Amplitude (can be positive or negative)

- The frequency response described in this manner (using $A(e^{j\omega})$ instead of $|H(e^{j\omega})|$) exhibits **continuous phase**
- DETAILS: Continuous representation of the phase response
- Other type of a filter, which does not distort *envelope of modulated signals*, is called **generalized linear phase** filter
- Digital filter exhibits generalized linear phase, if

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\alpha\omega-\beta}, \quad \alpha, \beta \in \mathcal{R} \qquad (3)$$

Phase response of a filter II

- **Group/ phase delay** is defined as

$$\tau_g(\omega) = -\frac{d\phi(e^{j\omega})}{d\omega}, \;\; \tau_p(\omega) = -\frac{\phi(e^{j\omega})}{\omega} \tag{4}$$

where $\phi(e^{j\omega})$ is the phase response of the filter

- Unit of τ_g, τ_p is sample
- Systems with linear phase exhibit constant group/phase delay

$$\tau_g(\omega) = \tau_p(\omega) = \alpha, \tag{5}$$

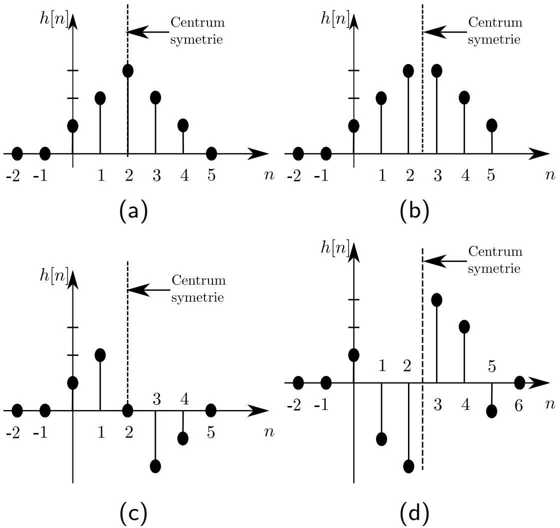
that is, they only delay the signal and do not deform it in the time-domain (if the signal spectrum lies in the pass-band of the filter).

- Group/ phase delay *need not* be integer
- DETAILS: Non-integer group delay

Phase response of a filter III

- As stated in prior lectures, the filter with linear phase need to be a FIR system
- Based on a type of impulse response $h[n]$ with length $N + 1$ there exist four types of the filter with (generalized) linear phase:
  1. 1.typ - symmetric $h[n]$, N is even, linear phase
  2. 2.typ - symmetric $h[n]$, N is odd, linear phase
  3. 3.typ - anti-symmetric $h[n]$, N is even, generalized linear phase
  4. 4.typ - anti-symmetric $h[n]$, N is odd, generalized linear phase
- N - order of FIR filter
- Group delay of these FIR filters is equal $\tau_g = N/2$
- Phase delay is for filters of type 1. and 2. equal $\tau_p = N/2$

Phase response of a filter IV



(a) Impulse response of a linear-phase FIR filter - type I, (b) Impulse response of a linear-phase FIR filter - type II
(c) Imp. response of a gen. linear-phase filter - type III, (d) Imp. response of a gen. linear-phase filter - type IV

Part III

Finite impulse response filters

Finite impulse response filters (FIR)

Finite Impulse Response:

- Finite number of non-zero samples within the impulse response $h[n]$
- Described by non-recursive LCCDE

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] \tag{6}$$

- Always stable, with a system function given by

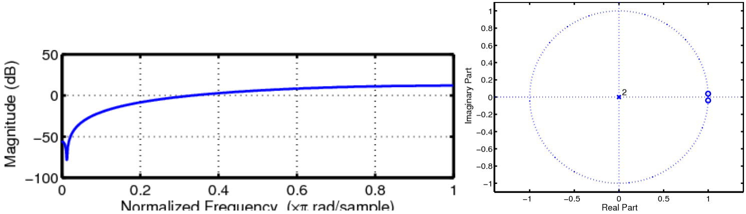
$$X(z) = \sum_{k=0}^{q} b[k]z^{-k} = C\sum_{k=1}^{q}(1-\beta_k z^{-1}), \tag{7}$$

  thus with q zeros and q-fold pole in the origin of the z-plane.
- FIR filters can be designed with linear phase, i.e., they do not distort signal (if its spectrum lies in the pass-band of the filter).

Part IV

Notch filter design

- Completely suppresses single frequency component
- Designed by setting a null in z-plane on a specific frequency ω_0
- EXAMPLE: Suppression of the power-hum

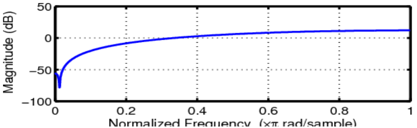


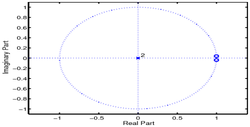
Notch filter - Magnitude response, Z-plane

Notch filter - IIR

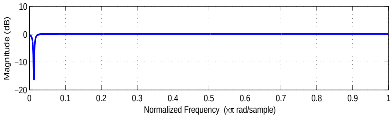
- Completely suppresses single frequency component
- Designed by setting a null in z-plane on a specific frequency ω_0
- A pole is set "close" to the zero to suppress distortion of neighbor frequencies
- EXAMPLE: Suppression of the power-hum
- $H(z) = \frac{z^2 - 1.9985z^1 + 1}{z^2 - 1.9796z^1 + 0.9801} = \frac{\left(z - e^{\frac{j\cdot 2\cdot\pi\cdot 50}{8000}}\right)\left(z - e^{-\frac{j\cdot 2\cdot\pi\cdot 50}{8000}}\right)}{\left(z - 0.99\cdot e^{\frac{j\cdot 2\cdot\pi\cdot 50}{8000}}\right)\left(z - 0.99\cdot e^{-\frac{j\cdot 2\cdot\pi\cdot 50}{8000}}\right)}$

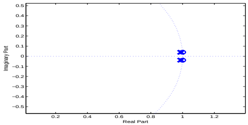
FIR:





IIR:





Part V

Impulse response truncation, filter design using windows

Impulse Response Truncation I

Impulse Response Truncation: (IRT) [zkrácení/oriznutí]

- Impulse response $h[n]$ of an ideal filter is infinite and non-causal
- However, it has finite energy, it is thus possible to "truncate" it
- Truncated and shifted impulse response the corresponds to a filter, which approximates the ideal filter
- The longer is the part of the impulse response, which is preserved, the more the truncated filter approaches to the ideal filter
- The ideal band-pass filter has the amplitude given by

$$A(e^{j\omega}) = \begin{cases} 1, & \omega_1 \leq |\omega| \leq \omega_2 \\ 0, & \text{else} \end{cases} \tag{8}$$

- Low-pass filter is obtained if $\omega_1 = 0$, the high-pass if $\omega_2 = \pi$
- The linear phase is desired, then the phase $e^{j\phi(e^{j\omega})}$ is given by (FIR filter of type I or II)

$$\phi(e^{j\omega}) = \begin{cases} -\omega N/2, & \omega_1 \leq |\omega| \leq \omega_2 \\ 0, & \text{else} \end{cases} \tag{9}$$

Impulse Response Truncation II

- Impulse response of the desired band-pass is obtained by

$$h_{BP}[n] = \frac{1}{2\pi}\int_{-\omega_2}^{-\omega_1} e^{j\omega(n-0.5N)}d\omega + \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} e^{j\omega(n-0.5N)}d\omega$$
$$= \frac{\omega_2}{\pi}\mathrm{sinc}\left[\frac{\omega_2(n-0.5N)}{\pi}\right] - \frac{\omega_1}{\pi}\mathrm{sinc}\left[\frac{\omega_1(n-0.5N)}{\pi}\right] \quad (10)$$

- Low-pass filter is achieved by substitution $\omega_1 = 0$

$$h_{LP}[n] = \frac{\omega_2}{\pi}\mathrm{sinc}\left[\frac{\omega_2(n-0.5N)}{\pi}\right] \quad (11)$$

- If N is even, high-pass is achieved by substitution $\omega_2 = \pi$ (cannot be designed for N odd, FIR of type II is not suitable for high-pass):

$$h_{HP}[n] = \delta[n-0.5N] - \frac{\omega_1}{\pi}\mathrm{sinc}\left[\frac{\omega_1(n-0.5N)}{\pi}\right] \quad (12)$$

Multi-pass filters

- Amplitude response of an ideal multi-pass filter is obtained by superposition of K suitable band-pass filters

$$A(e^{j\omega}) = \sum_{k=1}^{K} A_k(e^{j\omega}), \tag{13}$$

kde

$$A_k(e^{j\omega}) = \begin{cases} C_k, & \omega_{1,k} \leq |\omega| \leq \omega_{2,k} \\ 0, & \text{else} \end{cases} \tag{14}$$

- The impulse response of the ideal multi-band filter is thus given by

$$h_{MB}[n] = \sum_{k=1}^{K} \frac{C_k}{\pi} \left\{ \omega_{2,k} \text{sinc} \left[\frac{\omega_{2,k}(n-0.5N)}{\pi} \right] - \omega_{1,k} \text{sinc} \left[\frac{\omega_{1,k}(n-0.5N)}{\pi} \right] \right\} \tag{15}$$

- If N is even, (FIR type II is not suitable for na HP), then band-stop filter for $K=2$ is obtained by substitution $C_1 = C_2 = 1, \omega_{1,1} = 0, \omega_{2,2} = \pi$

Impulse Response Truncation IV

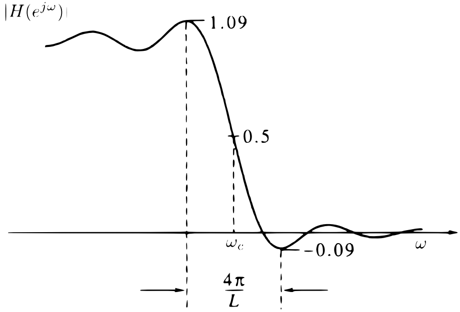
Filter order selection

- Impulse response $h_{ID}[n]$ of an ideal filter is infinite
- By truncation of $h_{ID}[n]$ we obtain an impulse response $h[n]$ of a filter, which differs from the ideal one by ripples δ_p/δ_s in the pass-/stop-band and existence of a transitional band (ripple = vlnka)
- With increasing order of the filter N, the transitional band becomes narrower; the ripples remain practically the same - *Gibbs phenomenon*
- The truncation of the impulse response can be described by

$$h[n] = h_{ID}[n]w_r[n] \tag{16}$$

  where $w_r[n]$ is rectangular window of length $L = N + 1$ (N - filter order).
- **Transitional band width** depends on the width of the main-lobe of a Dirichlet kernel and is equal $4\pi/L$
- **Ripples** δ_p, δ_s depend on the magnitude of a side-lobe of Dirichlet kernel and is practically independent of window $w_r[n]$ length
- Ripples $\delta_p \approx \delta_s \approx 0.09$, that is $A_p = 0.75$dB, $A_s = 21$dB.

- Ripples δ achieved using $w_r[n]$ are unsuitable for many real-world applications. These can only be suppressed by utilization of other windows (with lower magnitude of the side-lobes) when truncating the impulse response - Filter design using windows



Transitional band of a filter designed by the IRT method

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Filter design using windows

Filter design using windows:

- Generalization of the Impulse Response Truncation method
- Truncation of the impulse response can be described by

$$h[n] = h_{ID}[n]w[n], \tag{17}$$

$w[n]$ is a window of length $L = N + 1$ (N - desired filter order)

- The type of the window influences the properties of the filter, because

$$H(e^{j\omega}) = \frac{1}{2\pi}\left\{H_{ID}(e^{j\omega}) * W(e^{j\omega})\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{ID}(e^{j\lambda})W(e^{j\omega-\lambda})d\lambda \tag{18}$$

- This is smoothing of $H_{ID}(e^{j\omega})$ using the spectrum of the window $W(e^{j\omega})$
- **Main-lobe width:** influences the width of the transitional band
- **Side-lobe magnitude:** influences the size of the pass-/stop-band ripples

Filter design using windows II

Design algorithm:

1. According to allowed ripple size, the type of window is selected
2. According to desired transitional band width, the order/length of $h[n]$ is selected ($N = L - 1$)
3. According to equations stated for method IRT, L coefficients of the impulse response are computed, which are multiplied by the selected window

MATLAB: `B=FIR1(n,Wn)`

Parameters of frequently used windows:

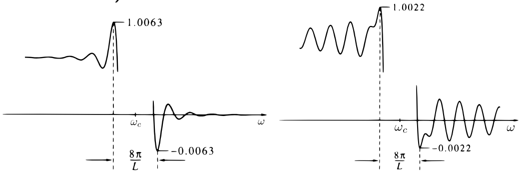
Window	Main-lobe width	Side-lobe magnitude [dB]	δ_p, δ_s
Rectangular	$4\pi/L$	-13,5	0.09
Bartlett	$8\pi/L$	-27	0.05
Hann	$8\pi/L$	-32	0.0063
Hamming	$8\pi/L$	-43	0.0022
Blackman	$12\pi/L$	-57	0.0002
Kaiser	Depends on α		

- $A_p = \max\{20\log_{10}(1+\delta^+), -20\log_{10}(1-\delta^-)\} \approx 8.6859\max(\delta^+, \delta^-)$
- $A_s = -20\log_{10}(\delta_s)$

Filter design using windows III

Properties of the commonly used windows

- **Bartlett window**: Exhibits monotonous magnitude (without ripples) in the vicinity of the transitional band
- Magnitude tolerance is therefore not well defined, it is usually stated as $\delta_p = \delta_s = 0.05$
- The decrease of ripples with respect to rectangular window is rather small, but the transitional band is twice as wide ($8\pi/L$)
- **Hann a Hamming window**: Equally wide transition band (slightly less than $8\pi/L$)
- *Hann:* Larger magnitude of ripples, more distinct ripple attenuation than Hamming
- *Hamming:* Lower magnitude of ripples, gradual ripple attenuation (compared to Hann)



Ripples of filters designed using Hann and Hamming windows

Filter design using windows IV

Properties of the commonly used windows (continuation)

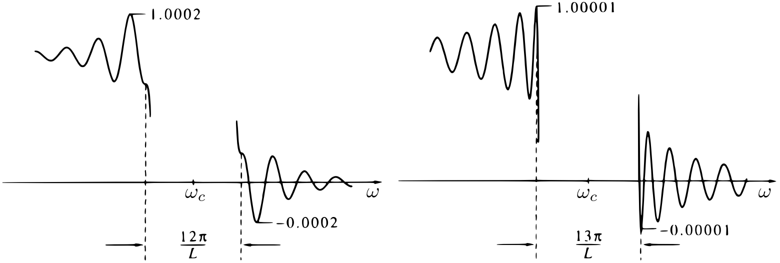
- **Blackman window**: Wide transition band ($12\pi/L$), very small ripples $\delta_p = \delta_s = 0.0002$
- **Kaiser window**: Frequently used window for filter design
- Its shape and properties can be altered using free parameter α
- Empiric formulas have been designed to relate α and specifications of the designed filter
- The order of the filter is selected as a nearest integer above the order N computed via (21)
- The formulas are empiric, the designed filter is not guaranteed to fulfill the desired specifications (usually, a higher order needs to be selected)

$$A = -20\log_{10}(\min\{\delta_p, \delta_s\}) \tag{19}$$

$$\alpha = \begin{cases} 0.1102(A-8.7), & A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21), & 21 < A \leq 50 \\ 0, & A \leq 21 \end{cases} \tag{20}$$

$$N = \frac{A - 7.95}{2.285|\omega_p - \omega_s|} \tag{21}$$

Filter design using windows V



Ripples of filters designed using Blackman a Kaiser window ($\alpha = 10$)

Source: BOAZ PORAT, A Course in Digital Signal Processing

Part I

FIR filter design (continued)

Frequency Sampling Filter Design:

- Method is based on equidistant sampling of the desired ideal frequency response $H_{ID}(e^{j\omega})$ in L points
- The samples

$$H[k] = H_{ID}(e^{j2\pi k/L}), \quad k = 0, \ldots, L-1 \tag{1}$$

  ae obtained, which correspond to the L-point DFT of the impulse response $h[n]$ of the FIR filter of order $N = L - 1$.
- The frequency response of the designed filter $H(e^{j\omega})$ corresponds exactly to the response of the ideal filter $H_{ID}(e^{j\omega})$
- However, the method does not provide any control, how the frequency response $H(e^{j\omega})$ behaves *between samples* of $H[k]$
- Filters designed by this method are therefore considered inadequate
- Improved behavior around transition band can be obtained by inclusion of samples, which describe it
- MATLAB: `B=FIR2(N,F,M)`

Filter design based on optimality criteria

- Filter design using windows is simple and leads to sufficiently accurate filters
- However, it is suboptimal from two points of view
- **Tolerances δ_p, δ_s must equal** and cannot be changed independently
- In practice, the ripple δ_s is required much lower than δ_p is allowed
- Impossibility of independent design leads to the necessity, to select tolerances in the pass-band too strict, in order to comply with the tolerances in the stop-band (results in higher order of the resulting filter)
- **Distribution of ripples is uneven** for most of the windows and diminishes in the direction away from the transitional band

Advanced design methods allow more freedom in the selection of the magnitude response, allow independently change δ_p, δ_s and/or distribute ripples equally

Least square design of frequency selective FIR filter

- Method allowing independent selection of tolerances δ_p/δ_s
- It is based on minimization of a criterion given by

$$\epsilon^2 = \int_0^{\pi} \left(V(\omega)[A_d(\omega) - A(\omega)]\right)^2 d\omega, \tag{2}$$

  where $V(\omega)$ is the weight (importance) assigned to frequency ω within the design and $(A_d(\omega) - A(\omega))$ is the deviation of the given and desired response
- **Weight** $V(\omega)$ is a non-negative number determining, how much the difference $(A_d(\omega) - A(\omega))$ is undesired in the given band
- Usually, it is selected reciprocal to the tolerances δ, the lesser the tolerance the bigger the weight
- Transition bands have zero weight
- Ususally, the linear phase is desired, the resulting $h[n]$ is thus symmetric
- MATLAB: `B=FIRLS(N,F,A,W)`
- REMARK: This design method should not be mistaken with design of *optimal filter in the least square sense* (LSE). The LSE filter is completely data-based, it is not a frequency selective filter (high pass, low-pass etc.)

Equiripple design of FIR filters

- Designs filters with evenly distributed ripples and independent selection of tolerances δ_p/δ_s
- Complicated computation of coefficients via numerical solution to an optimization problem
- Minimizes criterion given by

$$\epsilon = \max_{\omega \in \mathcal{S}} |E(\omega)| \tag{3}$$

where

$$E(\omega) = \tilde{V}(\omega)[\tilde{A}_d(\omega) - G(\omega)] \tag{4}$$

$$\tilde{V}(\omega) = V(\omega)F(\omega), \tilde{A}_d = \frac{A_d(\omega)}{F(\omega)} \tag{5}$$

- Function $F(\omega)$ depends on a filter type, e.g., FIR type I: $F(\omega) = 1$
- $V(\omega)$ is a weight function, $A(\omega)$ - amplitude of the designed filter
- $A(\omega) = F(\omega)G(\omega)$
- $\mathcal{S}$ - A set of frequencies in the interval $< 0, \pi >$, which correspond to pass-/stop-bands of the designed filter
- $A_d(\omega)$ represents the desired amplitude characteristics

Equiripple design of FIR filters II

- Optimization of criterion (3) stems from the *alternation theorem*
- The numerical solution was originally proposed by Remez (1957) - *Remez exchange algorithm*
- Nowadays, the design is realized through an algorithm proposed by Parks a McCllelan (1982) - *Parks-McClellan algorithm*
- The order of the filter is selected via an ad-hoc empirical formula designed by Kaiser

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{2.32|\omega_p - \omega_s|} \tag{6}$$

- MATLAB: `[N,Fo,Ao,W] = FIRPMORD(F,A,DEV,Fs)` - Determines the order based on equation (6)
  `B=FIRPM(N,Fo,Ao,W)` - Computation of the impulse response
- EXAMPLE: Comparison of the discussed methods for FIR design (`FIRDesignMethodsComparison.m`)

Part II

IIR filter design

Infinite Impulse Response Filters (IIR)

- Impulse response $h[n]$ is infinite right-sided sequence
- Described by recursive difference equation

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] - \sum_{k=1}^{p} a[k]y[n-k] \tag{7}$$

- Can be unstable, the system function is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{1 + \sum_{k=1}^{p} a[k]z^{-k}}, \tag{8}$$

  where q is the number of zeros and p is the number of poles.
- IIR filters cannot be designed to have *(generalized) linear* phase response
- Compared to FIR, filters with the same order have narrower transition band

IIR filter design

- IIR design methods are known for a long time (from analog system theory)
- Design is performed via *discretization of analog filters*
- Advantage: well known algorithms with given properties
- Disadvantage: limited design flexibility (to basic filter types LP, HP, BP, BS)
- Multi-band filter design is difficult
- IIR filters can be designed in the digital domain, but the methods are not very popular (either too difficult or mediocre results)
- Analog and digital filter coincide for the lowest band of frequencies (up to Nyquist frequency)
- IIR design is focused on the magnitude response, the phase response is of a minor importance
- *Phase distortion* can be significant due to non-linear phase

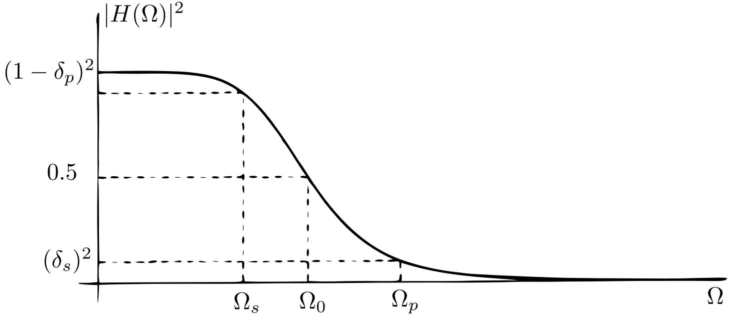
Butterworth filters I

- Butterworth filters are defined using squared magnited as

$$|H(\Omega)|^2 = \frac{1}{1 + (\frac{\Omega}{\Omega_0})^{2N}} \tag{9}$$

- **Butterworth filter properties**:
  - Magnitude response is monotone decreasing function of Ω
  - Maximum is located in $|H(\Omega)|_{\Omega=0} = 1$
  - Attenuation of square magnitude by 3dB (to 0.5) is located in Ω_0
  - Magnitude response is practically constant for low frequencies
- MATLAB:
  `[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs)` - Filter order for given specifications
  `[B,A] = BUTTER(N,Wn)` - Filter parameters

Butterworth filters II



Squared magnitude of analog Butterworth filter (N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

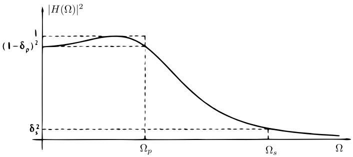
Chebyshev filters I

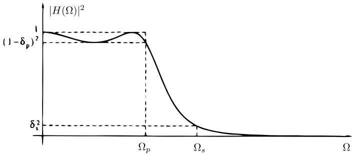
- Two filter types design based on Chebyshev polynomial
- Chebyshev filters exhibit monotone magnitude response either in the passband (type I) or in the stopband (type II)
- Ripples allow narrower transition band compared to Butterworth filter
- **Chebyshev filters type I** are defined using squared magnitude response as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 T_N^2(\frac{\Omega}{\Omega_0})}, \qquad (10)$$

  Ω_0, ϵ - free parameters, $T_N(\cdot)$ - Chebyshev polynomial of order N
- **Properties of Chebyshev I filters**:
  - Magnitude response exhibit ripples in the passband
  - Squared magnitude equals 1 (odd N) or $1/(1+\epsilon^2)$ (even N) for $\Omega = 0$
  - Maximum value 1 is achieved several times in the pass band (based on filter order N)
  - For $\Omega > \Omega_0$ is the magnitude response monotone decreasing

- MATLAB:
  `[N, Wp] = CHEB1ORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
  `[B,A] = CHEBY1(N,R,Wp)` - Filter parameter computation





Squared magnitude response of analog Chebyshev type I filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Chebyshev filters III

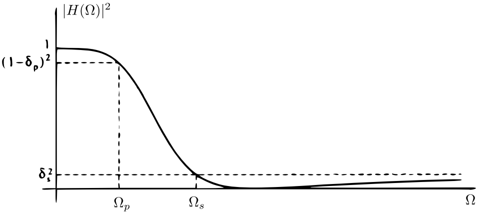
- **Chebyshev type II filters** are defined using the squared magnitude response as

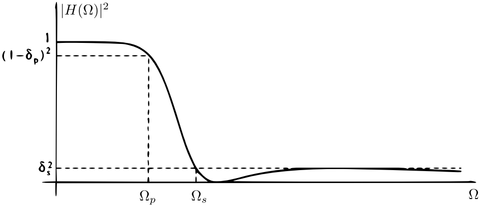
$$|H(\Omega)|^2 = 1 - \frac{1}{1 + \epsilon^2 T_N^2(\frac{\Omega}{\Omega_0})}, \tag{11}$$

Ω_0, ϵ - free parameters, $T_N(\cdot)$ - Chebyshev polynomial of order N

- **Properties of Chebyshev II filters**:
  - For $\Omega > \Omega_0$ exhibits the magnitude response ripples
  - Squared magnitude response for $\Omega = 0$ equals 1 for all N
  - For values $\Omega < \Omega_0$ is magnitude response monotone decreasing
- MATLAB:
`[N, Ws] = CHEB2ORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
`[B,A] = CHEBY2(N,R,Wst)` - Filter parameter computation

Chebyshev filters IV





Squared magnitude response of analog Chebyshev type II filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Elliptic filters

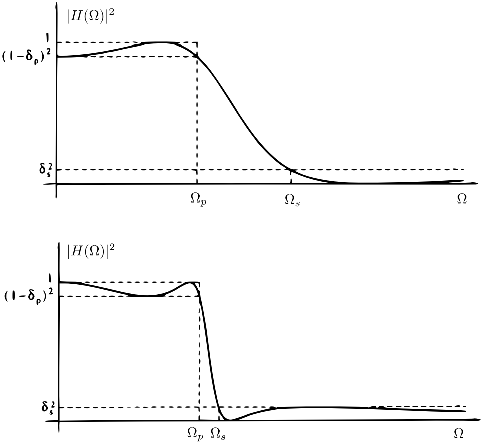
- **Elliptic filters** are defined using the squared magnitude response as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 R_N^2(\frac{\Omega}{\Omega_0})}, \tag{12}$$

Ω_0, ϵ - free parameters, $R_N(\cdot)$ - Chebyshev rational function of order N

- **Properties of Elliptic filters**:
  - Magnitude response is rippled in the passband and the in stopband
  - Ripples allow narrower transition band compared to other mentioned IIR filter types
- MATLAB:
`[N, Wp] = ELLIPORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
`[B,A] = ELLIP(N,Rp,Rs,Wp)` - Filter parameter computation

Elliptic filters II



Squared magnitude response of analog elliptic filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Comparison of FIR and IIR

FIR:

- Usually designed with linear phase - minimum phase distortion
- More intuitive filter design
- Always stable

IIR:

- Lower latency + narrower transition band compared to similar order FIR filter
- Lesser computational burden

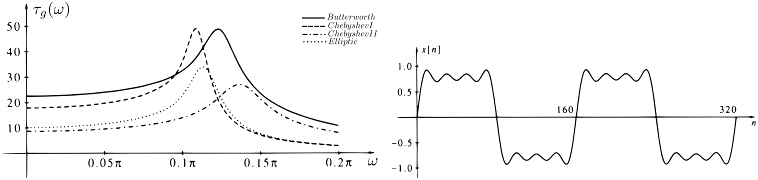
Phase distortion IIR/FIR:

- IIR cannot be designed with linear phase
- Signasl passing through IIR are distorted (each frequency component is delayed differently) even if its spectrum is located in the filter passband
- EXAMPLE: Comparison - phase distortion FIR/IIR

- Let us have a filter specification: $\omega_p = 0.1\pi, \omega_s = 0.2\pi, \delta_p = \delta_s = 0.001$
- Let us design IIR of all basic types (with minimum order) satisfying the specifications
- Let us generate a test signal (first four components of a square wave)

$$x[n] = \sum_{m=1}^{4} \frac{1}{2m-1} \sin(0.0125\pi(2m-1)n) \tag{13}$$

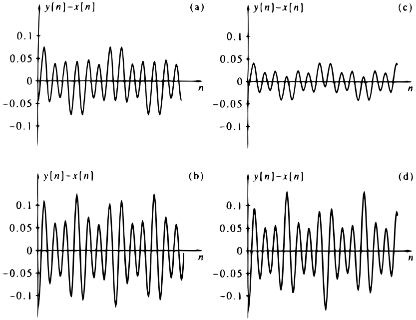
- Signal is band limited by the highest frequency 0.0875π, i.e., it is within the filter passband



(a) Designed filters: group delay, (b) Test signal $x[n]$

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

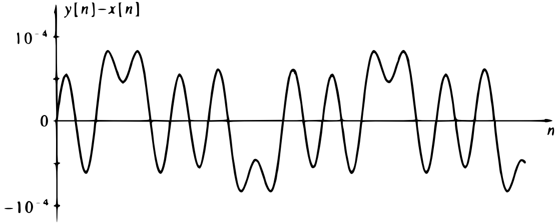
- If the signal is located within the passband of the filter, we aim for the lowest possible distortion
- Let us compare the difference between the output signal $y[n]$ and the input signal $x[n]$
- (Signal are shifted to compensate the varying group delay of the filters for the first harmonic $w = 0.0125\pi$)



$y[n] - x[n]$, (a) Butterworth, (b) Chebyshev I, (c) Chebyshev II, (d) Elliptic

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

- Allowed tolerance $\delta_p = 0.001$, phase distortion causes difference almost 100 times higher (in the time-domain)
- Let us repeat the experiment with FIR filter designed by windowing method and Kaiser window



$y[n] - x[n]$ for FIR designed by windowing method and Kaiser window

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

- Error of FIR is smaller than the desired tolerance
- FIR exhibits advantage against IIR: it eliminates the phase distortion
- FIR latency is higher compared to IIR ($\tau_{g,FIR} = 38$; $9 < \tau_{g,IIR} < 23$)

Part I

Optimal filter design - motivation (Least square error filter)

Optimal filters

- Until now we discussed the **frequency selective filters**
- Those are designed to have a specific shape of the frequency response (e.g., low-pass filter)
- The design does not require any training data, the filter is fully specified by the *filter specifications*

- The **optimal filters** follow a different design philosophy
- Their frequency response is designed by optimization of some criterion (e.g., least squares), evaluated on a set of *training signals*
- Properties of the optimal filters are thus given by the training signals; the design cannot proceed without them
- In general, these filters do not have frequency selective character
- Let us explain the optimal filter design using the example of *Target cancellation filters*

Part II

Target Cancellation Filters

Let us consider the following task:

- Two simultaneously active sound sources, recorded via a binaural microphone array
- Sound of each source should be recovered from the mixture separately
- ASSUMPTION 1: One source is fixed (two speaking people, one has to sit/stand, the other can be moving)
- ASSUMPTION 2: For some short time interval, the fixed source is active alone

The fixed source can be recovered using the **Target cancellation filter** (an example of an optimal filter) and the moving source can be estimated using an **adaptive filter** (details are outside the scope of the Lecture)

Target Cancellation Filters II

Formal problem description:

- Stereo recording with simultaneously active fixed source $s[n]$ and moving source $y[n]$ given by

$$\begin{aligned} x_{\mathrm{L}}[n] &= \{h_{\mathrm{L}} * s\}[n] + y_{\mathrm{L}}[n], \\ x_{\mathrm{R}}[n] &= \{h_{\mathrm{R}} * s\}[n] + y_{\mathrm{R}}[n]. \end{aligned} \tag{1}$$

  - $x_{\mathrm{L}}[n]$, $x_{\mathrm{R}}[n]$ ... two channels of the recordings
  - $y_{\mathrm{L}}[n]$, $y_{\mathrm{R}}[n]$ ... signal $y[n]$ at left and right microphone
  - $h_{\mathrm{L}}[n]$, $h_{\mathrm{R}}[n]$ ... acoustic impulse responses binding $s[n]$ to its image on left and right microphone

- NOTE: $\{h_{\mathrm{L}} * s\}[n]$ is an alternative way to denote convolution $\{h_{\mathrm{L}}[n] * s[n]\}$, which is sometimes used in the literature
- It emphasizes that we consider the nth sample of the sequence given by convolution of $h_{\mathrm{L}}[n]$ and $s[n]$.

Target Cancellation Filters III

Target Cancellation Filter (CF):

- CF blocks signal arriving from one direction in the environment
- CF is *time invariant* (i.e. LTI system), the target cannot change its position (Assumption 1).
- CF is a two-input single-output filter (Multi Input Single Output, MISO).

MISO filters in general:

- A set of single channel filters $g_i[n], i = 1 \dots I$ (for CF, $I = 2$)
- *Application:* Convolution of impulse response $g_i[n]$ with ith channel of the input (for $i = 1 \dots I$) and summation of all the output signals sample-wise.

Adaptive filter: changes its inner parameters (impulse/frequency response) in time.

Target Cancellation Filters IV

- CF can be designed as two filters: a general $g[n]$ (to be designed via least squares) and simple delay $g_2[n] = -\delta[n]$.
- Coefficients of g are selected to fulfill

$$\{g * h_{\mathrm{L}}\}[n] = h_{\mathrm{R}}[n], \tag{2}$$

  i.e., the response of the left sensor (with respect to source $s[n]$) is filtered to be equal with the response of the right sensor.
- The filter $g[n]$ is thus given through the room impulse responses $h_L[n]$, $h_R[n]$, which depend on the acoustic properties of the environment (reverberation) and the location of the fixed source
- The application of the CF results into output

$$\begin{aligned} v[n] &= \{g * x_{\mathrm{L}}\}[n] + \{g_2 * x_{\mathrm{R}}\}[n] \\ &= \{g * x_{\mathrm{L}}\}[n] - x_{\mathrm{R}}[n] \\ &= \{g * h_{\mathrm{L}} * s\}[n] + \{g * y_{\mathrm{L}}\}[n] - \{h_{\mathrm{R}} * s\}[n] - y_{\mathrm{R}}[n] \\ &= \{h_{\mathrm{R}} * s\}[n] + \{g * y_{\mathrm{L}}\}[n] - \{h_{\mathrm{R}} * s\}[n] - y_{\mathrm{R}}[n] \\ &= \{g * y_{\mathrm{L}}\}[n] - y_{\mathrm{R}}[n], \end{aligned} \tag{3}$$

  which does not contain $s[n]$, while y_{L} and y_{R} are passed through.
- Signal $v[n]$ thus represents our estimate of the moving source $y[n]$.

How to compute g in practice:

- Filter g is computed using some interval $(n = N_1, \ldots, N_2)$ within $x_{\mathrm{L}}[n]$ and $x_{\mathrm{R}}[n]$, where only the fixed source is active (ASSUMPTION 2).
- If $y_{\mathrm{L}}(n) = y_{\mathrm{R}}(n) = 0$, then g is given as a solution ot a set of equations

$$g = \arg\min_{g} \sum_{n=N_1}^{N_2} \Big| \{g * x_{\mathrm{L}} - x_{\mathrm{R}}\}[n] \Big|^2. \tag{4}$$

- This is a classical problem in signal processing, it is a **least squares design** of an optimal filter
- The filter is thus given through the training signals $x_L[n], x_R[n]$.
- MORE PRECISELY: by definition, $g[n]$ is given solely by impulse responses $h_L[n], h_R[n]$. In practice though, the estimate is also influenced by the signal $s[n]$, because $x_L[n] = \{h_L * s\}[n]$ a $x_R[n] = \{h_R * s\}[n]$.
- The training interval can be short (about 1 s of a signal).

Target Cancellation Filters VI

Adaptive filtering:

- Using the estimate of $v[n]$, we can also estimate the target $s[n]$ using the adaptive filtering.
- The adaptive (Wiener) filter is given in the spectral domain as

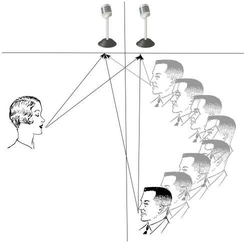
$$W[k,\ell] = \frac{|X[k,\ell]|^2}{|X[k,\ell]|^2 + \tau|V[k,\ell]|^2}. \tag{5}$$

  - $X[k,\ell]$, $V[k,\ell]$ ... short-time Discrete Fourier Transform (STFT) of signals $x_{\mathrm{L}}[n]$, $v[n]$
  - k ... index of a spectral bin
  - ℓ ... time index
  - τ ... free parameter (separation/distortion trade-off, classical Wiener $\tau = 1$)
- The STFT representation of target $s[n]$ is then given by

$$\widehat{S}[k,\ell] = W[k,\ell]X[k,\ell]. \tag{6}$$

- This is a variant of the *spectral masking*. We discussed another variant of this technique in the lecture about *spectral thresholding*.

Example:



- **Moving Source:** right half-space, variable distance to mics 0.5 m - 1.2 m, estimated using CF.
- **Fixed source:** left half-space, distance 1 m, estimated using adaptive filtering.

Original stereo recording:
Mixture channel 1
Mixture channel 2

Source estimates:
Moving source (via CF)
Fixed source (via masking)

Part III

Least square error optimal filters

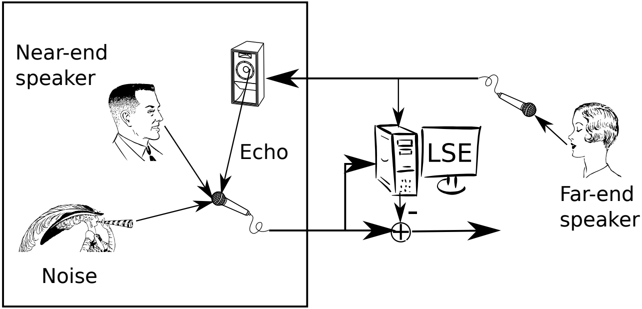
Least square error optimal filter I

Least square error optimal filter: (LSE)

- The design minimizes least square difference between filter output and the desired target signal
- Filter design is completely dependent on data, frequency response is not known a priori
- *Beware of confusing naming:* Least square error optimal filter is a completely different system compared to least square frequency selective filter (Lecture 12, given by filter specifications)
- Many applications - prediction, dereverberation, system identification, ..
- .. target (directional signal) cancellation filters; minimizes difference between channels of recording originating from a binaural microphone array
- ..echo cancellation, attenuation of sound repetitions arising during duplex communication (hands-free, conference rooms)

Echo cancellation

- Duplex communication: sound emanating from a loudspeaker is captured by a microphone and send back
- This creates an unpleasant repetition of the original sound
- Some sounds (clicks) can even amplify with each pass through the loop
- In practice, the filter need to be successively adapted to changes in the environment (e.g., location of microphone/loudspeaker); adaptive LSE variants are used (e.g., Recursive Least Squares - RLS)



Least square error optimal filter II

- Let us continue with variables defined in the previous section. Then, the task of the LSE filter is to process the signal x_{L} (defined on interval $n \in [N_1, N_2]$) by filter g (of order N_g), such that the output signal

$$\hat{x}_{\mathrm{R}}[n] = \sum_{k=0}^{N_g} g[k] x_{\mathrm{L}}[n-k] \tag{7}$$

  was as similar as possible to a target signal x_{R} in the least square sense.
- Let us define the error signal

$$e[n] = x_{\mathrm{R}}[n] - \hat{x}_{\mathrm{R}}[n], \tag{8}$$

  which should have on a given signal interval as small energy as possible, i.e., let us minimize the criterion

$$J = \sum_{n=N_1}^{N_2} e^2[n]. \tag{9}$$

Least square error optimal filter III

- Let us introduce a vector notation

$$\mathbf{x}_\mathrm{L}[n] = \begin{bmatrix} x_\mathrm{L}[n] \\ x_\mathrm{L}[n-1] \\ \vdots \\ x_\mathrm{L}[n-N_g+1] \end{bmatrix}, \mathbf{x}_\mathrm{R} = \begin{bmatrix} x_\mathrm{R}[N_1] \\ \vdots \\ x_\mathrm{R}[N_2] \end{bmatrix}, \hat{\mathbf{x}}_\mathrm{R} = \begin{bmatrix} \hat{x}_\mathrm{R}[N_1] \\ \vdots \\ \hat{x}_\mathrm{R}[N_2] \end{bmatrix}, \tag{10}$$

$$\mathbf{e} = \begin{bmatrix} e[N_1] \\ \vdots \\ e[N_2] \end{bmatrix}, \mathbf{g} = \begin{bmatrix} g[0] \\ \vdots \\ g[N_g] \end{bmatrix}. \tag{11}$$

- Convolution $\hat{x}_\mathrm{R} = g * x_\mathrm{L}$ can then be written as dot product

$$\hat{x}_\mathrm{R}[n] = \sum_{k=0}^{N_g} g[k] x_\mathrm{L}[n-k] = \mathbf{x}_\mathrm{L}^T[n]\mathbf{g} \tag{12}$$

Least square error optimal filter IV

- Error signal: $\mathbf{e} = \mathbf{x}_{\mathrm{R}} - \hat{\mathbf{x}}_{\mathrm{R}}$.
- Vector $\hat{\mathbf{x}}_{\mathrm{R}}$ can be also written as

$$\hat{\mathbf{x}}_{\mathrm{R}} = \begin{bmatrix} \mathbf{x}_{\mathrm{L}}^T[N_1]\mathbf{g} \\ \vdots \\ \mathbf{x}_{\mathrm{L}}^T[N_2]\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{\mathrm{L}}^T[N_1] \\ \vdots \\ \mathbf{x}_{\mathrm{L}}^T[N_2] \end{bmatrix} \cdot \mathbf{g} = \mathbf{A} \cdot \mathbf{g}. \qquad (13)$$

- Martix $\mathbf{A}$ with dimensions $(N_2 - N_1 + 1) \times N_g$ is therefore

$$\mathbf{A} = \begin{bmatrix} x[N_1] & x[N_1 - 1] & \dots & x[N_1 - N_b + 1] \\ x[N_1 + 1] & x[N_1] & \dots & x[N_1 - N_b + 2] \\ \vdots & \vdots & \vdots & \vdots \\ x[N_2] & x[N_2 - 1] & \dots & x[N_2 - N_b + 1] \end{bmatrix}. \qquad (14)$$

- Error signal: $\mathbf{e} = \mathbf{x}_{\mathrm{R}} - \hat{\mathbf{x}}_{\mathrm{R}} = \mathbf{x}_{\mathrm{R}} - \mathbf{A}\mathbf{g}$.

Least square error optimal filter V

- The task is to find such filter $\mathbf{g}$, such that the criterion measuring the energy of the error signal

$$J = \sum_{n=N_1}^{N_2} e^2[n]. \tag{15}$$

is as small as possible.

$$\begin{aligned} J = \mathbf{e}^T\mathbf{e} &= (\mathbf{x}_\mathrm{R} - \mathbf{A}\mathbf{g})^T(\mathbf{x}_\mathrm{R} - \mathbf{A}\mathbf{g}) \\ &= (\mathbf{x}_\mathrm{R}^T - \mathbf{g}^T\mathbf{A}^T)(\mathbf{x}_\mathrm{R} - \mathbf{A}\mathbf{g}) \\ &= \mathbf{x}_\mathrm{R}^T\mathbf{x}_\mathrm{R} - \mathbf{g}^T\mathbf{A}^T\mathbf{x}_\mathrm{R} - \mathbf{x}_\mathrm{R}^T\mathbf{A}\mathbf{g} + \mathbf{g}^T\mathbf{A}^T\mathbf{A}\mathbf{g} \\ &= \mathbf{x}_\mathrm{R}^T\mathbf{x}_\mathrm{R} - 2\mathbf{g}^T\mathbf{A}^T\mathbf{x}_\mathrm{R} + \mathbf{g}^T\mathbf{A}^T\mathbf{A}\mathbf{g}. \end{aligned} \tag{16}$$

- Global minimum is obtained by derivative of J by vector $\mathbf{g}$

$$\frac{dJ}{d\mathbf{g}} = \begin{bmatrix} \frac{dJ}{g[0]} \\ \vdots \\ \frac{dJ}{g[N_g]} \end{bmatrix}. \tag{17}$$

Least square error optimal filter VI

- By minimization of the criterion we obtain

$$\frac{dJ}{d\mathbf{g}} = -2\mathbf{A}^T\mathbf{x}_\mathrm{R} + 2\mathbf{A}^T\mathbf{A}\mathbf{g} \tag{18}$$

- After setting the derivative equal to zero vector we obtain

$$\mathbf{A}^T\mathbf{A}\mathbf{g} = \mathbf{A}^T\mathbf{x}_\mathrm{R} \tag{19}$$

- The analytic formula for the optimum filter is therefore

$$\mathbf{g} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}_\mathrm{R}. \tag{20}$$

- Matrix $\mathbf{A}^T\mathbf{A}$ of the set of linear equations (19) is toeplitz symmetric.
- Computation of $\mathbf{g}$ can be fastened considerably using Levinson-Durbinovy recursion (complexity $O(N_g^2)$)
- Classical approach to such solution is the Gauss elimination with complexity $O(N_g^3)$.