DSP Coursera II | 12.3.24

5. Discrete-Time Filters

5.1. Linear Time-Invariant Systems

LTI:

- with single input and single output
  ↳ filter

- Operator: transforms input to output

$$y[n] = \mathcal{H}\{x[n]\}$$

$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n]$$

- Linearity

$$\mathcal{H}\{\alpha x_1[n] + \beta x_2[n]\} = \alpha \mathcal{H}\{x_1[n]\} + \beta \mathcal{H}\{x_2[n]\} \qquad \alpha, \beta \in \mathbb{C}$$

- Time invariance

$$y[n] = \mathcal{H}\{x[n]\} \Leftrightarrow \mathcal{H}\{x[n-n_0]\} = y[n-n_0]$$

· LTI systems are completely characterized by impulse-response:

$$\underbrace{h[n]}_{\text{impulse-response}} = \mathcal{H}\{\overset{\text{impulse}}{\delta[n]}\}$$

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

(Reproducing formula)

with $h[n] = \mathcal{H}\{\delta[n]\}$

$$\Rightarrow \quad y[n] = \mathcal{H}\{x[n]\}$$

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

DSP Coursera I | 13.3.24

- Convolution:

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

- Ingredients:
  - sequence $x[n]$
  - second sequence $h[x]$ (i.e. impulse response of LTI)

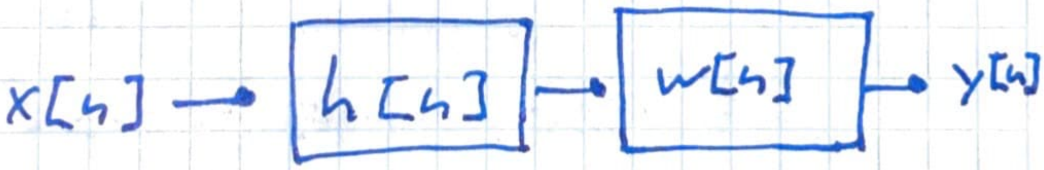
- Algorithm
  - time reverse $h[n]$
  - at each step k (from $-\infty$ to ∞):
    - center time reversed $h[n]$ in k (shift by $-k$)
    - compute inner product with $x[n]$

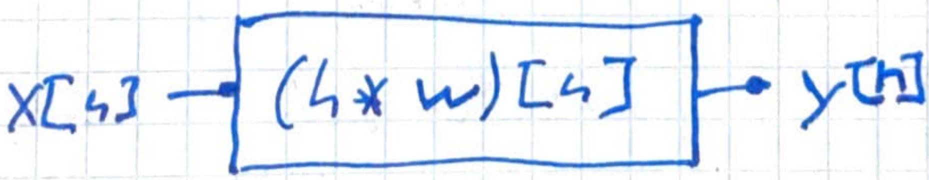
- Convolution properties:
  - linearity and time invariance
    → by definition
  - commutativity:

$$(x * h)[n] = (h * x)[n]$$

  - associativity for absolutely- and square-summable sequences

$$((x * h) * w)[n] = (x * (h * w))[n]$$





DSP Coursera II 15.3.24

5.2.2 Properties of the Impulse Response

LTI System: completely described by impulse response

$$h[n] = \mathcal{H}\{\delta[n]\}$$

FIR vs. IIR:

- impulse response: ∞-length signal
- taps: non-zero values of impulse response
- IIR: number of taps: ∞
- FIR: number of taps: finite
  ↳ impulse response: finite support sequence

Causality:

- causal if output does not depend on future input values
- "real-time" systems always causal
- non causal filters: batch-processing

Stability:

- BIBO stable: bounded-input bounded-output stable
  - ↳ bounded input leads to bounded output
- natural requirement: output will not „blow up" for reasonable input
- linearity and time-invariance do <u>not</u> guarantee stability !
- necessary and sufficient condition for LTI system: impulse response is absolutely summable → stable
- FIR filters are always stable

DSP Coursera II 15.3.24

5.3. Filtering by Example - Time Domain

Filtering: convolution with impulse response

5.3.1. FIR Filtering

- Smoothing sequence, removing noise
- simple approach: replace each point in sequence $x[n]$ by local average
  - local average: sample at n and $N-1$ predecessors

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$

↳ convolution → get impulse response by setting $x[n] = \delta[n]$

$$\hookrightarrow \quad h[n] = \frac{1}{N}\sum_{k=0}^{N-1}\delta[n-k] = \begin{cases} 1/N & \text{for } 0 \le n < N \\ 0 & \text{for } n < 0 \text{ and } n \ge N \end{cases}$$

⇒ finite support impulse response
↳ FIR filter

- Moving Average filter
- introduces delay

5.3.2 IIR Filtering

- leaky integrator

For FIR Filter:

$$y_M[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

↳ $$y_M[n] = \frac{M-1}{M} y_{M-1}[n-1] + \frac{1}{M} x[n]$$

$$= \lambda y_{M-1}[n-1] + (1-\lambda)x[n]$$

with: $\lambda = \frac{M-1}{M}$

For large M $y_{M-1}[n] \approx y[n]$

- average over $M-1$ points is almost equal as average over M points

↳ recursive calculation:

$$y[n] = \lambda y[n-1] + (1-\lambda) x[n]$$

↳ constant coefficient difference equation (CCDE)

- each output value depends on previous output value

- Need to define starting value for $y[n]$: initial condition
  - ↳ zero initial condition:
    - LTI System: impulse response to zero-only input → zero only output
    - $y[n] = 0$ for $n < N_0$
      - N_0: some time in past

- Impulse Response for leaky integrator
  - $x[n] = \delta[n]$
  - $y[n] = \lambda y[n-1] + (1-\lambda)\, x[n]$
  - ↳ $y[n] = 0$ for $n < 0$

    $y[0] = 1-\lambda$

    $y[1] = (1-\lambda)\lambda$

    $y[2] = (1-\lambda)\lambda^2$

    $\vdots$

    $y[n] = (1-\lambda)\lambda^n$

  - ↳ $h[n] = (1-\lambda)\lambda^n u[n]$

    Infinite impulse response (IIR)

DSP Coursera II 19.3.24

- stability of leaky integrator (IIR):
  - sufficient condition for stability:
    impulse response is absolutely summable
    $$\sum_{n=-\infty}^{\infty} |h[n]| = \lim_{n \to \infty} |1-\lambda| \frac{1-|\lambda|^{n+1}}{1-|\lambda|}$$
    ↳ finite for $|\lambda| < 1$
    λ: pole of system

- IIR filter design is less straightforward than FIR filter design

- IIR filter: less flexibility (compared to FIR)

- IIR filter: attractive for efficient implementation (needs less computational power)

5.4. Filtering in the Frequency Domain

Convolution in discrete-time convolution is equivalent to the multiplication of Fourier transforms in the frequency domain.

5.4.1. LTI "Eigenfunctions"

Complex exponential sequence (with frequency ω_0) as input of a LTI-system $\mathcal{H}$

$$\mathcal{H}\{e^{j\omega_0 n}\} = \underbrace{H(e^{j\omega_0})}_{\text{DTFT of } h[n] \text{ at } \omega = \omega_0} e^{j\omega_0 n}$$

$H(e^{j\omega_0})$: frequency response

↳ complex exponentials are „eigensequences" of LTI Systems. (analogous to complex exponential functions beeing eigenfunctions of continous-time LTI Systems)

Shape of input signals cannot be changed by LTI filter.

- Polar form:

$$\mathcal{H}\{e^{j\omega_0 n}\} = A_0 e^{j(\omega_0 n + \Theta_0)}$$

  - Output oscillation scaled by amplitude $A_0 = |H(e^{j\omega_0})|$
  - output shifted by $\Theta_0 = \measuredangle H(e^{j\omega_0})$

LTI systems cannot shift or duplicate frequencies

5.4.2. The Convolution and Modulation Theorems

Absolutely summable sequences $x[n], h[n]$.

Convolution: $y[n] = x[n] * h[n]$

DTFT of convolution:

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

Time Domain	Frequency Domain
$x[n] * y[n]$	$X(e^{j\omega}) \cdot Y(e^{j\omega})$
$x[n] \cdot y[n]$	$X(e^{j\omega}) * Y(e^{j\omega})$

DSP Coursera II 27.3.24

5.4.3. Properties of the Frequency Response

LTI System completely characterized by:

- impulse response
- frequency response

Frequency response: DTFT of impulse response

- Magnitude:
  - Shape magnitude spectrum of filter impulse response
  - boost, attenuate or eliminate part of frequency content
  - 4 categories:
    - assumption: impulse response is real
    - lowpass filters:
      - magnitude of transform concentrated around $\omega=0$
      - preserve low-frequency
      - attenuate high-frequency

- Highpass filters:
  - magnitude of transform concentrated around $\omega = \pm\pi$
  - preserve high frequency
  - attenuate low frequency
- Bandpass filters
  - magnitude of transform concentrated around $\omega = \pm\omega_p$
  - preserve frequencies around ω_p
  - attenuates frequencies elsewhere ($\omega = 0$, $\omega = \pm\pi$)
- Allpass filters:
  - magnitude of transform is constant over entire $[-\pi, \pi]$ intervall
  - can have constant gain
  - phase response important

- Stopband: frequency intervall for which magnitude of frequency response is zero (or negligible)
- Passband: Everywhere else, where frequ. are passed through the filter

DSP Coursera II 28.3.24

- Phase
  - equally important as magnitude
  - less intuitive
  - phase offset → delay in time domain
  - not always possible to express phase offset (in descrete time) as integer number of samples
    → fractional delay
  - for audio applications not very important
    - human ear is mostly insensitive to phase
  - Important for data transmission !
- Linear phase
  - $\measuredangle H(e^{j\omega}) = -\omega d$
  - for general filter with linear phase
    $$H(e^{j\omega}) = |H(e^{j\omega})| e^{-j\omega d}$$
  - just delay no phase distortion

- Group delay:
  - no linear phase
  - measure nonlinearity in phase
    ↳ group delay
  - Phase response around given frequency ω_0 using first-order Taylor expansion:
    - define $\varphi(\omega) = \measuredangle H(e^{j\omega})$
    - approximate $\varphi(\omega)$ around ω_0 as $\varphi(\omega_0 + \tau) = \varphi(\omega_0) + \tau \varphi'(\omega_0)$

    ↳ $H(e^{j(\omega_0+\tau)}) = |H(e^{j(\omega_0+\tau)})| e^{j\varphi(\omega_0+\tau)}$

    $\approx \left(|H(e^{j(\omega_0+\tau)})| e^{j\varphi(\omega_0)} \right) e^{j\varphi'(\omega_0)\tau}$

    - The delay for this group of frequencies is

      the negative of the derivative of the phase

      $\text{grd}\{H(e^{j\omega})\} = -\varphi'(\omega)$

      $= -\dfrac{d \measuredangle H(e^{j\omega})}{d\omega}$

5.6. Ideal Filters

- theoretical abstraction
- cannot be implemented

- Ideal Lowpass
  - "kills" all frequencies above cutoff freq. ω_c
  - leaves all frequencies below ω_c untouched

$$H_{lp}(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| \leq \pi \end{cases}$$

  - 2π periodicity
  - zero phase delay
  - bandwidth: $\omega_b = 2\omega_c$
  - Impulse response

$$h_{lp}[n] = \frac{\sin(\omega_c n)}{\pi n}$$

→ impulse response: symmetric, infinite ⇒ IIR

  - not possible to realize in finite number of operations
  - slow decay of impulse response
  - impulse response not absolutely summable

- ideal lowpass: important theoretical paradigm

↳ special names for funcs:

$$\text{rect}(x) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & |x| > 1/2 \end{cases}$$

$$\text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0 \end{cases}$$

↳ $\cdot$ $H_{lp}(e^{j\omega}) = \text{rect}\left(\frac{\omega}{\omega_b}\right)$

$\cdot$ $h_{lp}[n] = \frac{\omega_b}{2\pi} \text{sinc}\left(\frac{\omega_b}{2\pi} n\right)$

DTFT pair:

$$\frac{\omega_b}{2\pi} \text{sinc}\left(\frac{\omega_b}{2\pi} n\right) \xleftrightarrow{DTFT} \text{rect}\left(\frac{\omega}{\omega_b}\right)$$

DSP Coursera II | 3.4.24

- ideal Highpass
  - complementary to lowpass
  - eliminates frequencies below cutoff frequency ω_c
  - frequency response

$$H_{hp}(e^{j\omega}) = \begin{cases} 0 & |\omega| \leq \omega_0 \\ 1 & \omega_c < |\omega| \leq \pi \end{cases}$$

  - 2π periodicity as usual implicit
  - impulse response

$$h_{hp}[n] = \delta[n] - \frac{\omega_c}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi} n\right)$$

- ideal Bandpass
  - center frequency ω_0
  - bandwidth ω_b, $\frac{\omega_b}{2} < \omega_0$
  - frequency response (between $-\pi$ and π)

$$H_{bp}(e^{j\omega}) = \begin{cases} 1 & \omega_0 - \frac{\omega_b}{2} \leq \omega \leq \omega_0 + \frac{\omega_b}{2} \\ 1 & -\omega_0 - \frac{\omega_b}{2} \geq \omega \geq \omega_0 + \frac{\omega_b}{2} \\ 0 & \text{elsewhere} \end{cases}$$

  - 2π periodicity (as usual implied)

- impulse response

$$h_{bp}[n] = 2\cos(\omega_0 n)\frac{\omega_b}{2\pi}\,\mathrm{sinc}\left(\frac{\omega_b}{2\pi}n\right)$$

- Hilbert Filter
  - frequency response

$$H(e^{j\omega}) = \begin{cases} -j & 0 \leq \omega < \pi \\ +j & -\pi \leq \omega < 0 \end{cases}$$

  - 2π periodicity (as usual implicitly assumed)
  - impulse response

$$h[n] = \frac{2\sin^2(\pi n/2)}{\pi n} = \begin{cases} 0 & \text{for } n \text{ even} \\ \frac{2}{n\pi} & \text{for } n \text{ odd} \end{cases}$$

  - allpass filter: $|H(e^{j\omega})| = 1$
  - phase shift: $\pi/2$
  - used for efficient demodulation schemes

DSP Coursera II   4.4.24

5.7. Realizable Filters

- unlike ideal filters realizable filters can be implemented
- algorithm exists
- compute every output sample with
  - finite numbers of operations
  - finite amount of memory
- Impulse response of realizable filters need not be finite-support: IIR

5.7.1. Constant-Coefficient Difference Equations

- mathematically describing LTI system:
  ↳ „machine" that takes one input sample at a time and produces one output sample
- linearity → only linear operations: sums, multiplications by scalars
- time invariante → scalars are constant
- relizability → · finite number of adders and multipliers
  · finite amount of memory

- such a mathematical relationship ("machine")
  ↳ constant-coefficient difference equation
  CCDE
- most general form:
  - relationship between:
    - input signal $x[n]$ and
    - output signal $y[n]$

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

  - often (also in the book):
    coefficients a_k and b_k are real
  - if $a_0 = 1$ (usually the case)
    ↳ $$y[n] = \sum_{k=0}^{M-1} b'_k x[n-k] - \sum_{k=1}^{N-1} a'_k y[n-k]$$
    ↳ each output sample $y[n]$ is a linear combination of
      - past and present input values
      - past output values
  - can be rearranged to be computed as linear combination of future values of input and output
    ↳ anticausal
- CCDE's can be directly translated to code