Autori: Andra Băltoiu, Silviu Gheorghe, Gabriel Majeri, Rebeca Oprea

Procesarea Semnalelor
Laboratorul 8
Convoluție. Filtre

1 Convoluția

Convoluția reprezintă o operație de compunere a două semnale $f\colon \mathbb{R} \to \mathbb{R}$ și $g\colon \mathbb{R} \to \mathbb{R}$, care produce un nou semnal $f * g\colon \mathbb{R} \to \mathbb{R}$, definit prin expresia

$$(f * g)(t) := \int_{-\infty}^{+\infty} f(x) \cdot g(t - x)\, \mathrm{d}x$$

Convoluția este o operație **asociativă** și **comutativă**, cu alte cuvinte

$$(f * g) * h = f * (g * h)$$
$$f * g = g * f$$

pentru orice semnale $f, g, h\colon \mathbb{R} \to \mathbb{R}$.

Operația este des utilizată în prelucrarea semnalelor și în statistică, unde este interpretată ca o medie ponderată a valorilor din trecut.

Puteți înțelege mai bine conceptul urmărind acest video sau jucându-vă cu exemplele interactive de pe această pagină.

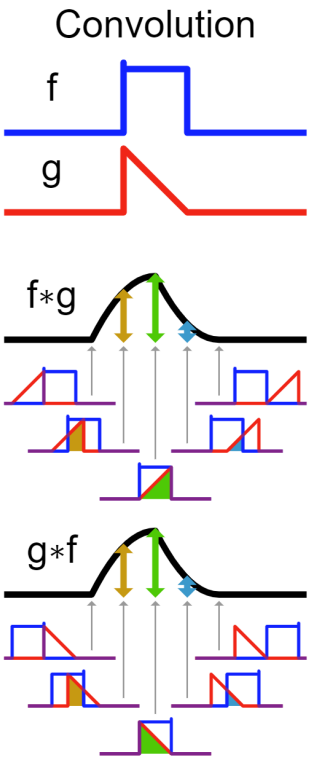

Figura 1: Convoluția a două semnale continue

Convoluția și **produsul** (element cu element) a două semnale sunt operații *duale* în domeniul timp și frecvență: convoluția în timp este echivalentă produsului în domeniul frecvență, produsul în domeniul timp este echivalent convoluției în domeniu frecvență.

Dacă notăm operatorul corespunzător transformatei Fourier cu $\mathscr{F}$, enunțul de mai sus, ce reprezintă *Teorema de Convoluție*, înseamnă

$$\mathscr{F}(f * g) = \mathscr{F}(f) \cdot \mathscr{F}(g)$$
$$\mathscr{F}(f \cdot g) = \mathscr{F}(f) * \mathscr{F}(g)$$

Convoluția în frecvență. Ferestre

Când achiziționăm un semnal periodic pe o durată anume, cel mai adesea durata de timp **nu** reprezintă un multiplu întreg al perioadei semnalului. Din acest motiv, varianta finită pe care o avem la dispoziție diferă de semnalul original (și de timp continuu). O importanță deosebită o au **tranzițiile abrupte** ce pot apărea datorită acestei diferențe. În special capetele intervalului măsurat reprezintă astfel de discontinuități.

Atunci când se aplică transformata Fourier pentru a afla și vizualiza spectrul semnalului, discontinuitățile vor apărea sub forma unor componente de frecvență care în semnalul original nu sunt prezente și vor produce fenomenul de ***leakage***.

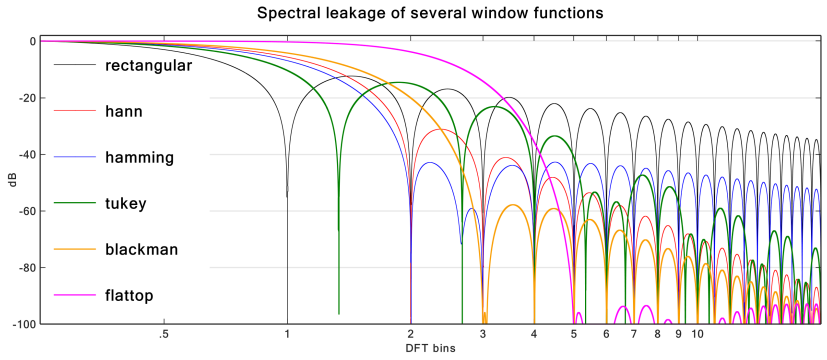

Figura 2: Diferite tipuri de ferestre și efectul de *leakage*

Sursa imaginii

În practică, unde semnalele sunt funcții cu multe componente de frecvență, fenomenul apare des, pentru că de obicei nu avem de-a face cu un semnal ce conține un număr întreg de perioade. Însă fenomenul se poate atenua utilizând diferite tipuri de **ferestre**

pentru a selecta un interval de timp dintr-un semnal, ce atenuează discontinuitățile de la capete.

Un semnal trecut printr-o fereastră se poate exprima

$$x_w[n] = x[n] \times w[n]$$

unde cu $w[n]$ am notat fereastra.

Acestui produs în domeniul timp îi corespunde operația de convoluție în domeniul frecvență. Frecvențele înalte datorate trecerii bruște la valoarea 0 vor fi atenuate de fereastră. Așadar, pentru a evita fenomenul de *leakage*, fereastra **nu** trebuie să aibă **discontinuități** pronunțate. Cu alte cuvinte, cu cât fronturile sunt **mai netede**, cu atât lobii secundari vor fi **mai atenuați**.

Reducerea fenomenului de *leakage* presupune ca lățimea lobului principal să fie cât mai mică, la fel și vârful lobilor secundari. Un alt criteriu în alegerea ferestrei este rata cu care lobii secundari descresc.

Exemple de ferestre

În general, diferite tipuri de semnale se pretează la diferite tipuri de ferestre. Spre exemplu, dacă nu se cunoaște nimic despre componentele semnalului, o fereastră potrivită este cea **de tip Hanning**. Aceasta e utilă de asemenea dacă semnalul este format din două sinusoide. Dacă însă sinusoidele sunt foarte apropiate, este mai potrivită o **fereastră uniformă** sau o **fereastră Hamming**.

Puteți vedea o listă cu diferite tipuri de ferestre și formulele care le definesc în Tabela 1.

Convoluția în timp. Filtre

Notăm cu x și h două semnale în timp. Operația de convoluție în cazul discret este dată de formula

$$y[n] = \sum_{k=0}^{M-1} h[k] \cdot x[n-k]$$

unde M este lungimea ferestrei. Convoluția presupune o inversare a axei timpului, urmată de o deplasare (*shift*) a coeficienților și o sumă de produse.

Tabela 1: Ferestre uzuale

Tip Fereastra	**Formula**
Dreptunghiulară	$w(n) = 1$
Hanning	$w(n) = 0.5\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right]$
Hamming	$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right)$
Blackman	$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right)$
Flat top	$w(n) = 0.22 - 0.42\cos\left(\frac{2\pi n}{N}\right) + 0.28\cos\left(\frac{4\pi n}{N}\right) - 0.08\cos\left(\frac{6\pi n}{N}\right) + 0.007\cos\left(\frac{8\pi n}{N}\right)$

Adesea unul din cele două semnale poate fi văzut ca un **filtru** și reprezintă un sistem liniar, invariant în timp.

Media alunecătoare (*rolling mean*) este unul din cele mai comune tipuri de filtre. Semnalul din Figura 3 reprezintă date din trafic, mai exact numărul de vehicule care trec printr-o intersecție la un moment de timp (setul de date din laboratorul precedent). Acestea au fost filtrate cu filtre medie alunecătoare având diferite dimensiuni ale ferestrei, N_w. Observați cum semnalul este netezit în mod diferit și întârzierile provocate de dimensiunea ferestrei.

2 Proiectarea Filtrelor

Proiectarea unui filtru se referă la specificarea coeficienților acestuia astfel încât să aibă un anumit răspuns în frecvență. Deoarece un filtru ideal nu poate fi implementat fizic (este necauzal și are suport infinit), în practică se caută un răspuns în frecvență cu toleranțe fixate.

Proiectarea Filtrelor FIR

Cea mai simplă metodă de proiectare a **filtrelor cu suport finit (FIR)**, utilizată în special când specificațiile nu sunt foarte precise, este *metoda ferestrei*. Aceasta presupune modularea în timp a răspunsului ideal cu o fereastră.

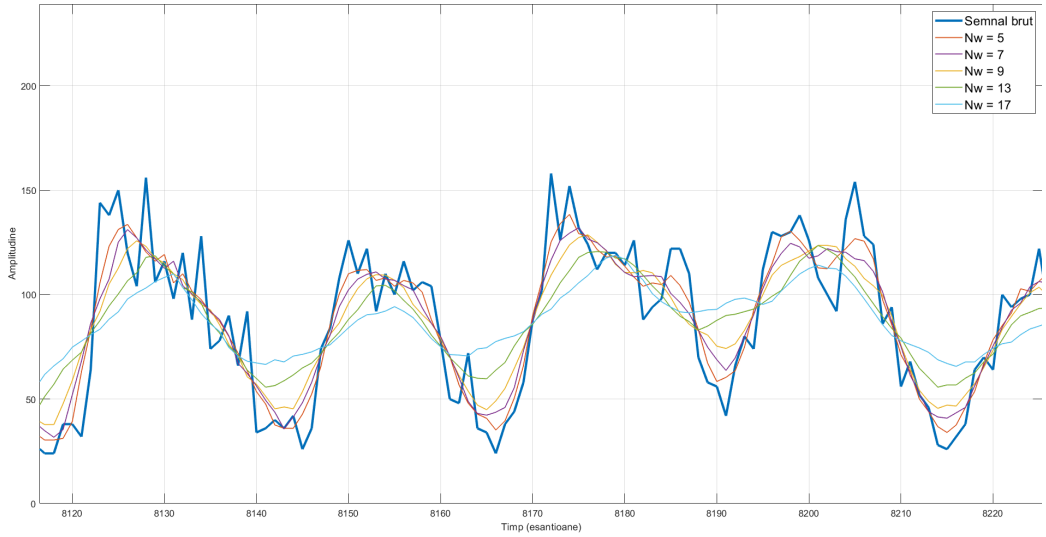

Figura 3: Filtru medie alunecătoare

Răspunsul ideal al unui filtru trece-jos este

$$D(\omega) = \begin{cases} 1, & \omega \in [0, \omega_t) \\ 0, & \omega \in [\omega_t, \pi] \end{cases} \tag{1}$$

unde ω_t reprezintă frecvența de tăiere. Însă în realitate un filtru nu va putea tăia perfect la ω_t, de aceea se introduce noțiunea de bandă de trecere $[0, \omega_b]$ și bandă de oprire $[\omega_s, \pi]$. Între cele două există banda de tranziție.

Performanțele filtrului pot fi schimbate modificând ordinul filtrului sau tipul ferestrei. Un filtru optim are ordin minim.

Ideal, răspunsul în frecvență al ferestrei trebuie să se apropie cât mai mult de **impulsul unitate**. Însă această cerință nu se poate realiza datorită **incertitudinii** de localizare în timp și frecvență: fereastra nu poate avea în același timp și suport finit și spectru concentrat.

Trunchierea răspunsului cu ajutorul ferestrei provoacă apariția **fenomenului Gibbs**, în care răspunsul în frecvență prezintă oscilații în apropierea frecvențelor de tranziție.

Proiectarea Filtrelor IIR

O altă familie este reprezentată de **filtrele cu suport infinit (IIR)**.

Proiectarea filtrelor presupune de obicei o serie de compromisuri, spre exemplu, între lățimea lobului principal și înălțimea lobilor secundari (corespunzători benzii de tranziție). Un alt compromis se datorează faptului că unui răspuns cu atenuare mare în banda de trecere îi corespunde o bandă de tranziție de asemenea mare, în timp ce unei benzi de tranziție mică îi corespunde și o atenuare mică. Un astfel de compromis poate fi ilustrat analizând două filtre des utilizate în practică: **Butterworth** și **Chebyshev**.

Filtrul Butterworth are un răspuns plat în banda de trecere (fără ondulații), în schimb compensează cu o tranziție foarte lentă. Din acest motiv este util acolo unde este necesar ca semnalul să nu fie deloc distorsionat de operația de filtrare, spre exemplu ca procedură de anti-aliere sau în aplicații audio.

Filtrul Chebyshev se folosește acolo unde componentele de frecvență sunt mai importante decât amplitudinea semnalului.

Figura 4 reprezintă date de trafic care au fost filtrate pentru a elimina frecvențele înalte utilizând două filtre diferite de ordin 5 și aceeași frecvență de tăiere: Butterworth și Chebyshev.

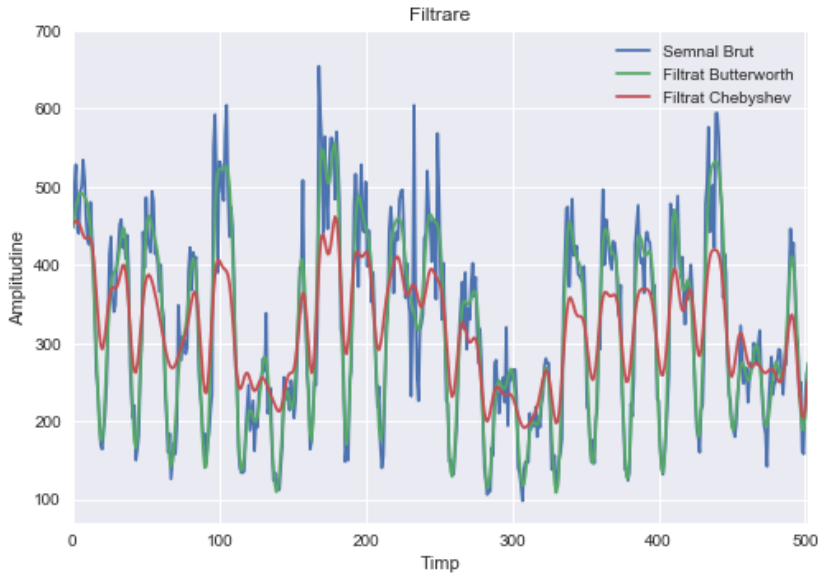

Figura 4: Filtrarea unui semnal cu Filtre Butterworth și Chebyshev

3 Ghid Python

Pentru proiectarea filtrelor veți avea nevoie de biblioteca `scipy`.

De regulă, funcțiile care implementează filtrele returnează **coeficienții** acestora, anume vectorii `b`, `a` reprezentând coeficienții polinoamelor de la numărător respectiv numitor. Alternativ, se poate opta pentru reprezentarea cu poli și zerouri.

O dată obținuți coeficienții filtrului, un semnal `x` se poate **filtra** utilizând funcția `scipy.signal.filtfilt(b, a, x)`.

Pentru a calcula **răspunsul în frecvență** al unui filtru, se poate utiliza funcția `scipy.signal.freqz(b, a)`. Aceasta returnează un vector `w` cu frecvențele pentru care este calculat răspunsul și un vector `h` de numere complexe, reprezentând răspunsul în frecvență. Când afișați grafic modulul acestora, folosiți scala logaritmică, anume

```
plt.plot(w, 20 * np.log10(np.abs(h)))
```

Proiectarea unui **filtru Butterworth** se poate face cu ajutorul funcției

```
scipy.signal.butter(N, Wn, btype='low')
```

Primul parametru, `N`, se referă la ordinul filtrului. Parametrul `Wn` se referă la frecvențele de tăiere. În cazul filtrelor trece-jos sau trece-sus, `Wn` este un scalar, iar în cazul filtrelor trece-bandă, un vector de 2 elemente, ce conține capetele benzii. Aceste valori sunt normalizate în $[0, 1]$, unde 1 este frecvența Nyquist. Parametrul `btype` specifică tipul filtrului: trece-jos (*low-pass*) ș.a.m.d.

Proiectarea unui **filtru Chebyshev** se poate face folosind

```
scipy.signal.cheby1(N, rp, Wn, btype='low')
```

Parametrul `rp` controlează atenuarea ondulațiilor în banda de trecere, în dB.

Pentru toate funcțiile de mai sus, puteți găsi în documentație lista completă a parametrilor.

4 Exerciții

1. Scrieți câte o funcție prin care să construiți o **fereastră dreptunghiulară**, respectiv o **fereastră de tip Hanning**. Funcțiile primesc ca parametru **lungimea ferestrei**. 1p

   Generați și afișați o sinusoidă cu frecvență $f = 100\,\text{Hz}$, amplitudine unitară și fază nulă, apoi treceți-o prin cele două tipuri de ferestre (pentru lungimea $N_w = 200$) și afișați grafic rezultatul.

2. Fișierul `traffic-one-week.csv`[1] conține date de trafic colectate pe o perioadă de o săptămână. Perioada de eșantionare este de o oră, iar valorile măsurate reprezintă numărul de vehicule care trec printr-o intersecție.

   (a) Încărcați datele în memorie și selectați din semnalul dat o porțiune corespunzătoare pentru **3 zile**, `x`, pe care veți lucra în continuare.

   (b) Utilizați funcția `np.convolve(x, np.ones(w), 'valid') / w` pentru a realiza un filtru de tip medie alunecătoare și neteziți semnalul obținut anterior. Setați dimensiuni diferite ale ferestrei (variabila `w` în codul de mai sus), spre exemplu 5, 9, 13, 17. Afișați grafic semnalele obținute. 1p

   (c) Dorințiți să filtrați **zgomotul** (frecvențe înalte) din semnalul cu date de trafic. Alegeți o **frecvență de tăiere** pentru un filtru trece-jos pe care îl veți crea în subpunctul următor. Argumentați alegerea făcută.

   Indicație: transformata Fourier a semnalului, calculată în laboratorul anterior, ar putea să vă ajute cu luarea acestei decizii.

   Pe lângă valoarea frecvenței în Herzi, determinați și valoarea ei normalizată între 0 și 1, unde 1 reprezintă frecvența Nyquist pentru semnalul de trafic. 1p

   (d) Utilizați funcția `scipy.signal.butter`, respectiv funcția `scipy.signal.cheby1`, pentru a **proiecta** filtrele Butterworth și Chebyshev de ordin 5, cu frecvența de tăiere W_n (valoare normalizată), stabilită la subpunctul anterior.

[1] Sursa originală a datelor se poate găsi aici.

Pentru început, setați atenuarea ondulațiilor la `rp = 5` (dB), urmând ca apoi să încercați și alte valori. 1p

(e) **Filtrați** datele de trafic cu cele 2 filtre și afișați semnalele filtrate împreună cu datele brute. Ce filtru alegeți dintre cele două și de ce? 1p

(f) Reproiectați filtrele alegând atât un ordin mai mic, cât și unul mai mare. De asemenea, reproiectați filtrul Chebyshev cu alte valori ale `rp` și observați efectul.

Stabiliți **valorile optime** ale parametrilor pentru a vă atinge scopul (cel de a filtra zgomotul din semnal). 1p