UNIVERSITATEA TEHNICA “GHEORGHE ASACHI” IAȘI

FACULTATEA AUTOMATICĂ ȘI CALCULATOARE

SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

DISCIPLINA ACHIZIȚIA ȘI PRELUCRAREA DATELOR-PROIECT

Coordonator,

Prof. Florina Ungureanu

Student,

Chiteală Tudor Matei

Iași, 2020

Cerințele proiectului:

ETAPA 1:

Realizarea interfeței în CVI . Citirea datelor din fișierul audio WAV, reprezentarea grafică, evaluarea și afișarea parametrilor în domeniul timp. Modalitați de filtrare în domeniul timp(mediere si element de ordin I).

ETAPA 2:

Extinderea interfeței. Analiza in frecvența a semnalelor din fisierul audio WAV.Reprezentarea spectrelor pe cate o secunda. Calcularea numarului de treceri prin zero a semnalului.Folosirea a doua tipuri de ferestre si a doua filtre pe o secunda.

Fișierul utilizat:

Fișierul folosit in aplicatie este un fișier audio WAV de 6 secunde care reprezinta sunetul unei vioare.

Mediul de dezvoltare:

LabWindows/CVI este o platformă pentru dezvoltare de software cu orientare pe aplicaţii de instrumentaţie.

Analiza în domeniul timp(Etapa 1):

In etapa 1 am avut de aplicat **filtrele de mediere** si **element de ordin I.**

Filtrare prin mediere:

```
if(filter == 0){
    int n;
    double sum0 = 0;
    GetCtrlVal(panel, Main_Panel_Median_Number_Switch, &n);
    for(int i = 0; i < n; i++)
        sum0 += waveData[i];
    for(int i = 0; i < n; i++)
        filterWave[i] = sum0/n;
    for(int i = n; i < waveInfo[1]; i++){
        sum0 -= waveData[i-n];
        sum0 += waveData[i];
        filterWave[i] = sum0/n;
    }
    PlotY(panel, Main_Panel_Graph_Panel_Filter, filterWave, npoints, VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, VAL_CONNECTED_POINTS, VAL_GREEN);
}
```

Variabila filter specifică ce tip de filtru se aplica pe semnal. Daca filtru este 0 se aplica filtrul de mediere.Acesta va lua in considerare ce fel de mediere se face(pe 16 sau 32 de elemente) dupa variabila **n**. In variabila sum0 este salvata suma primelor n elemente. Primele elemente **n** din filterWave vor avea sum0/n. In ultimul for vom incepe de indexul **n** . Din variabila sum0 vom

scoate ultimul element adica indexul[i-n] si vom adauga noua valoarea adica index i. Dupa vom introduce la indexul i sum0 /n. Exemplu: sum0 = sum0 – S0 + S16; sum0 = sum0 -S1 + S17;….

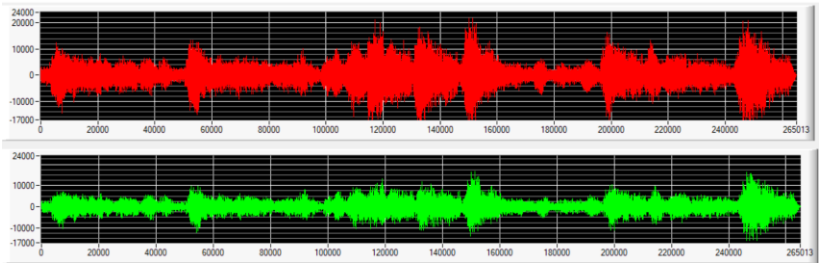
Filtrare de ordinal I:

```
double alpha;

GetCtrlVal(panel, Main_Panel_Alpha_Numeric, &alpha);
filterWave[0] = waveData[0];
for(int i = 1; i < waveInfo[1]; i++)
    filterWave[i] = (1-alpha)*filterWave[i-1] + alpha*waveData[i];

PlotY(panel, Main_Panel_Graph_Panel_Filter, filterWave, npoints, VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, VAL_CONNECTED_POINTS, VAL_GREEN);
```

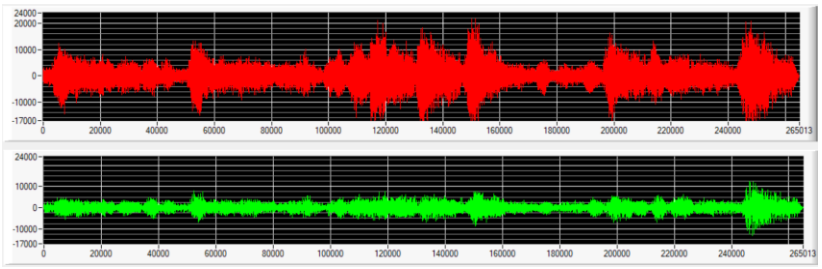
Pentru filtrul de ordinul I se aplica formula filt[i] = (1-alpha)*filt[i-1] + alpha*signal[i], unde filt[0] = wavedata[0]. Se parcurg toate punctele si se aplica formula specifica.Semnalul filtrat va fi filterWave.

Se poate observa cum filtrarea prin mediere cu 32 elemente filtreaza mai mult decat filtrarea cu 16 elemente

Mediere cu 16 elemente:

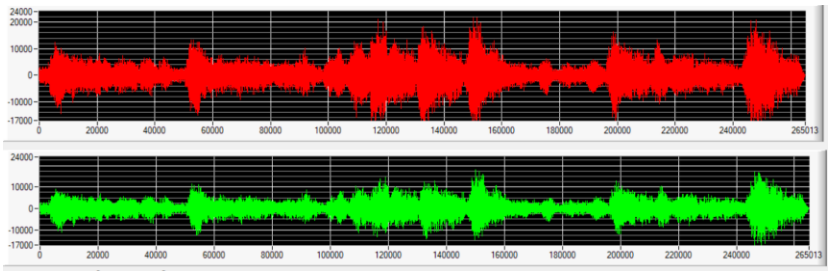


Mediere cu 32 elemente:

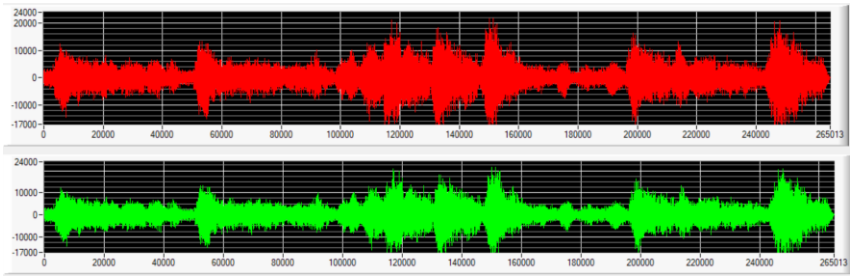


Pentru filtrarea de ordinal I se poate observa cum pentru un alpha mare semnalul nu este foarte mult filtrat.Dar pentru un alpha mic(mai apropiat de 0) filtrarea este mai mare.

Alpha = 0.2



Alpha = 0.9



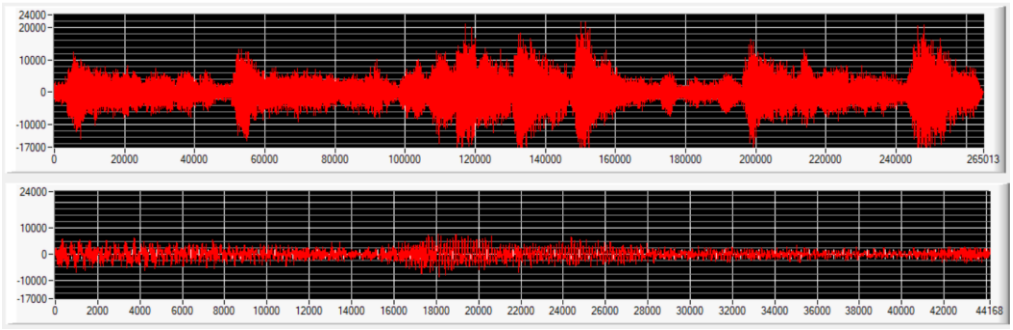
Parcurgerea semnalului din secunda in secunda:

```
int CVICALLBACK OnChangeSecond (int panel, int control, int event,
                                void *callbackData, int eventData1, int eventData2)
{
    double * temp;

    switch (event)
    {
        case EVENT_COMMIT:


            GetCtrlVal(panel,Main_Panel_Start_Numeric, &start);
            GetCtrlVal(panel, Main_Panel_Stop_Numeric, &stop);

            temp = (double *) calloc(npoints/6, sizeof(double));
            switch(control)
            {
                case Main_Panel_Next_Button:
                    if(stop<6){
                                DeleteGraphPlot(panel, Main_Panel_Graph_Panel_Filter, -1, VAL_IMMEDIATE_DRAW);

                                ++start;
                                ++stop;
                                SetCtrlVal(panel, Main_Panel_Start_Numeric, start);
                                SetCtrlVal(panel, Main_Panel_Stop_Numeric, stop);


                                for(int i=0;i<npoints/6;++i)
                                    temp[i] = filterWave[start*npoints/6 + i ];

                                PlotY(panel, Main_Panel_Graph_Panel_Filter, temp, npoints/6, VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, VAL_CONNECTED_POINTS, VAL_RED);
                                int id;
                                GetCtrlDisplayBitmap(panel, Main_Panel_Graph_Panel_Filter, 0, &id);
                                SaveBitmapToJPEGFile(id, "Poza.jpg", JPEG_PROGRESSIVE, 100);
                            }
                break;


                case Main_Panel_Prev_Button:
                    if(stop>1){
                                DeleteGraphPlot(panel, Main_Panel_Graph_Panel_Filter, -1, VAL_IMMEDIATE_DRAW);
                                --start;
                                --stop;
                                SetCtrlVal(panel, Main_Panel_Start_Numeric, start);
                                SetCtrlVal(panel, Main_Panel_Stop_Numeric, stop);

                                for(int i=0;i<npoints/6;++i)
                                    temp[i] = filterWave[start*npoints/6 + i ];


                                PlotY(panel, Main_Panel_Graph_Panel_Filter, temp, npoints/6, VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, VAL_CONNECTED_POINTS, VAL_RED);
                                int id;
                                GetCtrlDisplayBitmap(panel, Main_Panel_Graph_Panel_Filter, 0, &id);
                                SaveBitmapToJPEGFile(id, "Poza.jpg", JPEG_PROGRESSIVE, 100);

                            }

                break;

            }
            free(temp);
            break;
    }
    return 0;
```

Tot in etapa 1 se va parcurge semnalul din secunda in secunda. Fisierul audio este de 6 secunde. Si cea mai mica secunda reprezenta poate sa fie de la 0 la 1. Aceste constrangeri sunt evidențiate in stop < 6 și stop > 1. Daca este apasat butonul Next atunci start si stop vor creste cu 1 și pentru Prev acele doua variabile vor scadea cu 1 pentru a trece la urmatoarea secunda. Pentru a afisa secunda respectiva se vor lua doar npoints/6 puncte.Pentru a lua secventa corecta, „start” va specifica de la ce index va trebui sa incepem sa luam punctele.

La final se vor salva poze jpg cu semnalul pentru acel interval.

Semnal intre secundele 3 – 4:



Analiza in frecvență(Etapa 2):

In etapa 2 se calculează spectrul semnalului și se afișeaza pe cate o secundă.

Se observa ca pe parcursul afisarii spectrelor pe fiecare secunda acestea nu prezinta modificari foarte mari deoarece semnalul WAV este constant(sunetul unei viori)

Se vor vedea spectrele pe cate o secunda pentru semnalul inițial, semnalul dupa ferestruire și semnalul dupa ferestruire si filtrare.

```
                    convertedSpectrum, Unit);
switch(wind){
    case 0:
        ScaledWindowEx(nWave,N,RECTANGLE,0,&winConst);
        break;
    case 1:
        ScaledWindowEx(nWave,N,BLKMAN,0,&winConst);
        break;
}
```

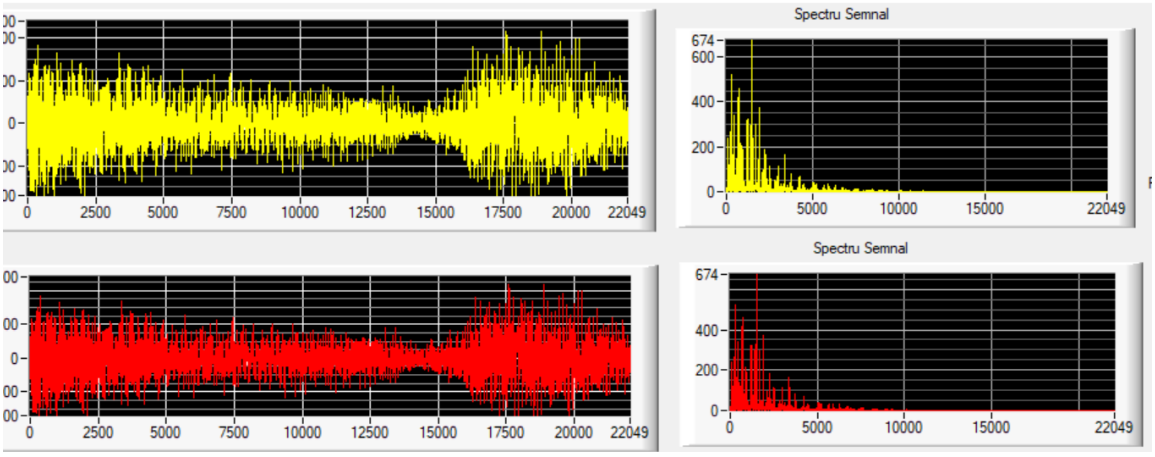
Pentru Window se foloseste funcția ScaledWindowEx din CVI.

Pentru a aplica ferestruierea dorita vom seta variabilele RECTANGLE și BLKMAN

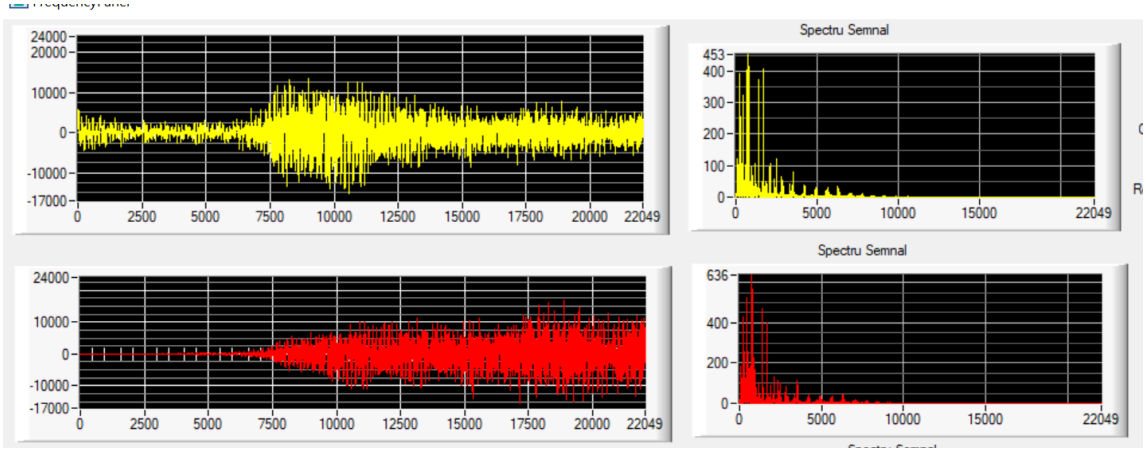
Pentru RECTANGLE funcția de transfer este w[n] = 1. Semnalul nu este modificat.

Pentru BLACKMAN funcția de transfer este w[n] = a0 – a1*cos(2*pi*n/N) + a2*cos(4*pi*n/N).

Pentru window de tip **RECTANGLE** semnalul nu se modifica.



Pentru window de tip **BLACKMAN** semnalul este filtrat la inceput dar nu complet pe tot parcursul secventei. Fereasta BLACKMAN este asemanatoare cu cea Hamming doar ca la finalul ecuatiei este adaugat un cosinus pentru reducerea “ripple”.



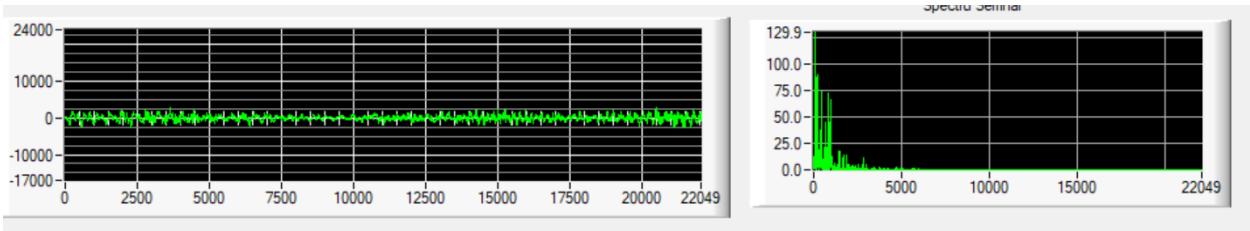
Filtrarea:

Pentru filtrul **CHEBYSHEV si NOTCH** se observa diminuarea intensitatilor semnalului.

CHEBYSHEV TIP 2:

Chebyshev invers(sau de tip 2) prezintă riplu doar în banda de stop. Zerourile unui filtru Chebyshev 2 sunt inverse zerourilor unui Chebyshev 1. Este aplicată formula:

$$H(j\omega) = \frac{1}{\sqrt{1 + \frac{1}{\varepsilon^2 Tn^2 \left(\frac{\omega}{\omega_t}\right)}}}$$



NOTCH:

Filtrul Notch este de tip OB special, are banda de oprire foarte ingusta.

Funcția de transfer este:

$$H(z) = \left(1 - e^{j\omega_0} z^{-1}\right)\left(1 + e^{-j\omega_0} z^{-1}\right)$$
$$= 1 - 2\cos(\omega_0)\, z^{-1} + z^{-2}$$