

Laboratorul 06.

Semnale digitale - procesarea simplă și eșantionarea

În acest laborator vom începe să experimentăm efectele semnalelor eșantionate și procesarea semnalelor digitale.

Materiale ajutătoare:

1. D. Johnson's book [<http://www.ece.rice.edu/~dhj/courses/elec241/col10040.pdf>]
a. Secțiunile 5.1-5.3

Exercițiul 1 -- Procesarea simplă [4p]

Vom încerca să implementăm un sistem de procesare digitală simplă, care funcționează după cum urmează:

- luați 5 eșantioane (samples) din semnalul de intrare, d_1, d_2, \dots, d_5 .
- faceți o medie a celor 5 eșantioane (samples), $a = 0.2 \cdot (d_1 + d_2 + d_3 + d_4 + d_5)$
- puneți la ieșire valoarea curentă $y(n) = a$

Urmăriți acești pași:

1. Creați câteva secvențe digitale care reprezintă sinusoidă de diferite frecvențe (1, 2, 10, 20, 100 Hz), la aceeași frecvență de eșantionare (folosiți același număr de eșantioane, e.g. $N = 128$).
2. Implementați sistemul de procesare menționat mai sus
3. Creați secvențele corespunzătoare de ieșire.
4. Plotați toate secvențele de intrare/ieșire
5. Ce fel de sistem de procesare este acesta ?
6. Cum ați putea implementa sistemul de procesare menționat mai sus, folosind funcția 'filter'? (opțional)

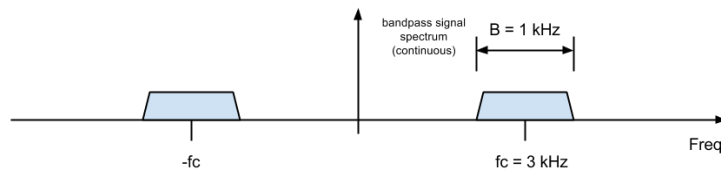
Pentru fiecare eșantion de intrare ($x(n)$) trebuie să produceți o valoare de ieșire ($y(n)$). Pentru primele patru eșantioane nu aveți suficiente valori de intrare diferite de zero, și pentru simplitate ori setați ieșirea pe zero, ori faceți media dintre 1, 2, 3, 4 eșantioane, în timp ce considerați ca celelalte să fie zero. Dar de la al 5-lea eșantion veți avea câte 5 eșantioane la îndemână (actualul $x(n)$ și anterioarele patru eșantioane).

Pentru crearea timpilor de eșantionare puteți folosi `0:perioada_esantionare:timp_maxim` sau funcția `linspace(0,timp_maxim,numar_esantioane)`

Exercițiul 2 -- Subeșantionare (Bandpass sampling) [3p]

În acest exercițiu vom experimenta subeșantionarea, metodă care reduce frecvența de eșantionare, necesară pentru a obține întreg spectru al semnalului.

Un semnal real cu o frecvență purtătoare (centrată) la 3kHz este eșantionată la 8 kHz. Lățimea de bandă a semnalului este 1 kHz, cum puteți vedea mai jos:



Task-uri/întrebări:

1. De ce avem energie în ambele frecvențe f_c și $-f_c$?
2. Folosiți Octave pentru a calcula transformata Fourier Discretă a unei sinusoide de 3 kHz eșantionată la 8 kHz . Folosiți $N = 64$ pentru fft. Folosiți acest cod pentru DFT:

```
figure;
fx = zeros(1, N);
fidx = (fs/N) * linspace(0, N-1, N);
spectrum = fft(x1, N);
stem(fidx, abs(spectrum));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Spectrum of x1');
```

3. Plotați (cu 'stem') rezultatele și observați frecvențele. Acum folosiți "fftshift" pe rezultatul obținut de la "fft" și plotați din nou rezultatele. Folosiți acest cod:

```
figure;
fidx = (fs/N)*linspace(-N/2, N/2-1, N);
stem(fidx, abs(fftshift(spectrum)));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Zero-centred frequency spectrum of x1');
```

4. De ce într-un caz avem semnal la 3 kHz și 5 kHz în timp ce, în celălalt caz pare să avem la -3 kHz și 3 kHz ?
5. Care este semnificația semnalului de 5 kHz ? Când plotați rezultatele de la "fft" luați în considerare că, în mod implicit output-ul transformatei Discrete Fourier, deci și "fft", arată frecvențele de la 0 la f_s (frecvența de eșantionare).

6. Putem să reducem frecvența de eșantionare în timp ce încă detectăm spectrul dorit de frecvențe ? (vedeți bandpass sampling). Cât de mult putem reduce frecvența de eșantionare? Ce se întâmplă dacă reducem frecvența de eșantionare sub $2B$?

Exercițiul 3 -- Reconstruire de semnal [3p]

Pentru reconstruirea unui semnal ideea este: - dacă se face sampling pe un semnal cu $f_s > 2 \cdot W$ (unde W este frecvența maximă a semnalului, i.e. respectiv Nyquist), atunci puteți să reconstruiți din semnalul digital semnalul analog (exceptând o eroare de cuantizare) pe baza unui low pass filter (cu un sinc).

Vom simula un semnal sinus analog, pe care îl vom eșantiona, apoi îl vom reconstrui. Pentru asta ar trebui să faceți următorii pași:

- simulați un semnal sinus 'analog'. Pentru aceasta generați un semnal sinus de frecvență 1Hz cu un număr mare de puncte (i.e. 5000 de puncte). Pentru timp folosiți:

```
t = linspace(0, 5, N_analog);
```

(i.e. sinusoida face 5 perioade).

- construiți semnalul digital, adică varianta eșantionată a semnalului analog, folosind un număr mai mic de puncte (i.e. de 100 de ore mai puține) din semnalul analog.
- faceți din semnalul digital un semnal continuu (analog), prin crearea unui semnal de aceeași lungime ca cel analog inițial, folosind valorile din semnalul digital pentru valorile de la momentele eșantionate și zero în rest.
- faceți convoluție între un sinc (de frecvență f_s , dar centrat pe zero, adică un semnal simetric) și semnalul continuu creat anterior.

Pentru a face convoluție cu un semnal sinc, folosiți codul urmator:

```
N_sinc = N_analog;  
t_sinc = linspace(-0.2, 0.2, N_sinc/10);  
sincvec = sinc(samples_digital*t_sinc);  
s_cont_filtered = conv(s_cont, sincvec);
```

, unde `samples_digital` reprezintă numărul de eșantioane pentru semnalul digital.

Trebuie să folosiți funcția `stem`, nu `plot` pentru reprezentare !

ps/labs/06.txt · Last modified: 2020/11/11 13:21 by ionut.gorgos