

APÉNDICE

En el apéndice se muestran las tablas de resultados obtenidos para cada una de las 76 señales analizadas con el programa, así como el código del programa escrito en Matlab.

1.- Tablas con resultados.

➤ NIÑOS SANOS

HEALTHY	FUNDAMENTAL FREQUENCY			
	F0 Mean (Hz)	F0 Max(Hz)	F0 Min(Hz)	Std F0 (Hz)
Case 1	488	918	386	120
Case 2	509	918	344	155
Case 3	467	918	174	86
Case 4	422	887	170	73
Case 5	475	837	204	156
Case 6	504	918	287	119
Case 7	489	900	228	143
Case 8	494	569	363	33
Case 9	535	837	209	109
Case 10	430	473	259	40
Case 11	442	918	220	137
Case 12	507	744	335	56
Case 13	533	918	219	145
Case 14	560	918	336	176
Case 15	533	918	218	145
Case 16	581	918	304	104
Case 17	532	918	176	110
Case 18	509	911	377	85
Case 19	577	918	222	110
Case 20	569	918	291	102
PROM.				110,2
MAX.	581	918	386	
MIN.	422	473	170	

HEALTHY	RESONANCES FREQUENCIES					
	F1(Hz)	F2(Hz)	F3(Hz)	Std1(Hz)	Std2(Hz)	Std3(Hz)
Case 1	1358	3710	5341	488	1325	2166
Case 2	1411	3656	5714	531	1097	1760
Case 3	1360	4294	6719	474	329	630
Case 4	1317	4367	6705	433	604	797
Case 5	1452	4424	6100	459	885	1552
Case 6	1426	4685	6472	392	557	1748
Case 7	1404	4628	6377	379	654	1613
Case 8	2177	4673	5498	383	510	2733
Case 9	1191	3707	6440	459	3707	6440
Case 10	1882	4106	6882	231	303	1208
Case 11	1563	4038	6175	447	652	2334
Case 12	2298	4750	2905	482	713	3108
Case 13	1227	3366	5302	446	223	554
Case 14	1190	3327	5509	172	747	1384
Case 15	1227	3366	5302	229	446	554
Case 16	1098	3072	5069	357	566	1233
Case 17	1112	3200	5150	287	346	1139
Case 18	1147	3204	5357	182	444	748
Case 19	1409	3476	5412	711	818	813
Case 20	1146	3198	5457	130	401	702

PROM				383,6	766,35	1660,8
MAX	2298	4750	6882			
MIN	1098	3072	2905			

HEALTHY	VOICED PARTS					Power Max (dB)
	Num Time	Max Durat(s)	Min Durat(s)	Mean Durat(s)	Total Durat(s)	
Case 1	5/5	1,200	0,080	0,664	3,320	-2,53
Case 2	3/5	2,520	0,140	1,393	4,180	-2,18
Case 3	3/5	1,100	0,100	0,727	2,180	-5,49
Case 4	5/5	0,980	0,020	0,500	2,500	-4,29
Case 5	7/5	0,840	0,080	0,397	2,780	-3,47
Case 6	4/5	2,320	0,060	0,770	3,080	-3,94
Case 7	5/5	1,820	0,040	0,676	3,380	-2,58
Case 8	4/5	2,720	0,100	0,850	3,400	-2,37
Case 9	5/5	1,060	0,420	0,660	3,300	-2,50
Case 10	4/5	1,740	0,020	0,840	3,360	-5,24
Case 11	7/5	2,320	0,020	0,594	4,100	-2,67
Case 12	1/5	3,000	3,000	3,000	3,000	-6,39
Case 13	5/5	1,920	0,160	0,688	3,240	-1,69
Case 14	3/5	2,580	0,060	1,067	3,200	-3,40
Case 15	5/5	1,920	0,160	0,688	3,340	-3,61
Case 16	4/5	2,200	0,180	0,755	3,020	-0,52
Case 17	5/5	1,680	0,020	0,536	2,680	-3,84
Case 18	5/5	1,360	0,100	0,864	4,320	-2,58
Case 19	7/5	2,360	0,020	0,614	4,300	-1,35
Case 20	4/5	1,940	0,120	0,990	3,960	-2,60

PROM						-3,25
MAX		3,000		3,000	4,320	-0,52
MIN		0,840		0,397	2,180	-6,39

➤ NIÑOS CON HIPOTIROIDISMO.

IPOTIROIDEI	FUNDAMENTAL FREQUENCY			
	F0 (Hz)	F0 Max(Hz)	F0 Min(Hz)	Std F0(Hz)
CASE 1	603	918	417	130
CASE 2	644	918	289	190
CASE 3	597	918	420	161
CASE 4	651	918	171	167
CASE 5	448	918	155	202
CASE 6	509	918	191	247
CASE 7	614	918	197	235
CASE 8	717	918	172	262
CASE 9	731	918	241	267

PROM				206,78
MAX	731	918	420	
MIN	448	918	155	

IPOTIROIDEI	RESONANCES FREQUENCIES					
	F1(Hz)	F2(Hz)	F3(Hz)	Std1(Hz)	Std2 (Hz)	Std3(Hz)
CASE 1	1357	3477	5458	421	682	1772
CASE 2	1214	3706	5188	806	1090	2054
CASE 3	1511	3543	5395	681	1031	1869
CASE 4	799	2374	4745	249	1042	1529
CASE 5	1305	4361	6747	493	804	1948
CASE 6	2030	4043	5220	918	1304	2574
CASE 7	1328	3618	4023	1044	1630	2496
CASE 8	2090	4413	4625	1032	1469	3060
CASE 9	2498	4517	4467	1227	1687	3240

PROM				763,44	1193,22	2282,44
MAX	2498	4517	6747			
MIN	799	2374	4023			

IPOTIROIDEI	VOICED PARTS					Power Max (dB)
	Num Time	Max Durat(s)	Min Durat(s)	Mean Durat(s)	Total Durat(s)	
CASE 1	6/5	1,780	0,020	0,570	3,420	-2,79
CASE 2	9/5	1,320	0,020	0,496	4,460	-2,87
CASE 3	10/5	1,140	0,040	0,278	2,780	-3,76
CASE 4	9/5	0,380	0,020	0,222	2,000	-3,11
CASE 5	9/5	1,640	0,020	0,318	2,540	-5,06
CASE 6	10/5	0,680	0,020	0,266	2,660	-6,21
CASE 7	12/5	0,920	0,020	0,217	2,600	-3,13
CASE 8	7/5	0,480	0,140	0,291	2,040	-4,82
CASE 9	10/5	0,620	0,020	0,150	1,500	-4,72

PROM						-4,05
MAX		1,780		0,570	4,460	-2,79
MIN		0,380		0,150	1,500	-6,21

➤ NIÑOS CON PROBLEMAS DE AUDICIÓN .

DEAF	FUNDAMENTAL FREQUENCY			
	F0 (Hz)	F0 Max(Hz)	F0 Min(Hz)	Std F0(Hz)
CASE 1	445	811	157	139
CASE 2	393	651	114	110
CASE 3	719	1000	247	160
CASE 4	582	890	146	235
CASE 5	616	918	306	211
CASE 6	645	983	221	241
CASE 7	401	709	39	157
CASE 8	583	1000	77	174
CASE 9	433	1000	64	119
CASE 10	450	1000	82	85
CASE 11	632	1143	196	234
CASE 12	564	1000	107	163
CASE 13	708	959	267	221
CASE 14	553	959	202	143
CASE 15	484	959	274	135
CASE 16	577	959	256	177
CASE 17	700	959	360	193
CASE 18	714	959	268	202
CASE 19	687	959	181	253
CASE 20	462	959	210	153
CASE 21	497	788	174	65
CASE 22	541	711	125	101
CASE 23	617	959	216	210
CASE 24	740	959	361	188
CASE 25	537	959	180	185
CASE 26	501	959	199	170
CASE 27	452	959	260	71
CASE 28	510	788	230	86
CASE 29	512	959	317	139
CASE 30	535	959	314	184
CASE 31	483	568	224	48
CASE 32	430	882	250	92
CASE 33	412	625	195	88
CASE 34	410	959	163	133
CASE 35	402	495	261	50
CASE 36	419	843	300	76
CASE 37	509	959	276	166

CASE 38	546	947	275	100
CASE 39	499	877	239	96
CASE 40	514	959	268	105
CASE 41	451	508	312	71
CASE 42	496	959	272	101
CASE 43	529	919	267	121
CASE 44	528	913	196	196
CASE 45	457	834	112	162
CASE 46	641	959	305	192
CASE 47	440	686	266	76

PROM				144,19
MAX	740	1143	361	
MIN	393	495	39	

DEAF	RESONANCES FREQUENCIES					
	F1(Hz)	F2(Hz)	F3(Hz)	Std1(Hz)	Std2(Hz)	Std3(Hz)
CASE 1	894	2150	3499	143	400	859
CASE 2	741	1621	2915	147	457	396
CASE 3	890	2501	2771	143	534	1517
CASE 4	788	2174	3319	103	501	1062
CASE 5	823	2442	1751	79	286	1731
CASE 6	862	2451	1455	88	560	1537
CASE 7	764	2158	2311	201	623	1302
CASE 8	802	1951	2204	160	474	1114
CASE 9	961	2164	1551	403	645	1405
CASE 10	792	1844	2145	180	485	1196
CASE 11	1065	2136	2182	407	595	1357
CASE 12	836	2116	2319	225	521	1196
CASE 13	1034	2662	3497	380	497	1243
CASE 14	1119	2656	2931	329	701	1636
CASE 15	1059	2539	3501	358	542	1204
CASE 16	1045	2539	3512	274	412	1046
CASE 17	1202	2582	3328	580	694	1367
CASE 18	965	2317	3761	263	233	575
CASE 19	959	2411	3267	401	60	1298
CASE 20	994	2805	3577	220	504	1162
CASE 21	1128	3042	3619	544	436	1229

CASE 22	1039	2444	3647	276	395	468
CASE 23	1008	2403	3471	152	301	589
CASE 24	949	2557	3607	358	341	840
CASE 25	1231	2737	3159	312	694	1271
CASE 26	1157	2850	3830	326	447	741
CASE 27	1588	3004	3417	383	45	1683
CASE 28	1085	2507	3225	420	702	1303
CASE 29	1333	2864	3480	364	565	1365
CASE 30	902	2763	3121	172	442	1634
CASE 31	883	3287	4618	294	1336	2430
CASE 32	659	3555	4692	157	1542	2677
CASE 33	1012	2998	4305	338	1465	2324
CASE 34	1386	4085	2312	456	1521	2985
CASE 35	847	2608	5146	343	558	1061
CASE 36	822	2187	4637	373	1315	2329
CASE 37	1137	2684	4775	519	763	2115
CASE 38	822	2486	495	235	550	1008
CASE 39	804	2765	4736	216	719	1282
CASE 40	1160	2750	4492	397	959	1992
CASE 41	1098	2631	4753	389	545	1593
CASE 42	974	2713	4931	287	549	1045
CASE 43	997	2647	5013	418	653	967
CASE 44	834	2653	4623	104	713	1649
CASE 45	896	2395	3176	299	930	2595
CASE 46	899	2525	4436	204	256	1380
CASE 47	822	2567	4800	196	169	1320

PROM				289,70	609,26	1405,91
MAX	1588	4085	5146			
MIN	659	1621	495			

DEAF	VOICED PARTS					Power Max (dB)
	Num/Time	Max Durat(s)	Min Durat(s)	Mean Durat(s)	Total Durat(s)	
CASE 1	2/5	2,120	0,740	1,430	2,860	-3,71
CASE 2	2/5	0,960	0,020	0,490	0,980	-4,66
CASE 3	2/5	3,320	0,060	1,690	3,380	-3,04
CASE 4	2/5	2,260	0,760	1,510	3,020	-2,75
CASE 5	8/5	0,640	0,020	0,300	2,400	-1,62
CASE 6	5/5	2,180	0,020	0,616	3,080	-3,62
CASE 7	1/5	1,780	1,780	1,780	1,780	-3,10
CASE 8	8/5	0,540	0,020	0,205	1,640	-2,90
CASE 9	9/5	0,800	0,020	0,253	2,280	-3,12
CASE 10	8/5	1,000	0,060	0,343	2,740	-1,18
CASE 11	7/5	1,040	0,040	0,563	3,940	-2,98
CASE 12	10/5	0,900	0,020	0,252	2,520	-1,73
CASE 13	5/5	2,460	0,020	0,960	4,800	-1,81
CASE 14	11/5	0,820	0,020	0,307	3,380	-2,57
CASE 15	6/5	1,400	0,020	0,627	3,760	-3,07
CASE 16	5/5	1,000	0,220	0,484	2,420	-2,50
CASE 17	4/5	1,220	0,080	0,670	2,680	-4,35
CASE 18	4/5	1,040	0,020	0,460	1,840	-3,80
CASE 19	8/5	1,040	0,020	0,372	2,980	-3,32
CASE 20	8/5	0,840	0,020	0,370	2,960	-2,67
CASE 21	10/5	0,740	0,020	0,196	1,960	-1,03
CASE 22	5/5	0,420	0,020	0,236	1,180	-1,77
CASE 23	3/5	1,480	0,080	0,553	1,660	-3,78
CASE 24	4/5	1,120	0,120	0,580	2,320	-2,65
CASE 25	5/5	1,020	0,060	0,372	1,860	-1,03
CASE 26	8/5	1,260	0,040	0,550	4,400	-4,07
CASE 27	5/5	2,100	0,140	0,784	3,920	-2,92
CASE 28	8/5	1,380	0,020	0,483	3,860	-1,65
CASE 29	7/5	1,640	0,040	0,597	4,180	-4,14
CASE 30	7/5	1,980	0,040	0,640	4,480	-2,64
CASE 31	3/5	1,300	0,080	0,667	2,000	-1,22
CASE 32	6/5	0,320	0,060	0,153	0,920	-2,01
CASE 33	6/5	0,740	0,140	0,357	2,140	-0,55
CASE 34	12/5	0,620	0,020	0,168	2,020	-1,92
CASE 35	8/5	0,540	0,040	0,163	1,300	-2,47
CASE 36	6/5	1,160	0,140	0,607	3,640	-0,90
CASE 37	7/5	1,580	0,020	0,437	3,060	-2,03
CASE 38	6/5	0,980	0,020	0,330	1,980	-2,16
CASE 39	5/5	0,400	0,200	0,300	1,500	-1,98
CASE 40	9/5	1,240	0,020	0,369	3,320	-2,62

CASE 41	4/5	1,520	0,440	0,790	3,190	-2,32
CASE 42	8/5	1,720	0,020	0,375	3,000	-2,85
CASE 43	6/5	1,300	0,160	0,683	4,100	-1,47
CASE 44	4/5	0,260	0,120	0,185	0,740	-2,95
CASE 45	4/5	1,740	0,020	0,650	2,600	-2,54
CASE 46	5/5	1,120	0,100	0,580	2,900	-2,67
CASE 47	4/5	0,900	0,260	0,525	2,100	-3,16

PROM						-2,55
MAX		3,320		1,780	4,800	-0,55
MIN		0,260		0,153	0,740	-4,66

2.- Código Matlab.

```
%      PROGRAMA PARA EL ANÁLISIS Y COMPARACIÓN
%      DE SEÑALES VOCALES PRE-POST INTERVENCIÓN/TRATAMIENTO
%
%      parametros de ingreso: nombre de la señal, tipo paciente
%      nombre: path completo del archivo *.wav
%      tipo:
%      1 = adulto masculino
%      2 = adulto femenino
%      3 = neonato
%      4 = cantante

function singolo(file,tipo)

close all;

f=statusbar('Work in progress...');
p=0;
statusbar(p,f);
tagpre=300;

%Bucle para hallar el nombre de la señal
j=1;
i=length(file)-4;
while i~=0
    if file(i)~='\ '
        file1(j)=file(i);
        i=i-1;
        j=j+1;
    else
        i=0;
    end
end

j=length(file1);
for i=1:length(file1)
    nombre(i)=file1(j);
    i=i+1;
    j=j-1;
end

% Análisis de la señal
[s_TP,s_P,s_nne,s_asselimite,p1,voiced]=F0_noise(file,tipo,tagpre);

%Impresión de los resultados
plotsingolo(s_TP,s_P,s_nne,s_asselimite,p1,tipo,voiced);

p=p+0.46;
statusbar(p,f);
formanti_reali(file,tipo,tagpre);
p=p+0.46;
statusbar(p,f);
```

```

%Figura vacía para tipo distinto de neonato para la parte voiced
if tipo~=3
    figure(600)
    subplot(3,1,2)
    title(' ')
    axis off
    hold off
end

%Figura con la parte de RESULT I referente al nombre de la señal
figure(1)
subplot(20,1,2);
title(['SIGNAL => ',nombre]);
axis off;
hold off;

% Las figuras se guardan en formato .png
numfigure=2;
for i=1:numfigure
    p=p+0.01;
    print (['-f',num2str(300+i)],'-dpng',['./Risultati-S/img/',
        'SINGLE-',num2str(i),'.png']);
    statusbar(p,f);
end

print ( '-f11', '-dpng', ['./Risultati-S/img/', 'SINGLE-3.png']);
print ( '-f12', '-dpng', ['./Risultati-S/img/', 'SINGLE-4.png']);
print ( '-f600', '-dpng', ['./Risultati-S/img/', 'SINGLE-5.png']);
print ( '-f1', '-dpng', ['./Risultati-S/img/', 'SINGLE-6.png']);
print ( '-f2', '-dpng', ['./Risultati-S/img/', 'SINGLE-7.png']);
p=p+0.01;
delete *.mat;
statusbar(p,f);
close all
delete(f)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%          FUNCIÓN PARA OBTENER LA FRECUENCIA FUNDAMENTAL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function[Tempo_Pitch_2,Pitch,nne,asselimite,p,voicedT]=F0_noise(nome,
                                                                    tipo,tagprespost)

[s,Fs]=wavread(nome);
s=s(:,1);
if tipo==1
    Liminf=50;
    Limsup=250;

```

```

else if tipo==2
    Liminf=100;
    Limsup=350;

else if tipo==3
    Liminf=150;
    Limsup=900;

else if tipo==4
    Liminf=50;
    Limsup=1200;
end
end
end

end

% Parámetros definidos por defecto
N_NNE=32768;    % Número de líneas para calcular la FFT
dip=5;          % Número de líneas para el DIP
Bw=2;           % Ancho de banda de la ventana Hamming
Intorno=10;     % Tolerancia en el cálculo de la F0

Lung_s=length(s);
tempoTotale=Lung_s/Fs;
tempo=[1:Lung_s]/Fs;

%=====
%                               Método SIFT
%=====

%Normalización de la señal
s=s-mean(s);
s=s/max(s);

[Pitch_1,Tempo_Pitch_1,voicedT,Dim_frame_1]=Sift2(s,Fs,Liminf,Limsup);

%=====
%                               Método AMDF
%=====

voiced=voicedT(1,:);
%Tolerancia para el cálculo de la F0 en Hz
Gap=10;
%Número de muestras de F0 según el método SIFT
maxframe=length(Pitch_1);
cont=1;
time=1/Fs;
stop=0;

for i=1:maxframe-1,

```

```

%Se excluyen los bloques unvoiced
if((Pitch_1(i)==0)|(i==maxframe-
1&Pitch_1(maxframe)==0)|(voiced(i)==0))

%Tamaño del primer bloque
if cont==1
    Dim_frame_2(cont)=Dim_frame_1;
    Pitch_2(cont)=0;
    Pitch(cont)=0;
    start=stop+1;
    stop=stop+Dim_frame_2(cont);
    Tempo_Pitch_2(cont)=start/Fs;
    Tempo_frame(cont)=stop/Fs;
    time=Tempo_frame(cont);
    cont=cont+1;

%Tamaño de los bloques sucesivos
else
    blocco=round(Tempo_Pitch_1(i)*Fs);
    if stop>=blocco
        if (blocco-start)<0.005*Fs
            cont=cont-1;
            start=start-Dim_frame_2(cont-1);
        end
        stop=blocco-1;
        Tempo_frame(cont-1)=stop/Fs;
        Dim_frame_2(cont-1)=stop-start+1;
    end
    Dim_frame_2(cont)=Dim_frame_1;
    Pitch_2(cont)=0;
    Pitch(cont)=Pitch_1(i);
    start=stop+1;
    stop=stop+Dim_frame_2(cont);
    Tempo_Pitch_2(cont)=start/Fs;
    Tempo_frame(cont)=stop/Fs;
    time=Tempo_frame(cont);
    cont=cont+1;
end

else

    while ((time<Tempo_Pitch_1(i+1))|(i==maxframe-1 &
time<tempoTotale))
        if (i==1)
            Pitch_min=min(Pitch_1(i), Pitch_1(i+1)); % Hz
            Pitch_max=max(Pitch_1(i), Pitch_1(i+1)); % Hz
            if Pitch_min==0
                Pitch_min=Liminf;
            end
            Liminf=Pitch_min-Gap; % Hz
            Limsup=Pitch_max+Gap; % Hz
        else
            Pitch_min=min(Pitch_1(i-1:i+1)); % Hz
            Pitch_max=max(Pitch_1(i-1:i+1)); % Hz
            if Pitch_min==0
                Pitch_min=Liminf;
            end
            Liminf=Pitch_min-Gap; % Hz
            Limsup=Pitch_max+Gap; % Hz
        end
    end
end

```

```

        Dim_frame_2(cont)=round((2*Bw*N_NNE)/((Liminf*N_NNE/Fs)-(
(dip+1))));
        start=stop+1;
        stop=stop+Dim_frame_2(cont);
        if stop>Lung_s
            if (Lung_s-start+1)<0.005*Fs
                cont=cont-1;
                start=start-Dim_frame_2(cont);
            end;
            stop=Lung_s;
            Dim_frame_2(cont)=stop-start+1;
        end

        s_frame=s(start:stop);
        length(s_frame);
        %Filtro wavelets
        [s_frame]=WaveCWT2(s_frame);
        %Uso del método AMDF
        [Pitch_2(cont)]=Amdf2(s_frame,Fs,Liminf,Limsup);
        Pitch(cont)=Pitch_2(cont);

        %Cálculo del ruido
        [nne(cont)]=NNE(s_frame, Fs, Pitch_2(cont), Bw, Intorno, N_NNE);
        Tempo_Pitch_2(cont)=start/Fs;
        Tempo_frame(cont)=stop/Fs;
        time=Tempo_frame(cont);
        cont=cont+1;
    end

end
end

%Cálculo de la frecuencia fundamental media
cont=cont-1;
Dim_frame_2=Dim_frame_2(1:cont);
count=1;
i=1;

if Pitch_2(1)~=0
    p(1)=Pitch_2(1);
    count=2;
end
for j=2:cont
    if Pitch_2(j)~=0
        if (j<cont & Pitch_2(j+1)==0)
            i=i+1;
        end
        if j==cont
            i=i+1;
        end
        p(count)=Pitch_2(j);
        count=count+1;
    end
end
count=count-1;
p=p(1:count);

```

```

%Figura con los armónicos y el espectrograma
asselimite=Lung_s/Fs;
figure(1+tagprespost)
hold on

%Se consideran ventanas todas iguales a la media de las calculadas
lf=round(median(Dim_frame_2));
lf2=round(lf/2);
tc=1/Fs;
freq=(0:lf2-1)/lf/tc;
lffft=floor(lf/3);
specgram(s,freq,Fs,lf,lffft)
title('Spectrogram ')
xlabel('Time [s]')
ylabel('Frequency [Hz]')
Armoniche=zeros(length(Pitch_2),40);
for k=1:40
    Armoniche(:,k)=k.*Pitch_2';
    plot(Tempo_Pitch_2,Armoniche(:,k),'k')
end

if tipo <= 2
    axis([0 tempoTotale 0 6000])
else
    axis([0 tempoTotale 0 8000])
end
hold off
colorbar('vert')

save (strcat(nome, '.dati_fondam', '.mat'))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%          FUNCIÓN PARA EL CÁLCULO DE F0 CON EL MÉTODO SIFT
%          Y EL CÁLCULO DE LAS PARTES VOICED
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function[Pitch,Tempo_Pitch,voicedT,Dim_frame]=Sift2(s,Fs,Liminf,Limsup)

%Filtro pasa banda con pasante 50Hz 1kHz y banda de transición de 50
%Hz
c = firpmord( [0 50 1000 1050], [0 1 0], [0.01 0.01 0.01], Fs,
'cell');
h = firpm(c{:});

Lung_s=length(s);
%Dimensión de las ventanas
Dim_frame=ceil(3/Liminf*Fs);

```

```

%Número de bloques en los que se divide la señal
maxframe=floor(Lung_s/Dim_frame);

R=7;

lim_inf=R*floor(Fs/Limsup);
lim_sup=R*ceil(Fs/Liminf);

for i=1:maxframe,
    maxframe-i+1;
    start=(i-1)*Dim_frame+1;
    stop=(i-1)*Dim_frame+Dim_frame;
    s1=s(start:stop);
    s1=s1-mean(s1);
% Estima del orden del IF con SVD y DME
    ord=ordine2(s1);
    v=svd(ord,0);
    N=dinsog2(diag(v));
    Nord(i)=N;

% Cálculo de los coeficientes de IF
    TH=ar(s1,N,'fb');
    a=polydata(TH);

% Cálculo del residuo
    res=conv(s1,a);
    res=res(ceil(N/2):ceil(N/2)+Dim_frame-1);

% Se filtra el residuo con el filtro paso banda
    res1=conv(res,h);
    res1=res1(ceil(length(h)/2):length(res1)-floor(length(h)/2));

% Cálculo de la correlación del residuo
    corr=xcorr(res1,'biased');
    corr=corr(ceil(length(corr)/2):length(corr));
    p=interp(corr,R);
    clear corr
    corr=p;
    clear p
    corr=corr/max(corr);

    [val,Pitch(i)]=max(corr(lim_inf:lim_sup));

    peak(i)=val;

    Tempo_Pitch(i)=start/Fs;
end

%Valor umbral para el cálculo V/UV
thresh=0.7;
if peak(1)<thresh
    voiced(1)=0;
else
    voiced(1)=1;
end
if peak(2)<thresh
    if peak(3)<thresh
        voiced(2)=0;
        voiced(1)=0;
    end
end

```

```

        else
            voiced(2)=0;
        end
    else
        if voiced(1)==0
            if (peak(3)>=thresh | peak(2)>=0.77)
                voiced(2)=1;
                voiced(1)=1;
            else
                voiced(2)=0;
            end
        else
            voiced(2)=1;
        end
    end
end

for i=3:maxframe-1
    if peak(i)>=thresh
        if (voiced(i-1)==0 & voiced(i-2)==1)
            if (peak(i+1)>=thresh|peak(i)>=0.77)
                voiced(i)=1;
                voiced(i-1)=1;
            else
                voiced(i)=0;
            end
        else
            if (voiced(i-1)==0 & voiced(i-2)==0)
                if (peak(i+1)>=thresh|peak(i)>=0.77)
                    voiced(i)=1;
                else
                    voiced(i)=0;
                end
            else
                voiced(i)=1;
            end
        end
    else
        if (voiced(i-1)==1 & voiced(i-2)==1)
            if peak(i)<0.525
                voiced(i)=0;
            else
                voiced(i)=1;
            end
        else
            voiced(i)=0;
        end
    end
end

end
voiced(maxframe)=voiced(maxframe-1);
voicedT=[voiced;TT];

```

```

%Figura con las partes voiced y unvoiced
figure(600)
hold on
TT=[1:maxframe]*(Dim_frame/Fs);
plot(TT,peak)
line=ones(maxframe,1);
line=line*thresh;
plot(TT,line)
plot(TT,voiced,'r')

```



```

lf2=round(ll/2);
tc=1/Fs;
fre=(0:lf2-1)/ll/tc;
lffft=floor(ll/3);
lp2=length(Dim_frame_2);
formanti=zeros(lp2,3);
f=zeros(lp2,3);
stop=0;
count=1;
k=1;
p=1;
for iii=1:lp2
    lp2-iii+1;
    start=stop+1;
    stop=stop+Dim_frame_2(iii);

    if Pitch_2(iii)~=0
        s=pian2(start:stop);
        %Normalización de la señal
        s=s-mean(s);
        s=s/max(s);
        na=round(Fs/1000);

        %Cálculo de la potencia media para cada ventana
        h=spectrum.yulear;
        Hpsd=psd(h,s,'Fs',Fs);
        pwr(p)=10*log10(avgpower(Hpsd));
        p=p+1;

        %Orden del modelo según el tipo de señal
        if tipo==1
            nafix=na;
        elseif tipo==2
            nafix=round(0.75*na);
        elseif tipo==3
            nafix=round(0.5*na);
        elseif tipo==4
            nafix=na;
        end
        if na<=12
            nafix=nafix*2;
        end

        %Cálculo del modelo AR con el método forward-backward
        arfix=ar([s'],nafix,'fb');
        sigma=arfix.NoiseVariance;

        %Coeficientes del modelo AR
        cfix=polydata(arfix);
        lf=round(Dim_frame_2(iii)/2);
        freq=(0:lf-1)/Dim_frame_2(iii)/tc;
        hhfix=zeros(1,lf);
        j=sqrt(-1);
        z1=exp((-j)*2*pi*freq*tc);
        for i=1:nafix+1
            hhfix=hhfix+cfix(i)*z1.^(i-1);
        end
    end
end

```

```

HFIX=hhfix.^(-1);
HHFIX=HFIX.*conj(HFIX)*tc*sigma;
PSD=10*log10(HHFIX/max(HHFIX));

stop1=round(8000*(length(freq))/(max(freq)));
aux=zeros(1,stop1);
cont=1;

if stop1<length(PSD)
    tope=stop1;
else
    tope=length(PSD);
end

%Se busca el máximo de la PSD para hallar las formantes
for j=1:tope-1
    if (PSD(j+1)-PSD(j))>0
        aux(j+1)=1;
    end
end

for k=1:stop1-1
    if aux(k)>aux(k+1)
        formanti(iii,cont)=freq(k);
        cont=cont+1;
    end
end

end

formanti=formanti(:,1:3);

if Pitch_2(iii)~=0
    f(count,:)=formanti(iii,:);
    count=count+1;
end

end

%Potencia máxima de entre todos los bloques
Maximapot=max(pwr);

delete(h);
count=count-1;
f=f(1:count,:);
media=mean(f);
sigma=std(f);
save (strcat(nome, '.dati_formanti', '.mat'));

s1=['mean F1=' num2str(media(1), '%4.1f') ' std F1='
    num2str(sigma(1), '%4.1f')];
s2=['mean F2=' num2str(media(2), '%4.1f') ' std F2='
    num2str(sigma(2), '%4.1f')];
s3=['mean F3=' num2str(media(3), '%4.1f') ' std F3='
    num2str(sigma(3), '%4.1f')];

```

```

%Figura con el espectrograma y las formantes
figure(2+tagprepost)
hold on
specgram(signal,fre,Fs,ll,lffft)
plot(Tempo_Pitch_2,formanti(:,1),'k.')
plot(Tempo_Pitch_2,formanti(:,2),'b.')
plot(Tempo_Pitch_2,formanti(:,3),'g.')
legend(s1,s2,s3)
title('Spectrogram and resonance frequencies')
xlabel('Time [s]')
ylabel('Frequency [Hz]')

```

```

if tipo==1
    axis([0 tempoTotale 0 6000])
elseif tipo==2
    axis([0 tempoTotale 0 6000])
elseif tipo==3
    axis([0 tempoTotale 0 10000])
elseif tipo==4
    axis([0 tempoTotale 0 8000])
end
hold off

```

```

%Figura con parte de la tabla RESULT I
figure(1)

```

```

subplot(20,1,9)
title('*RESONANCE FREQUENCIES')
axis off
subplot(20,1,11)
title(['Mean F1 = ',num2str(media(1),'%4.1f'), 'Hz ; Std F1 = ',
num2str(sigma(1),'%4.1f')])
axis off
subplot(20,1,12)
title(['Mean F2 = ',num2str(media(2),'%4.1f'), 'Hz ; Std F2 = ',
num2str(sigma(2),'%4.1f')])
axis off
subplot(20,1,13)
title(['Mean F3 = ',num2str(media(3),'%4.1f'), 'Hz ; Std F3 = ',
num2str(sigma(3),'%4.1f')])
axis off
subplot(20,1,15)
title('-----
-----')
axis off
subplot(20,1,17)
title(['*POWER MAX = ',num2str(Maximapot,'%4.3f'),'dB'])
axis off
subplot(20,1,19)
title('-----
-----')
axis off

```

```

% % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%
%                               FUNCIÓN PARA LA IMPRESIÓN DE LAS FIGURAS
%
% % % % % % % % % % % % % % % % % % % % % % % % % % % % %

```

```

function
plotsingolo(pre_TP,pre_P,pre_nne,pre_asselimite,p1,tipو,voicedT)

%Cálculo de las frecuencias voiced
%Inicialización de las variables
inicio=0;
final=0;
n=1;
part_voiced_ini=0;
part_voiced_fin=0;
pitch_medio1 =mean(p1);
TT=voicedT(2,:);
voiced=voicedT(1,:);
if tipo==3
    for i=1:(length(voiced))

        %Inicio de una parte voiced
        if (inicio==0)&(voiced(i)==1)
            part_voiced_ini=TT(i);
            inicio=1;
        end

        %Final de una parte voiced
        if (inicio==1)& (voiced(i)==0)
            part_voiced_fin=TT(i);
            total=part_voiced_fin-part_voiced_ini;
            inicio=0;
            tablavoiced(n,:)=[part_voiced_ini, part_voiced_fin,total];
            n=n+1;
            final=1;
        end
    end

    if final==0
        if voiced(length(voiced))==1
            part_voiced_fin=TT(length(TT));
            total=part_voiced_fin-part_voiced_ini;
            tablavoiced(n,:)=[part_voiced_ini, part_voiced_fin,total];
        else
            tablavoiced(n,:)=[0,0,0];
        end
    end
end

%Guarda en el archivo voiced los datos obtenidos
fid=fopen('Risultati-S\tablavoiced.txt','wt');
fprintf(fid,'Inicio   Fin       Duración\n');
fprintf(fid,'%6.4f   %6.4f   %6.4f\n',tablavoiced');
fclose(fid);

end

```

```

%Figura con la melodía
figure(11)

%Si no es de neonato se dibuja toda en azul
if tipo~=3
    plot(pre_TP,pre_P,'b*')
    ylabel('Frequency [Hz]')
    xlabel('Time [s]')
    title(['F0 - AMDF - Mean F0 =',num2str(pitch_medio1,'%4.1f'), ' ';
Std = ',num2str(std(p1),'%4.1f')])
    axis([0 max(pre_asselimite) min(pre_P) max(pre_P)])

%Para los neonatos dibuja la parte voiced en rojo y la unvoiced en
azul
else
    [N,M]=size(tablavoiced);
    l=1;
    k=1;
    u=1;
    p=1;
    dentro=0;
    hold on

%Cálculo de la F0 máx y mín dentro de partes voiced y dibujo de cada
parte
    for i=1:length(pre_P)

        if u<=N

            %Parte unvoiced
            if((pre_TP(i)<=tablavoiced(u,1))||(pre_TP(i)>=tablavoiced(N
,2)))&&(dentro==0)
                tiempoUV(k)=pre_TP(i);
                valorUV(k)=pre_P(i);
                k=k+1;
            end

            %Parte voiced
            if(pre_TP(i)>=tablavoiced(u,1))&&(pre_TP(i)<=tablavoiced(u,
2))
                tiempoV(l)=pre_TP(i);
                valorV(l)=pre_P(i);
                l=l+1;
                dentro=1;
            end

            %Primer valor unvoiced
            if((pre_TP(i)>=tablavoiced(u,2))&&(dentro==1))
                tiempoV(l)=pre_TP(i);
                valorV(l)=pre_P(i);
                hold on;
                for i=1:length(valorV)
                    if valorV(i)==0
                        plot(tiempoV(i),valorV(i),'b.')
                        valorV(i)=mean(pre_P);

```

```

        else
            plot(tiempoV(i),valorV(i), 'r*')
        end
    end
    l=1;
    u=u+1;
    dentro=0;

    %Frecuencia más y mín dentro de la parte voiced
    maxfreq(p)=max(valorV);
    minfreq(p)=min(valorV);
    p=p+1;
end

if (i==(length(pre_TP)))
    if dentro==1
        tiempoV(l)=pre_TP(i);
        valorV(l)=pre_P(i);
        hold on;
        for i=1:length(valorV)
            if valorV(i)==0
                plot(tiempoV(i),valorV(i), 'b.')
                valorV(i)=max(valorV);
            else
                plot(tiempoV(i),valorV(i), 'r*')
            end
        end
        maxfreq(p)=max(valorV);
        minfreq(p)=min(valorV);
    else
        tiempoUV(k)=pre_TP(i);
        valorUV(k)=pre_P(i);
    end

end

end

end

ylabel('Frequency [Hz]')
xlabel('Time [s]')
title(['F0 - AMDF - Mean F0 =',num2str(pitch_medio1, '%4.1f'), ' ';
        'Std = ',num2str(std(p1), '%4.1f')])
axis([0 max(pre_asselimite) min(pre_P) max(pre_P)])
plot(tiempoUV,valorUV, 'b.')
hold off;

end

%Figura con la parte de la tabla RESULT I
figure(1)
subplot(20,1,4)
title('*FUNDAMENTAL FRENQUENCY')
axis off;
subplot(20,1,6)

```

```

title(['Mean F0=', num2str(pitch_medio1, '%4.1f'), 'Hz ; Std =', num2str(std(p1), '%4.1f')])
axis off;

if tipo==3
subplot(20,1,7)
title(['Max F0=', num2str(max(maxfreq), '%4.1f'), 'Hz ; Min F0=', num2str(min(minfreq), '%4.1f'), 'Hz'])
axis off;

end
subplot(20,1,8)
title('-----');
axis off;
hold off;

%Figura con la tabla RESULT II
figure(2)
if tipo==3
subplot(20,1,2)
title('*VOICED PARTS')
axis off;

subplot(20,1,4)
title(['Star End Total'])
axis off;

for i=1:N
subplot(20,1,i+4)
title([num2str(tablavoiced(i,1), '%4.3f'), 's', ' ', num2str(tablavoiced(i,2), '%4.3f'), 's', ' ', num2str(tablavoiced(i,3), '%4.3f'), 's'])
axis off;
end

subplot(20,1,N+6)
title(['Max duration =', num2str(max(tablavoiced(:,3)), '%4.3f'), 's ; Min duration =', num2str(min(tablavoiced(:,3)), '%4.3f'), 's ; Mean duration =', num2str(mean(tablavoiced(:,3)), '%4.3f'), 's'])
axis off;

subplot(20,1,N+7)
title(['Total duration =', num2str(sum(tablavoiced(:,3)), '%4.3f'), 's ; Number Parts =', num2str(length(tablavoiced(:,3)), '%1.0f')])
axis off;

else
%Si no es neonato no aparece nada por pantalla
subplot(3,1,2)
title(' ')
axis off

end

```

```

%Figura con el jitter, Rap y NNE sólo para señales de no neonatos
if tipo~=3
[pre_jitter,premeanJ] = jitter (pre_P);
[pre_rap,premeanR] = rap (pre_P);

figure(12)
    asselimito = max(pre_asselimito);
    subplot(3,1,1)
    plot(pre_TP,pre_jitter*100,'-b.')
    axis([0 asselimito 0 5])
    title('JITTER - RAP - NNE')
    ylabel('Jitter %')
    xlabel(['~Jitter% = ',num2str(premeanJ*100,'%1.4f')], 'fontsize',9)

    subplot(3,1,2)
    plot(pre_TP,pre_rap*100,'-b.')
    axis([0 asselimito 0 5])
    ylabel('RAP %')
    xlabel(['~RAP% = ',num2str(premeanR*100,'%1.4f')], 'fontsize',9)

    subplot(3,1,3)
    indx1=find(pre_nne~=0);
    nnex1=pre_nne(indx1);
    snne1=length(pre_nne);
    stp21=length(pre_TP);
    tmin1=min(snne1,stp21);
    index1=isnan(nnex1)|isinf(nnex1);
    ind1=(nnex1(index1~=1));

    DysInd=mean(ind1);
    if DysInd<-15
        Vindex1='VOCE SANA';
    else
        Vindex1='VOCE ALTERATA';
    end

    plot(pre_TP(1:tmin1),pre_nne(1:tmin1),'-b.')

    axis([0 max(pre_asselimito) -30 5])
    ylabel('ANNE [dB]')
    xlabel([Vindex1,' ~NNE= ', num2str(mean(ind1),'%4.1f'),' dB', '
        Time [s]'], 'fontsize',9)

%Figura vacía para el caso de los neonatos
else
    figure(12)
    subplot(3,1,2)
    title(' ')
    axis off
    hold off
end

```