Procesarea Semnalelor
Laboratorul 6.
Convoluție. Filtre

1 Convoluția

Convoluția reprezintă o operație asociativă și comutativă de compunere a două semnale

$$f * g = g * f \tag{1}$$

Operația este des utilizată în prelucrarea semnalelor și în statistică, unde este interpretată ca o medie ponderată a valorilor din trecut. Dacă întâmpinați dificultăți în înțelegerea conceptului, [1] prezintă o vizualizare grafică a operației de convoluție.

Convoluția și produsul (la nivel de element) a două semnale sunt operații inverse în domeniul timp și frecvență: convoluția în timp este echivalentă produsului în domeniul frecvență, produsul a două semnale în domeniul timp este echivalent convoluției celor două semnale în domeniu frecvență. Notând operatorul transformatei Fourier cu $\mathcal{F}$, enunțul de mai sus, ce reprezintă Teorema de Convoluție, se scrie

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g) \tag{2}$$
$$\mathcal{F}(f \cdot g) = \mathcal{F}(f) * \mathcal{F}(g) \tag{3}$$

1.1 Convoluția în frecvență. Ferestre

Atunci 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 ce în semnalul original nu sunt prezente și vor produce fenomenul de *leakage*. În practică, unde semnalele sunt funcții cu multe componente, fenomenul apare des, pentru că adesea nu avem de-a face cu un semnal ce conține un număr

Tabela 1: Ferestre uzuale

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

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

$$x_w[n] = x[n] \times w[n] \tag{4}$$

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

Acestui produs în domeniul timp îi corespunde operația de convoluție în domeniul frecvență.

Ceea ce înseamnă că frecvențele înalte datorate trecerii bruște din 0 vor fi atenuate de fereastrei îi corespund puteri mici ale acestor frecvențe. 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 sunt mai atenuați.

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

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 e sinusoidele sunt foarte apropiate, este mai potrivită o fereastră uniformă sau o fereastră Hamming.

1.2 Convoluția în timp. Filtre

Notăm cu $x(t)$ și $h(t)$ două semnale în timp. Operația de convoluție între cele două este

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \tag{5}$$

Iar în cazul discret

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{6}$$

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

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

Media alunecătoare este unul din cele mai comune tipuri de filtre. Semnalul din Figura 1 reprezintă date din trafic, mai exact numărul de vehicule care circulă printr-o locație la un moment de timp [2]. 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.

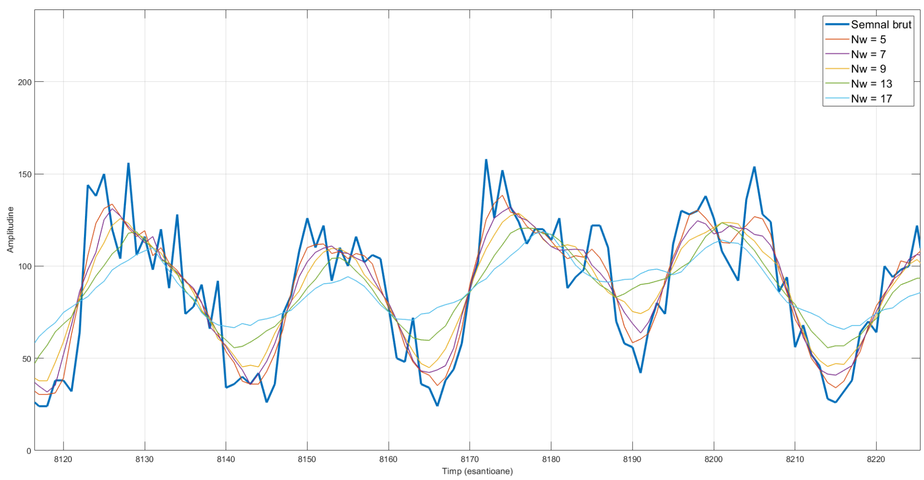

Figura 1: Filtru medie alunecătoare

2 Proiectarea Filtrelor

Proiectarea unui filtru se referă la specificarea coeficienților acestuia astfel încât să aibă un răspuns în frecvență impus. 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.

2.1 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.

Răspunsul ideal al unui filtru trece-jos este

$$D(\omega) = \begin{cases} 1 & , \omega \in [0, \omega_t] \\ 0 & , \omega \in [\omega_t, \pi] \end{cases} \tag{7}$$

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 (M) 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.

2.2 Proiectarea Filtrelor 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 Cebyshev.

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 Cebyshev se folosește, însă, acolo unde mai importante decât amplitudinea semnalului sunt componentele de frecvență.

Figura 2 reprezintă date de trafic auto, care au fost filtrat pentru a elimina frecvențele înalte utilizând două filtre diferite de ordin 5 și aceeași frecvență de tăiere: Butterworth și Chebyshev.

3 Ghid Python

Pentru proiectarea filtrelor este necesar să importați biblioteca `scipy`. De regulă, funcțiile care implementează filtrele returnează coeficienții acestora, anume vectorii `b,a` reprezentând polinoamele de la numărător respectiv numitor. Alternativ, se poate opta pentru reprezentarea poli-zerouri.

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

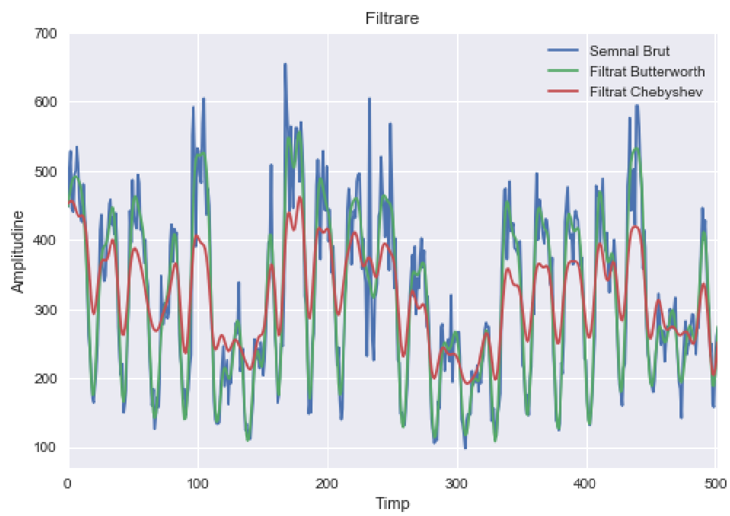

Figura 2: Filtrarea unui semnal cu Filtre Butterworth și Chebyshev

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, folosiți scala logaritmică, anume `plot(w, 20 * np.log10(abs(h)))`.

Proiectarea unui filtru Butterworth se face cu ajutorul funcției `scipy.signal.butter(N, Wn, btype='low')`.

Primul parametru, `N`, se referă la ordinul filtrului. `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, ș.a.m.d).

Proiectarea unui filtru Chebyshev se face folosind `scipy.signal.cheby1(N, rp, Wn, btype='low')`.

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

Pentru ambele funcții de mai sus, căutați în documentație lista completă a parametrilor.

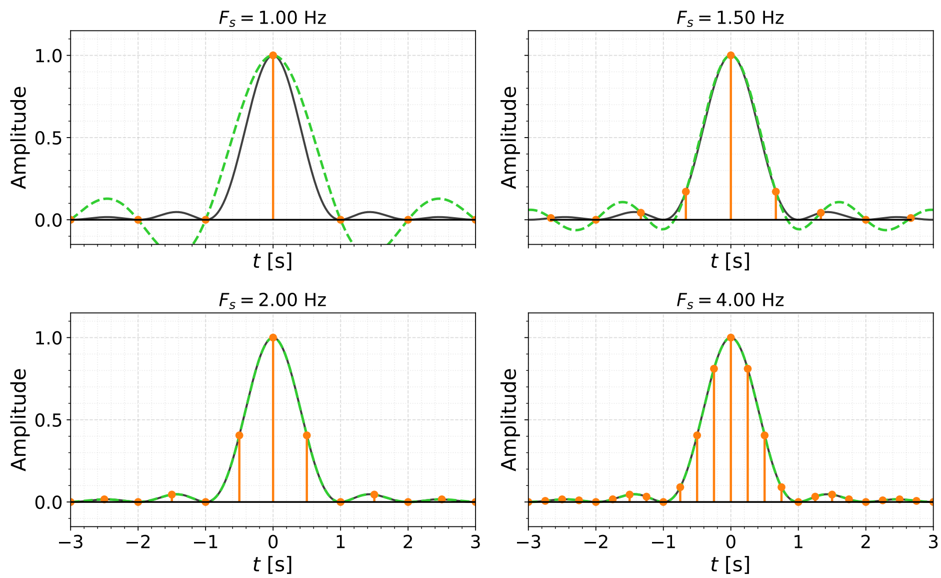

Figura 3: Funcția $sinc^2(t)$, reconstrucția sa și puncte de eșantionare.

4 Exerciții

1. Reproduceti desenele din Figura 3 urmând pașii:
   - Generați într-o figură funcția $x(t) = \text{sinc}^2(Bt)$, pentru $B = 1$, în intervalul $[-3, 3]$. Folosiți *numpy.sinc*.
   - Eșantionați funcția de la sus cu $1, 1.5, 2$, și 4 Hz, notați cu T_s perioada de eșantionare. Punctul $t = 0$ este eșantionat mereu. Desenați punctele eșantionate pe grafic $x[n]$.
   - Pe fiecare plot, suprascrieți peste funcția originală cea reconstruită $\hat{x}(t) = \sum_n x[n]\text{sinc}\left(\frac{t-nT_s}{T_s}\right)$.
   - Schimbați valoarea lui B și observați ce se întâmplă.

2. Generați un vector $x[n]$ aleator de dimensiune $N = 100$. Calculați iterația $x \leftarrow x * x$ de trei ori. Afișați cele patru grafice în aceeași figura, unul sub altul. Ce observați? Repetați experimentul dar cu un semnal bloc rectangular (unu pe un interval scurt și apoi zero în rest).

3. Vi se dau două polinoame $p(x)$ și $q(x)$ cu grad maxim N generate aleator cu coeficienți întregi. Calculați produsul lor $r(x) = p(x)q(x)$ folosing convoluția: folosind înmulțirea polinoamelor directă și apoi folosind *fft*.

4. Generați un vector aleator $\mathbf{x}$ de dimensiune $n = 20$ care să conțină o funcție eșantionată aleasă de voi. Alegeți un număr întreg d și deplasați

circular vector cu d poziții, numiți noul vector $\mathbf{y}$. Folosind vectorul original $\mathbf{x}$ și vectorul deplasat $\mathbf{y}$ recuperați deplasarea d folosind formula $\mathrm{IFFT}(\overline{\mathrm{FFT}(\mathbf{x})} \cdot \mathrm{FFT}(\mathbf{y}))$ (aceasta se numește teorema de deplasare circulară). Folosiți și formula $\mathrm{IFFT}(\mathrm{FFT}(\mathbf{y}) \oslash \mathrm{FFT}(\mathbf{x}))$. Care este diferența dintre cele două? (Aici $\cdot$ și $\oslash$ sunt înmulțirea și împărțirea între vectori, element cu element.)

5. Scrieți câte o funcție prin care să construiți o fereastră dreptunghiulară și o fereastră de tip Hanning. Funcțiile primesc ca parametru dimensiunea ferestrei. Afișați grafic o sinusoidă cu $f = 100$, $A = 1$ și $\varphi = 0$ trecută prin cele două tipuri de ferestre de dimensiune $N_w = 200$.

6. Fișierul `Train.csv` [2] conține date de trafic înregistrate pe o perioadă de 1 săptămână. Perioada de eșantionare este de 1 oră, iar valorile măsurate reprezintă numărul de vehicule ce trec printr-o anumită locație.

   (a) 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 $5, 9, 13, 17$.

   (c) Dorind 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 continuare. Argumentați. Care este valoarea frecvenței în Hz și care este valoarea frecvenței normalizate între 0 și 1, unde 1 reprezintă frecvența Nyquist?

   (d) Utilizând funcțiile și `scipy.signal.butter` și `scipy.signal.cheby1` proiectați filtrele Butterworth și Chebyshev de ordin 5, cu frecvența de tăiere W_n stabilită mai sus. Pentru început setați atenuarea ondulațiilor, rp = 5 dB, urmând ca apoi să încercați și alte valori.

   (e) Filtrați datele de trafic cu cele 2 filtre și afișați semnalele filtrate împreună cu datele brute. Ce filtru alegeți din cele 2 și de ce?

   (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 încercați pentru a vă atinge scopul.

Bibliografie

[1] Swarthmore College. Convolution demo and visualization. https://lpsa.swarthmore.edu/Convolution/CI.html.

[2] Bullet train timeseries data. Time series modelling - predicting traffic growth. https://www.kaggle.com/datasets/lampubhutia/bullettrain-timeseries-data?select=Train.csv.