



UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI DISCIPLINA ACHIZITIA SI
PRELUCRAREA DATELOR PROIECT

Realizarea unei interfete in LabWindows/CVI care achizitioneaza, filtreaza si calculeaza spectrul de putere a unui semnal dintr-un fisier WAV

Student,
Taga Dan-Claudiu
1307A

Iasi 2020

1. Rezumat

Folosind limbajul python, se convertește un fișier .wav într-un fișier .txt, alături de un alt fișier .txt în care se afla numărul de puncte și rata de esantionare. Aceste fișiere .txt vor fi apoi încărcate într-o interfață LabWindows/CVI, folosind limbajul C.

În etapa 1, se analizează semnalul în domeniul timp; adică, va fi afișat pe un graf, alături de alte informații relevante cum ar fi min/max, dispersie, etc., apoi, în funcție de opțiunile alese de utilizator, va fi filtrat fie prin mediere fie cu un element de ordin 1. Din rezultatul filtrării se va alege un interval de secunde ce va fi afișat pe un graf.

În etapa 2, se alege un interval de secunde din vectorul cu semnalul nefiltrat, se va afișa, și i se va calcula spectrul de putere al primelor N valori, N fiind numărul de puncte pentru Transformata Fourier. Spectrul rezultat va fi afișat. Apoi se va alege secunda 2 din semnalul nefiltrat. Semnalul obținut va fi fereștruit fie prin fereastră de tip Gauss, fie de tip Welch. După fereștruire, semnalul va fi filtrat prin unul din două filtre: mediere pe 16 valori sau filtru Bessel de grad 4 și 6 trece jos pt 1/2 din spectru. Semnalul rezultat va fi afișat și i se va calcula spectrul, care va fi și el afișat.

2. Cerințe

Medii de dezvoltare: Python 3.8 + NumPy + SciPy, LabWindows/CVI.

Fișierul utilizat: 50.wav, un fișier wav de ~6 secunde.

Cerințe proiect:

- Încărcare fișier wav (50.wav)

- Afișare semnal pe control tip Graph.

- Calcularea și afișarea valorilor min/max, dispersie, medie, mediana.

- Se implementează funcțiile pentru filtrare prin două metode: mediere (pe 16 sau 32 de elemente) cu un element de ordin I conform relației:
$$\text{filt}[i] = (1 - \alpha) * \text{filt}[i-1] + \alpha * \text{signal}[i]$$

- Realizați afișarea grafică a semnalului filtrat pe intervale de timp de o secundă.

- Pentru fiecare interval de timp se salvează imaginile obținute pentru semnalul dat și cel filtrat.
- Se include în aplicație un nou panou conform implementării SwitchPanels de pe Moodle. Se înlocuiește "Acquisition" cu "Frequency".
- În panoul nou se va implementa prelucrarea în frecvență a fișierului wav.
- Se include un control de tip Graph pentru reprezentarea spectrului întregului semnal și un control numeric tip intreg pentru selectarea numărului de puncte (N) pentru Transformata Fourier.
- Se va calcula și reprezenta spectrul pe semnalul întreg sau pe câte o sec.
- Se completează analiza semnalului în domeniul timp cu calculul și afișarea numărului de treceri prin zero a semnalului (zero-crossing).
- Utilizează două tipuri de ferestre (Welch, Gauss) și două tipuri de filtre (mediere pe 16 valori sau filtru Bessel de grad 4 și 6 trece jos pt 1/2 din spectru) pentru procesarea semnalului. Procesarea se face pentru o singură secundă (secunda 2). Se reprezintă semnalul și spectrul înainte și după filtrare. Comentarii privind rezultatul obținut.

3. Analiza în domeniul timp

Fisierul .txt a fost încărcat într-un vector waveData folosind funcția FileToArray(..), la fel ca și rata de esantionare și numărul de puncte.

Pentru a obține punctul maxim, minim, media, mediana, dispersia am folosit funcții labwindows CVI:

Pentru max și min: MaxMin1D(..);

Pentru medie: Mean(..);

Pentru mediana: Median(..);

Pentru numărul de câte ori a trecut prin zero semnalul (zerocrossing), am creat o funcție în care am înmulțit câte 2 puncte consecutive, iar dacă rezultatul e mai mic ca 0, înseamnă că a trecut prin 0, și crește un increment de numărare a trecerii din prin 0.

Apoi valorile obținute au fost puse pe controale de pe interfața folosind funcția SetCtrlVal(...);

Apoi am creat funcțiile de filtrare prin mediere și filtrare cu un element de ordin 1 (filtrare alpha). Funcția de filtrare prin mediere primește ca parametru numărul de puncte cu care se face medierea. Funcția de filtrare alpha primește ca parametru variabila alpha din formula

filtrării. Alpha nu ar trebui să fie 1 sau 0. Utilizatorul poate selecta din cele două filtrări printr-un control de pe interfață, la fel cum poate selecta valorile parametrilor n și α .

Filtrarea rezultată se va salva într-un alt vector, `filteredData`.

Pentru a afișa o perioadă specifică de pe vectorul filtrat, am creat un pointer care arată spre o secundă anume $\text{pointer} = \text{filteredData} + (\text{int})(\text{secStart} * \text{sampleRate})$; (o secundă = `sampleRate` puncte).

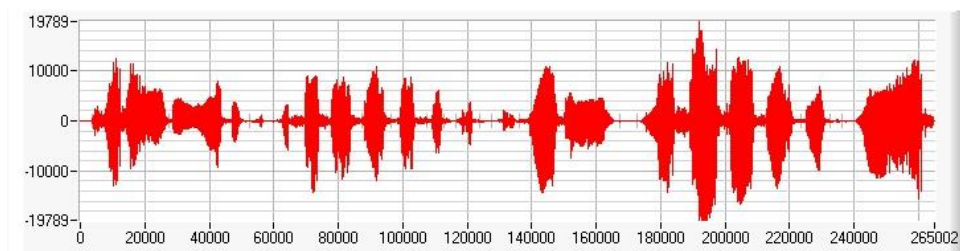
Vectorul va fi afișat folosind funcția `PlotY`, afișând un număr de $(\text{secStop} - \text{secStart}) * \text{sampleRate}$ puncte, unde `secStop` și `secStart` își iau valorile de pe controalele interfeței. În sensul acesta, se afișează strict perioada de la `secStart` la `secStop`.

Pentru a salva imagini ale grafurilor, am folosit bitmap-uri.

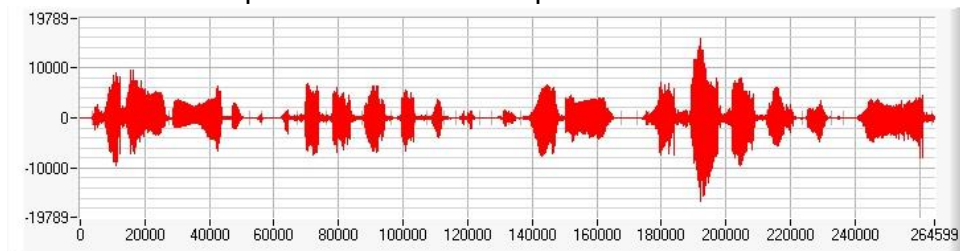
```
GetCtrlDisplayBitmap(...) -> SaveBitmapToJPEGFile(...) -> DiscardBitmap(...);
```

Exemple de rezultate:

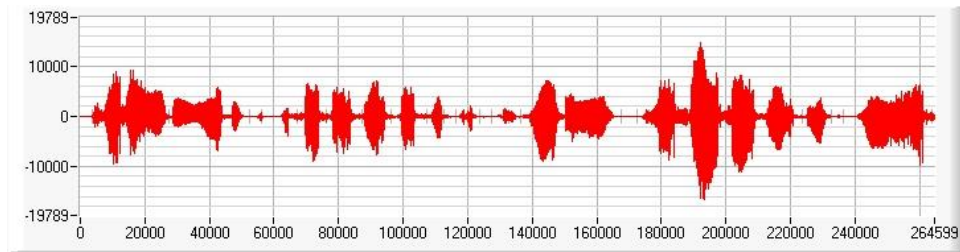
Semnalul nefiltrat:



Semnalul filtrat prin mediere cu 16 puncte:

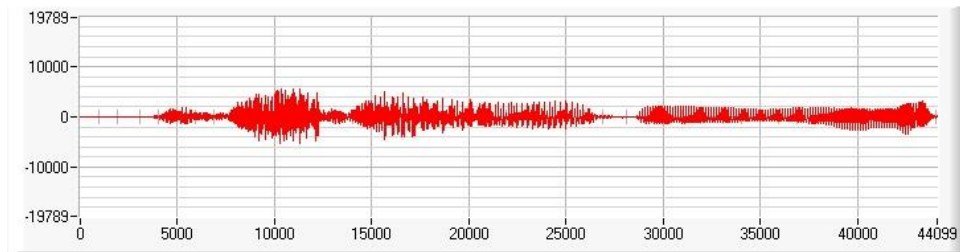


Semnalul filtrat cu α de valoare 0.16:

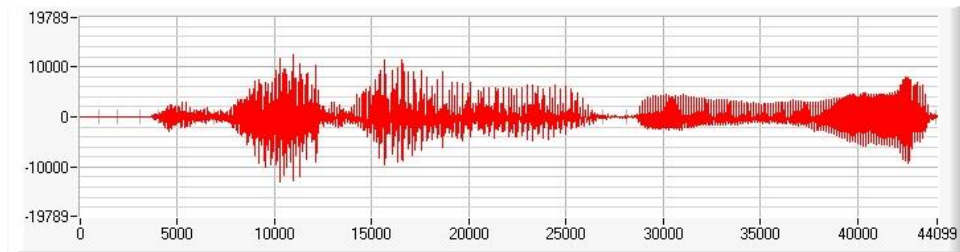


Se poate observa cum filtrările scad amplitudinile semnalului.

Perioada 0-1 filtrată prin mediere cu 32 puncte:



Perioada 0-1 filtrată cu alpha de valoare 1:



Se poate constata că în urma filtrării cu alpha de valoare 1, semnalul este de fapt nefiltrat.

Rezultatele de minim, maxim, media, mediana, dispersia și zero crossing ale semnalului nefiltrat.

Min	Mean	Max	Median	Dispersion	Zero Crossing
-23151.00	0.00	19787.00	0.00	2424.75	17052

Putem observa că media și mediana sunt 0, datorită formei semnalului.

4. Analiza în domeniul frecvență

Folosind un control de tip switch, în interfață apare un al panou în care se va lucra în domeniul frecvență.

Folosind un pointer care arată spre semnalul nefiltrat waveData, putem alege strict un interval din semnal, de la 0 la 6 secunde (idem cu prima parte).

Am calculat spectrul acestui interval din semnal (perioada care poate fi intreg semnalul, cateva secunde sau o secunda) folosind functiile:

```
ScaledWindowEx(pointer, N, RECTANGLE_, -1, &windowConstants);
```

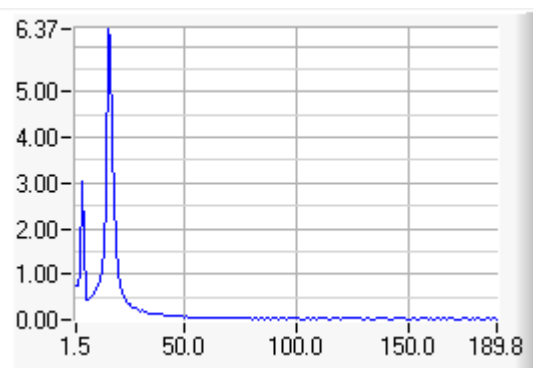
```
AutoPowerSpectrum(pointer, N, dt, autoSpectrum, &df);
```

```
PowerFrequencyEstimate(autoSpectrum, N/2, -1, windowConstants,  
df, 7, &freqPeak, &powerPeak);
```

```
SpectrumUnitConversion(autoSpectrum, N/2, 0,  
SCALING_MODE_LINEAR, DISPLAY_UNIT_VRMS, df, windowConstants,  
convertedSpectrum, unit);
```

Unde N este numarul punctelor Transformatei Furier. In sensul acesta, programul calculeaza spectrul primelor N puncte din intervalul dat. N poate fi chiar numarul punctelor unei secunde. (44100 puncte)

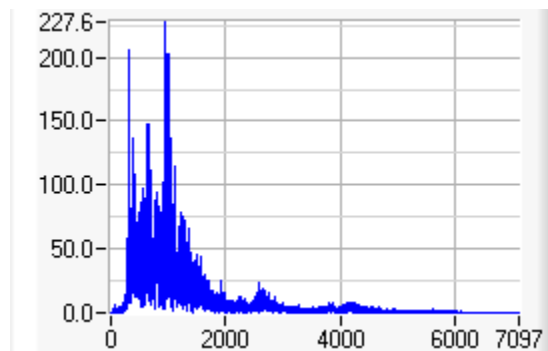
Spectrul de putere a primelor 2048 puncte din semnalul nefiltrat:



PowerPeak si FreqPeak pentru spectrul primelor 2048 puncte din semnalul nefiltrat:

powerPeak	freqPeak	N
106.38	356.32	2048

Spectrul de putere a primei secunde din semnalul nefiltrat:



PowerPeak si FreqPeak pentru spectrul primei secunde din semnalul nefiltrat:

powerPeak	freqPeak	N
98499.87	953.83	44099

Pentru a ferestrui semnalul, ca apoi sa-l filtram, mai intai am ales un pointer spre semnalul nefiltrat, mai specific o secunda din semnal. Am ales secunda 2.

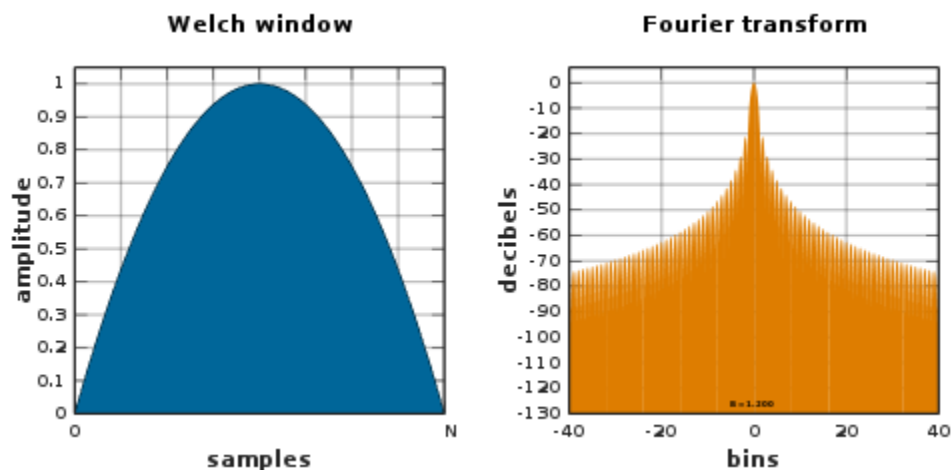
Am folosit functia LinEv1D(pointer2, sampleRate, 1, 0, window) pentru a evalua liniar vectorul in care se afla secunda ce va fi filtrata.

Apoi, pe vectorul rezultat, window, aplicam ferestruirile:

```
if (tip_fereastră == 0)
    WelchWin(window, sampleRate);
else
    GaussWin(window, sampleRate, 0.2);
```

Fereastră Welch consta dintr-o singura sectiune parabolica:

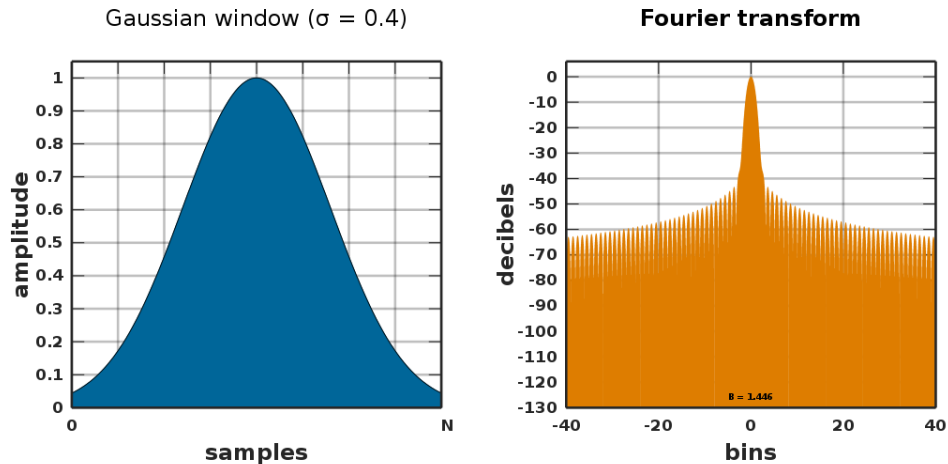
$$w[n] = 1 - \left(\frac{n - \frac{N}{2}}{\frac{N}{2}} \right)^2, \quad 0 \leq n \leq N.$$



Fereastra Gaussiană:

$$w[n] = \exp\left(-\frac{1}{2}\left(\frac{n - N/2}{\sigma N/2}\right)^2\right), \quad 0 \leq n \leq N.$$

$$\sigma \leq 0.5$$



Utilizatorul poate alege ce fereastră dorește dintr-un control ring.

În urma fereștrării, urmează filtrarea semnalului.

În acest sens, semnalul poate fi filtrat fie prin mediere pe 16 puncte, fie prin filtru Bessel de grad 4 și 6 trece jos până la 1/2 din spectru.

Medierea pe 16 puncte:

```
int i, j;
double sum = 0;
for ( i = 0; i < sampleRate; i++ )
{
    for ( j = 0; j < 16; j++ )
    {
        if ( i-j >= 0 )
            sum+=window[i-j];
    }

    secundaFiltrata[i] = sum/16;
    sum = 0;
}
```

Medierea pe 16 constă în media aritmetică a cel mult 16 puncte și salvarea rezultatului în punctul dat. În acest sens, se va face media pe cei mai apropiați 16 vecini din stânga ai unui punct de pe vector.

Filtru Bessel trece jos de grad 4 si 6 pentru $\frac{1}{2}$ din spectru se obtine folosind functia

`Bssl_LPF(window,sampleRate,sampleRate,sampleRate/4,4,secundaFiltrata),`
(respectiv

`Bssl_LPF(window,sampleRate,sampleRate,sampleRate/4,6,secundaFiltrata)).`

Deoarece este pentru $\frac{1}{2}$ din spectru, rezulta ca frecventa de taiere va fi jumatate din spectru, si anume jumatate din spectrul maxim, adica rata de esantionare/2, rezultand rata de esantionare/4, adica $\text{sampleRate}/4$.

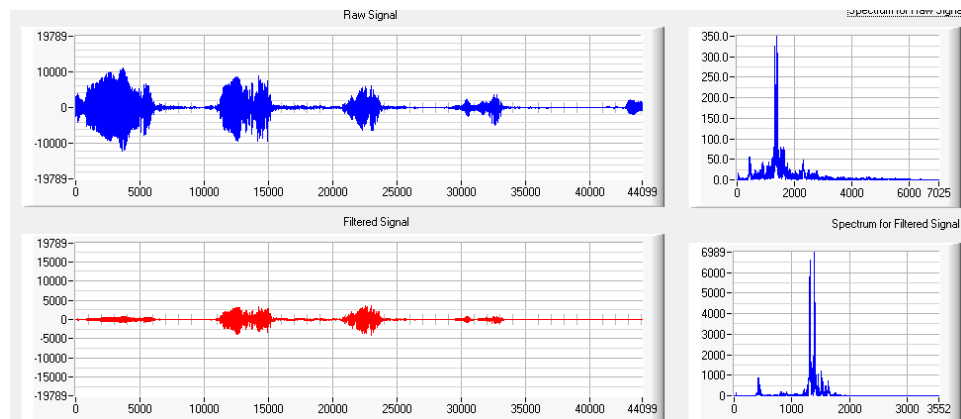
Un filtru Bessel este un tip de filtru analogic liniar cu o intarziere maxima plata de grup / faza (raspuns de faza maxim liniar), care pastreaza forma de unda a semnalelor filtrate in banda de trecere.

In cazul nostru, unde avem filtru Bessel trece jos, acesta va pastra semnalele de amplitudine mica.

Secunda filtrata e apoi mutata intr-un vector, caruia ii calculam spectrul prin aceeasi metoda ca semnalului nefiltrat.

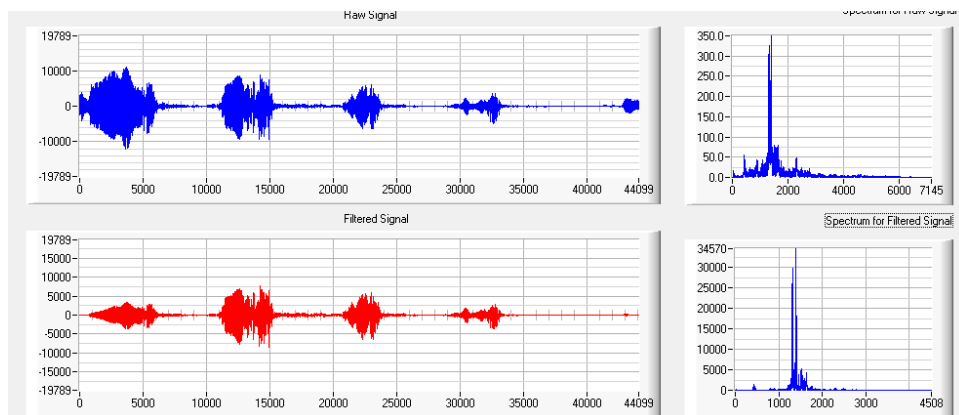
Rezultate:

Intreaga secunda 2 nefiltrata (sus), spectrul ei, si apoi filtrata cu ferestruire Gauss si filtrata prin mediere pe 16 puncte:



Se pot observa asemanarile spectrelor, dar si diferenta cauzata de filtrare.

Filtrarea secunde 2 cu ferestruire Welch, filtrata prin filtru Bessel de grad 6:



Se poate observa cum sunete de frecvență joasă au fost păstrate.