Laboratorul 04.

Semnale în domeniul frecvență

Materiale ajutătoare:

1. D. Johnson's book [http://www.ece.rice.edu/~dhj/courses/elec241/col10040.pdf]
    a. Secțiunile 4.5 (Exercițiul 1), 4.7 (Exercițiul 2), 4.6 (Exercițiul 3)

Exercițiul 1 - aproximare de semnale [6p]

Orice semnal este format dintr-o sumă a o infinitate de sinusoide complexe. Ce se întâmplă dacă facem suma doar peste un număr finit de astfel de termeni, ignorând termenii de ordin superior? În acest caz vom forma un semnal care aproximează semanlul original, iar aproximarea este cu atât mai bună cu cât folosim mai mulți termeni.

În acest exercițiu vom vedea cât de bine este aproximat un semnal, prin observarea erorii ϵ_N dintre semnalul original $s(t)$ și aproximarea $s_N(t)$ folosind doar termeni de ordin $\leq N$ din seria Fourier (termenii corespunzători $k \in \{-N, \ldots, N\}$)

Vom calcula rădăcina pătratică medie (eng. root mean square - RMS) a semnalului de eroare ϵ_N dat de:

$$\epsilon_N(t) = s(t) - s_N(t),$$

Aproximarea $s_N(t)$ este dată de primii termeni din Seria Fourier.

$$s_N(t) = \sum_{k=-N}^{N} c_k e^{\frac{j2\pi kt}{T}}$$

iar RMS-ul lui ϵ_N este dat de:

$$\mathrm{rms}(\epsilon_N) = \sqrt{\sum_{k=-\infty}^{-N+1} |c_k|^2 + \sum_{k=N+1}^{\infty} |c_k|^2} = \sqrt{2 \cdot \sum_{k=N+1}^{\infty} |c_k|^2}$$

Task-ul vostru este să determinați valoarea lui N astfel încât $\mathrm{rms}(\epsilon_N)$ este aproape 0 și după aceea să vedeți că într-adevăr semnalul reconstruit aproximează bine semnalul original.

O să folosim din nou semnalul dreptunghiular, de amplitudine 'A' în intervalul [0, T/2] și '-A' în intervalul [T/2, T] care are coeficienții Fourier dați de formula:

$$c_k = \begin{cases} \frac{2}{j\pi k} A & k \quad impar \\ 0 & k \quad par \end{cases}$$

Pentru asta ar trebui să urmăriți următorii pași:

1. Creați semnalul original. Folosiți, de exemplu T=100 și A=1 pentru a genera semnalul cu valoarea 1 în primele 50 de eșantioane și -1 în ultimele 50. Reprezentaţi grafic semnalul ca funcţie de timp (unde timpul începe de la 1 până la 100) [1p]. Puteți să vă folosiți de acest cod:

```
A = 1;
T = 100;
x = 1:T;
```

```
s = -A*ones(1, T);
s(1:(T/2)) = A;
```

2. Calculați coeficienții Fourier c_k pentru $k = [0 : 500]$. Plotați amplitudinile $|c_k|^2$ ca funcție de k. [1p]
3. Calculați $\mathrm{rms}(\epsilon_N)$ pentru fiecare $N \in \{1, \ldots, 500\}$. Vedeți explicația de mai jos pentru a putea calcula RMS folosind Teorema lui Parseval. Plotați (cu plot, semilogy și loglog) valoarea rms pentru $N \in \{1, \ldots, 500\}$.[3p]
4. Determinați cel mai mic N astfel încât $\mathrm{rms}(\epsilon_N) < 0.05$ și reconstruiți semnalul original folosind acest număr de coeficienți. Trebuie să folosiți atât coeficienții pozitivi cât și negativi (de ex. de la -N la N) pentru a reconstrui semnalul. Reconstruiţi semnalul doar cu ajutorul acestor coeficienţi, folosind formula: $s_N(t) = \sum_{k=-N}^{N} c_k e^{j\frac{2\pi kt}{T}}$. Reprezentați grafic semnalul reconstruit și comparați-l cu semnalul inițial. [1p]

Semnalele reale au următoarea proprietate: coeficienții Fourier negativi sunt conjugații complecși ai celor pozitivi $c_{-k} = c_k^*$. De asemenea, semnalele impare s(-t) = -s(t), au coeficienții complet imaginari, obținând $c_{-k} = -c_k$

Pentru a calcula RMS al erorii trebuie să calculăm suma pentru toți coeficienții c_k cu $|k| > k_0$, adică o infinitate de termeni. Putem încerca doar să aproximăm această sumă, sau ne putem folosi de unele proprietăți ale seriei Fourier pentru a o calcula exact. Mai precis, vom folosi Teorema lui Parseval prin care putem calcula puterea unui semnal în două feluri, în domeniul timp, integrând semnalul la pătrat peste o perioadă sau în frecvență calculând suma pătratelor modulului ale fiecărui coeficient:

$$\frac{1}{T}\int_0^T s^2(t)dt = \sum_{-\infty}^{\infty} |c_k|^2$$

Folosind cele descrise mai sus puteți calcula exact RMS-ul erorii:

1. calculați puterea totală calculând integrala din teorema lui Parseval pentru semnalul dreptunghiular.
2. scădeți pătratele termenilor de la $c_k, k \in \{-N \ldots N\}$ obținând suma termenilor necesari pentru calcularea RMS.

$$\mathrm{rms}(\epsilon_{N-1}) = \sqrt{2 \cdot \sum_{k=N}^{\infty} |c_k|^2} = \sqrt{\sum_{k=-\infty}^{\infty} |c_k|^2 - \sum_{k=-N+1}^{N-1} |c_k|^2} = \sqrt{\sum_{k=-\infty}^{\infty} |c_k|^2 - (2 \cdot \sum_{k=1}^{N} |c_k|^2 + |c_0|^2)}$$

Exercițiul 2 - filtrare [4p]

Vom vedea în continuare ce efect are un filtru trece-jos asupra unui semnal. În special ne interesează care sunt termenii din seria Fourier a semnalului rezultat în urma filtrării.

Dacă știm funcția de transfer a unui filtru trece-jos (sau alt tip de sistem liniar), care primește la intrare un semnal (pentru care putem găsi coeficienții Fourier c_k), la curs am arătat că putem găsi coeficienții Fourier (c_k^y) ai semnalului rezultat ca:

$$c_k^y = H(\frac{k}{T}) \cdot c_k$$

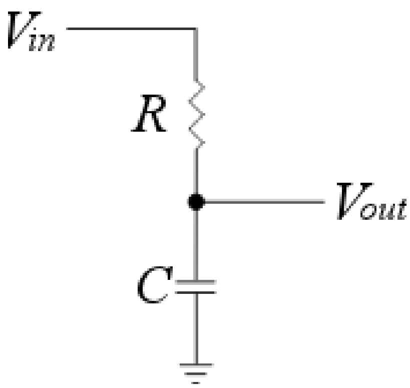
Astfel putem reconstrui semnalul de ieșire folosind coeficienții Fourier c_k^y.

În acest exercițiu trebuie să calculăm coeficienții Fourier ai output-ului unui filtru trece-jos, dat fiind un semnal de intrare de tip puls cu amplitudine $A = 1$ și pulsul de durată $\Delta = \frac{T}{5}$. Știm că coeficienții Fourier ai semnalului sunt dați de:

$$c_k = A \cdot e^{-j\frac{\pi k \Delta}{T}} \cdot \frac{\sin(\frac{\pi k \Delta}{T})}{\pi k} = A \cdot e^{-j\frac{\pi k \Delta}{T}} \cdot \frac{\Delta}{T} \cdot \text{sinc}(\frac{\pi k \Delta}{T}).$$

Atenție: Octave folosește funcția sinc normalizată sinc(x) = $\frac{\sin(\pi x)}{\pi x}$

Puteți vedea circuitul filtrului trece-jos în următoarea imagine:



Funcția de transfer a circuitului (pe care am determinat-o la curs și la laboratoarele anterioare) este următoarea:

$$H(f = \frac{k}{T}) = \frac{1}{1 + j2\pi RC\frac{k}{T}}$$

unde R și C sunt rezistența și respectiv capacitatea.

Task-ul vostru este să determinați coeficienții output-ului și să reconstruiți semnalul de ieșire pentru diferite frecvențe de tăiere.

Pentru aceasta urmăriți următorii pași:

1. Generați semnalul puls și plotați-l. Puteți folosi T=100 de eșantioane (puncte), dintre care doar Δ nu sunt egale cu 0. [1p]
2. Calculați primii N=30 coeficienți Fourier pozitivi c_k ai semnalului și plotați-i. Pentru a îi reprezenta va trebui să folosiţi funcţia 'stem'. De asemenea, va trebui să reprezentaţi doar magnitudinea, folosind funcţia 'abs'. [1p]
3. Calculați coeficienții Fourier asociați semnalului de output, c_k^y, folosind formula de mai sus și plotați-i ca mai sus. Pentru asta va trebui să alegeți o frecvență de cut-off f_c care va determina valorile R și C ($RC = \frac{1}{2\pi f_c}$). [1p]

Puteți încerca următoarele valori pentru f_c:

1. $f_c = 0.1/T$ (frecvența de cut-off e mult mai mică decât frecvența fundamentală a semnalului ⇒ filtrare puternică)
2. $f_c = 1/T$ (frecvența de cut-off = frecvența fundamentală ⇒ puterea este înjumătățită)
3. $f_c = 10/T$ (frecvența de cut-off mult mai mare decât frecvența fundamentală ⇒ filtrare slabă)
4. Reconstruiți semnalul de output cu ajutorul seriei Fourier (folosind formula de la exercițiul 1) [1p]

Exercițiul 3 - comunicație digitală [Bonus]

Am văzut la curs că pentru a transmite 2 biți simultan putem folosi două frecvențe diferite (f1, f2) pentru a coda o valoare de 2 biți:

1. '00': folosim un semnal egal cu 0 (nicio frecvență)

2. '01': folosim o sinusoidă ce conține doar prima frecvență ($\sin(2\pi f_1 t)$)
3. '10': folosim o sinusoidă ce conține doar a doua frecvență ($\sin(2\pi f_2 t)$)
4. '11': folosim ambele frecvențe f_1 și f_2

Task-ul vostru e să creați o secvență random de 10 valori între 0 și 3 (pentru a folosi toate valorile de mai sus) și apoi să o codificați folosind 2 sinusoide așa cum este descris mai sus. Pentru asta ar trebui să:

1. selectați frecvențele f_1 și f_2 astfel încât ele să folosească aceeași frecvență fundamentală (de ex.: $f_1 = 1 \cdot f_t$, $f_2 = 2 \cdot f_t$);
2. plotați semnalul rezultat folosind perioada ($1/f_t$) pentru fiecare valoare transmisă;
3. verificați că semnalul rezultat codează secvența voastră random;

Notă: pentru a genera o secvență random de valori întregi inspectați funcția 'randi' din MATLAB.

ps/labs/04.txt · Last modified: 2020/11/01 17:13 by ionut.gorgos