

[ANEXO 5]

Código del Programa ondícula vía FFT Implementado en Scilab

```
// Display mode
mode(0);

// Display warning for floating point exception
ieee(1);

//programa para la obtencion de los numeros MIDI de cada nota
//inicializacion de las variables

y = [];
Y = [];
armonicos = [];
armonicof = [];
evento = [];
duracion = [];
nota = [];
m = [];
clear nombre

//inicializar los otros programas
exec("SCI/sciwork/num2midi.sci");
exec("SCI/sciwork/set_delta.sci");
exec("SCI/sciwork/gausswin.sci");

nombre=input('Introduzca el nombre del archivo WAVE que se va a analizar?','string');
nombreM=input('Introduzca el nombre del archivo MIDI de salida','string');

//lee el archivo WAVE a analizar
[y,fs,bits]=wavread(nombre);

//lee la longitud de la data del archivo WAVE
long = wavread(nombre,'size');

//asigna la longitud a un escalar
//nota si el archivo es estereo se leera uno solo de los canales
N = long(1,2);

if long(1,1)~=1 then
    y = y(1,1:N-1);
```

```

end;

//calcula la energia de la senal
E = sum(y.*y)/N;

//duracion del segmento
seg = 0.2;

//cacula el numero entero de muestras que hay en 0.2 segundos
paso = fix(fs*seg);

//calcula la longitud aproximada de la data en un numero entero de "pasos"
//menos un paso
long_pasos =(N-fix(N/paso))-paso;

n =1:fix(paso/2);

//vector que contiene la longitud de una vetana en tiempo
t = (1:fs*seg)/fs;

//ventana gaussiana
//w = gausswin(mtlb_double(fs)*seg);

//selecciona hasta que nota-MIDI se va hacer el analisis dependiendo de la
//la frecuencia de muestreo de la señal
select fs
    case 11025 then
        NF = 79;
    case 22050 then
        NF = 92;
    case 44100 then
        NF = 127;
end;

T = max(size(t));

for nf = 12:NF
    //senal patron (armonicos)
    //w'.*
    //la constante 51.37006=2*pi*440*(-69/12)
    armonicos=gausswin(T,nf/12)'.*(sin((51.37006*(2^(nf/12)))*t)+
(3/4)*sin((51.37006*(2^((nf+12)/12)))*t))+
(2/3)*sin((51.37006*(2^((nf+19)/12)))*t)+
(1/2)*sin((51.37006*(2^((nf+24)/12)))*t)+
(2/5)*sin((51.37006*(2^((nf+28)/12)))*t)+
(1/3)*sin((51.37006*(2^((nf+31)/12)))*t));

    //transformada de Fourier de la señal patron
    armonicosf(nf,1:T) = abs(fft(armonicos,-1));

end;

```

```

k = 1;
l = 1;

for i = 1:paso:long_pasos

    j = i+paso-1;

    //determina si el segmento es una nota o un silencio

    energia_segmento = sum(y(i:j).*y(i:j))/(paso-1);
    //energia del segmento

    if energia_segmento>=E*0.1 then
        //calcula la fft de cada segmento
        Y = abs(fft(y(i:j)));

        //se toma la mitad del espectro de la señal
        Y = Y(n);

        //comparacion con la senal patron de cada nota
        //la variable nf es el numero midi de cada nota y NF es el numero total
        //de notas a analizar que depende de la frecuencia de muestreo de la
        //senal a analizar
        for nf = 12:NF

            //producto punto del espectro de la señal patron con el de la senal a analizar

            m(1,nf+1) = matrix(sum(armonicof(nf,1:T/2).*Y),1,-1);

        end;
        //vector que contiene la posicion de los maximos del producto punto entre
        //el modelo y la señal
        nota(1,k) = find(m==max(m))-1;

        if k==1 then
            //960=tick por negra 2=negra por segundos (tempo)
            segmento = (seg*960)*2;
            duracion = set_delta(segmento);
            evento(1,1) = 144;
            evento(1,2) = nota(k);
            evento(3) = 127;
            for l = 4:3+max(size(duracion))
                evento = mtlb_i(evento,l,mtlb_e(duracion,l-3));
            end;

        elseif nota(k)==nota(k-1) then
            L = l-max(size(duracion))+1;
            segmento = segmento+(seg*960)*2;
            duracion = set_delta(segmento);
            u = 1;
            for l = L:L+max(size(duracion))-1

```

```

        evento(1,l) = matrix(duracion(u),1,-1);
        u = u+1;
    end;

elseif nota(k)~=nota(k-1) then
    if nota(k-1)~=(-5) then
        evento(1,l+1) = nota(k-1);
        evento = mtlb_i(evento,l+2,0);
        evento = mtlb_i(evento,l+3,0);
        evento = mtlb_i(evento,l+4,nota(k));
        evento = mtlb_i(evento,l+5,127);
        u = 1;
        L = l+6;
        segmento = (seg*960)*2;
        duracion = set_delta(segmento);
        for l = L:L+max(size(mtlb_double(duracion)))-1
            evento = mtlb_i(evento,l,mtlb_e(duracion,u));
            u = u+1;
        end;
    else
        evento(1,l+1) = nota(k);
        evento = mtlb_i(evento,l+2,127);
        u = 1;
        L = l+3;
        segmento = (seg*960)*2;
        duracion = set_delta(segmento);
        for l = L:L+max(size(mtlb_double(duracion)))-1
            evento = mtlb_i(evento,l,mtlb_e(duracion,u));
            u = u+1;
        end;
    end;
end;
else
    if k==1 then
        nota(1,k) = -5;
        k = 0;
    elseif nota(k-1)==(-5) then
        L = l-max(size(duracion))+1;
        //960=tick por negra 2=negra por segundos (tempo)
        segmento = mtlb_a(mtlb_double(mtlb(segmento))),(seg*960)*2);

        duracion = set_delta(segmento);
        u = 1;
        for l = L:L+max(size(mtlb_double(duracion)))-1
            evento(1,l) = matrix(mtlb_e(duracion,u),1,-1);
            u = u+1;
        end;
        nota(1,k) = -5;
    elseif nota(k-1)~=(-5) then
        segmento = (seg*960)*2;

        duracion = set_delta(segmento);

```

```

    evento(1,l+1) = nota(k-1);
    evento = mtlb_i(evento,l+2,0);
    L = l+3;
    u = 1;
    for l = L:L+max(size(mtlb_double(duracion)))-1
        evento = mtlb_i(evento,l,mtlb_e(duracion,u));
        u = u+1;
    end;
    nota(1,k) = -5;
end;
end;

k = k+1;

end;
//apaga la ultima nota si no es un silencio
if nota(1,k-1)~= -5 then
evento=[evento nota(1,k-1) 0 0];
end
num2midi(evento,nombreM);
disp('listo');

```