

Prelucrarea Semnalelor

Laboratorul 6.

Proiectarea Filtrelor

1 Noțiuni introductive

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.

1.1 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

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

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 (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.

1.2 Proiectarea Filtrelor IIR

Proiectarea filtrelor presupune de obicei o serie de compromisuri. Spre exemplu, între lăţimea lobului principal (corespunzător) şi înălţimea lobilor secundari (corespunzători benzii de tranziţie). Un alt compromis se poate datora faptului că unui răspuns cu atenuare mare în banda de trecere are banda de tranziţie de asemenea mare, în timp ce unei benzi de tranziţie mică îi corespunde şi o atenuare mică.

Un astfel de compromis poate fi ilustrat analizând două filtre des utilizate în practică, Butterworth şi Cebyshev.

Filtrul Butterworth are un răspuns plat în banda de trecere (fără ondulaţii), în schimb compensează cu o tranziţie foarte lentă. Din acest motiv este util acolo unde este necesar ca semnalul să nu fie deloc distorsionat de operaţia de filtrare, spre exemplu ca procedură de anti-aliere sau în aplicaţii audio. Filtrul Cebyshev se foloseşte, însă, acolo unde mai importante decât amplitudinea semnalului sunt componentele de frecvenţă.

2 Ghid Python

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

O dată obţinuţi coeficienţii filtrului, un semnal x se poate filtra utilizând funcţia `scipy.signal.filtfilt(b, a, x)`.

Pentru a calcula răspunsul în frecvenţă al unui filtru, se poate utiliza funcţia `scipy.signal.freqz(b,a)`. Aceasta returnează un vector `w` cu frecvenţele pentru care este calculat răspunsul şi un vector `h` de numere complexe, reprezentând răspunsul în frecvenţă. Când afişaţi grafic, folosiţi scala logaritmică, anume `plot(w, 20 * np.log10(abs(h)))`.

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

Primul parametru, `N`, se referă la ordinul filtrului. `Wn` se referă la frecvenţele 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 conţine capetele benzii. Aceste valori sunt normalizate în $[0, 1]$, unde 1 este frecvenţa Nyquist. Parametrul `btype` specifică tipul filtrului.

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

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

Pentru ambele funcţii de mai sus, căutaţi în documentaţie lista completă a parametrilor.

3 Exerciții

1. Implementați și trasați caracteristicile în frecvență ale ferestrelor prezentate în laboratorul anterior.
2. Încărcați fișierul `traffic.csv`, ce conține date de trafic eșantionate la o oră.
 - (a) Calculați DFT și afișați componentele de frecvență.
 - (b) Dorind să filtrați zgomotul (frecvențe înalte), 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?
 - (c) 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 undulațiilor, $rp = 5$ DB, urmând ca apoi să încercați și alte valori.
 - (d) Folosiți funcția `scipy.signal.freqz` pentru a calcula răspunsul în frecvență al filtrelor și afișați-l grafic.
 - (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, reprojetați filtrul Chebyshev cu alte valori ale rp și observați efectul. Stabiliți valorile optime ale parametrilor încercați pentru a vă atinge scopul.