

[ANEXO 6]

Código del Programa ondícula vía correlación 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
x = [];
y = [];
armonicos = [];
evento = [];
nota = [];
m = [];

//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
// !! L.14: Matlab function wavread not yet converted, original calling sequence used
[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:N-1);
end;

//calcula la energia de la senal
```

```

E = sum(y.*y)/N;

//duracion del segmento
seg = 0.2;
//(2^13)/fs;

//numero de muestras de un segmento
T=fs*seg;

//calcula el numero entero de muestras que hay en "seg" segundos
paso = fix(T);

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

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

TT = max(size(t));

//ventana de Gauss
//w=gausswin(T)'';

//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;

for nf = 12:NF
    //senal patron (armonicos)
    //mtlb_double(mtlb_t(gausswin(T,nf/12))) .*
    armonicos(nf,1:TT) =gausswin(TT,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));
end;

k = 1;
l = 1;

//comparacion con la senal patron de cada nota
//la variable nf es el numero de 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 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

        for nf = 12:NF

            //correlacion entre la sanal patron y la senal a analizar
            x = convol(armonicos(nf,1:TT),y(i:j));
            //(2^(nf/24))*

            //calcula de la energia de cada segmento y la guarda en el vector m
            m(1,nf+1) = matrix(sum(x.*x),1,-1);
        end;

        //vector que contiene la posicion de los maximos del producto punto entre
        //el modelo y la señal
        [maximo,columna] = max(m);
        nota(1,k)=columna-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,duracion(l-3));
            end;

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

```

```

    u = 1;
    L = l+6;
    segmento = (seg*960)*2;
    duracion = set_delta(segmento);
    for l = L:L+max(size(duracion))-1
        evento = mtlb_i(evento,l,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(duracion))-1
        evento = mtlb_i(evento,l,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(mtlb_double(mtlb(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(duracion))-1
            evento(1,l) = duracion(u);
            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(duracion))-1
            evento = mtlb_i(evento,l,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(k-1)~=(-5) then  
    evento = [evento,nota(k-1),0,0];  
end;  
//funcion que escribe el archivo MIDI  
num2midi(evento,nombreM);  
disp('listo!!!');
```