

[ANEXO 3]

Código del Programa ondícula vía FFT Implementado en MatLab

```
%programa para la obtencion de los numeros MIDI de cada nota

%inicializacion de las variables
y=[];
Y=[];
armonicos=[];
armonicof=[];
evento=[];
duracion=[];
nota=[];
m=[];

%pide al usuario el nombre del archivo Wave a analizar
nombre=input('Introduzca el nombre del archivo WAVE
que se va a analizar? ','s');

%pide al usuario el nombre del archivo Wave a analizar
nombreM=input('Introduzca el nombre del archivo MIDI de salida? ','s');

%lee el archivo WAVE a analizar
[y,fs]=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(1,1);

if long(1,2)~=1
    y=y(1:N-1,1);
end

%calcula la energia de la senal
E=dot(y,y)/N;

%almacena la duracion del segmento
seg=0.2;

%calcula 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 "paso's"
%menos un paso
long_pasos=N-fix(N/paso)-paso;

%vector que contiene la mitad de las muestras contenidas en 0,2s
n=1:fix(paso/2);

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

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

T=length(t);

for nf=12:NF
    %señal de comparacion (armonicos), gausswin ventana gaussiana
    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 de comparacion
    armonicosf(nf,1:T)=abs(fft(armonicos));

end

k=1;
l=1;

%deteccion de las notas en cada segmento
for i=1:paso:long_pasos

    j=i+paso-1;

    %determina si el segmento es una nota o un silencio
    energia_segmento=dot(y(i:j),y(i:j))/(paso-1); %energia del segmento

    %calcula la nota si no se trata de un segmento silencioso
    if energia_segmento>=E*0.1
        %calcula la fft de cada segmento
        Y=abs(fft(y(i:j)));
    end
end

```

```

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

%comparacion con la señal 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 señal
    %a analizar
    m(nf+1)=dot(armonicof(nf,1:T/2),Y);
end

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

%si se trata del primer segmento no silencioso *****(caso 1)*****
if k==1
    %960=tick por negra 2=negra por segundos (tempo)
    segmento=seg*960*2;

    %se expresa la duracion en formato delta time
    duracion=set_delta(segmento);

    %evento MIDI encender nota en el canal 1
    evento(1)=144;

    %activa la nota k
    evento(2)=nota(k);

    %velocidad de 127
    evento(3)=127;

    %se agraga la duracion en formato delta time al vector evento
    for l=4:3+length(duracion)
        evento(l)=duracion(l-3);
    end

%si no es el primer segmento sonoro se realiza un procedimiento similar
%a lo anterior pero no se agraga el evento encender nota en el canal 1

%si la nota actual es igual a la anterior y no es el primer
%segmento sonoro se realiza el siguiente proceso
%se va acumulando la duracion en la variable segmento
elseif nota(k)==nota(k-1)
    L=l-length(duracion)+1;
    segmento=segmento+seg*960*2;
    duracion=set_delta(segmento);
    u=1;

```

```

        for l=L:L+length(duracion)-1
            evento(l)=duracion(u);
            u=u+1;
        end

%si la nota actual es diferente a la anterior se realiza el siguiente
%proceso
elseif nota(k)~=nota(k-1)

    %si la nota actual es diferente a la anterior, no es el primer
    %segmento sonoro y la nota anterior es un silencio***(caso 2)
    if nota(k-1)~=-5

        %se apaga la nota anterior
        evento(l+1)=nota(k-1);
        evento(l+2)=0;
        evento(l+3)=0;

        %se activa la nota k con velocidad de 127
        evento(l+4)=nota(k);
        evento(l+5)=127;
        u=1;
        L=l+6;

        %se inicializa la variable segmento que contiene la
        %duracion en ticks
        segmento=seg*960*2;
        duracion=set_delta(segmento);
        for l=L:L+length(duracion)-1
            evento(l)=duracion(u);
            u=u+1;
        end

        %%si la nota actual es diferente a la anterior, no es el primer
        %segmento sonoro y la nota anterior no es un silencio***(caso 3)
    else
        evento(l+1)=nota(k);
        evento(l+2)=127;
        u=1;
        L=l+3;
        segmento=seg*960*2;
        duracion=set_delta(segmento);
        for l=L:L+length(duracion)-1
            evento(l)=duracion(u);
            u=u+1;
        end
    end
end
end
%si es un segmento silencioso
else
    %si no se ha detectado nign segmento sonoro
    if k==1

```

```

        nota(k)=-5;
        k=0;

%si la nota anterior es un silencio y ya se ha detectado algun
%segmento sonoro
elseif nota(k-1)==-5
    L=1-length(duracion)+1;
    %960=tick por negra 2=negra por segundos (tempo)
    segmento=segmento+seg*960*2;
    duracion=set_delta(segmento);
    u=1;
    for l=L:L+length(duracion)-1
        evento(l)=duracion(u);
        u=u+1;
    end
    nota(k)=-5;

%si la nota anterior no es un silencio y ya se ha detectado algun
%segmento sonoro
elseif nota(k-1)~-5
    segmento=seg*960*2;
    duracion=set_delta(segmento);
    evento(l+1)=nota(k-1);
    evento(l+2)=0;
    L=l+3;
    u=1;
    for l=L:L+length(duracion)-1
        evento(l)=duracion(u);
        u=u+1;
    end
    nota(k)=-5;
end
end

k=k+1;

end

%si la ultima nota detectada no es un silencio se apaga
if nota(k-1)~-5
evento=[evento nota(k-1) 0 0];
end

%se llama la funcion num2midi para escribir el archivo midi con los datos
%del vector evento
num2midi(evento,nombreM);

```