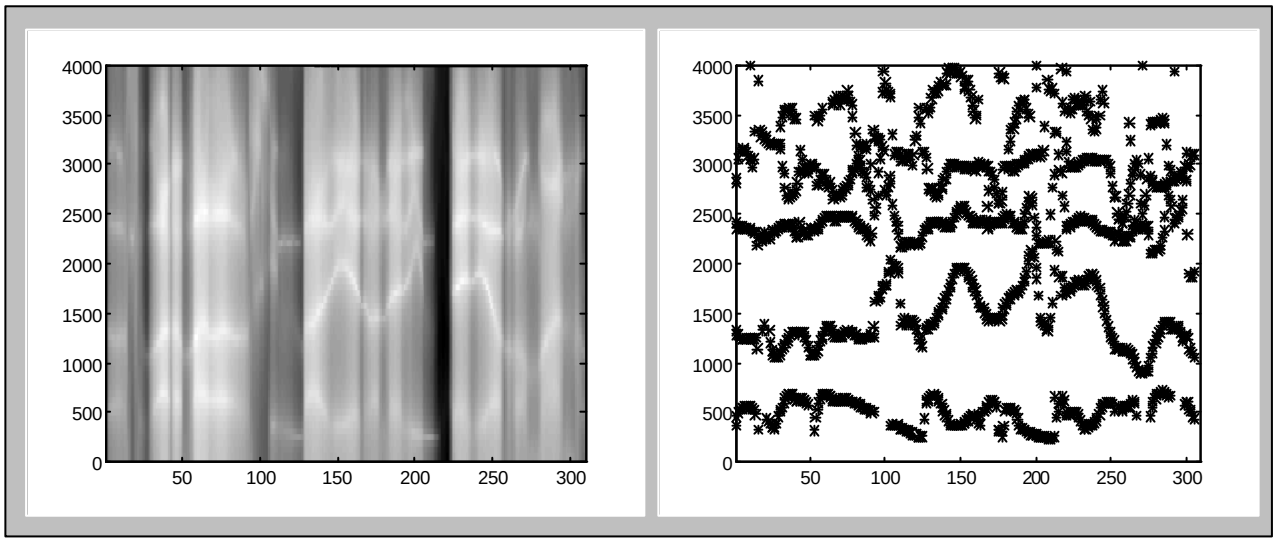
DEPARTAMENTO DE INGENIERÍA
ELECTRÓNICA Y COMUNICACIONES

CENTRO POLITÉCNICO SUPERIOR

UNIVERSIDAD DE ZARAGOZA

TECNOLOGIAS DE LA VOZ



PRÁCTICA II

Análisis localizado, métodos de estimación de pitch y formantes

ANALISIS LOCALIZADO
Métodos de estimación de pitch y formantes

I. Introducción

Una herramienta básica en el procesado de señales es la Transformada de Fourier. En la definición de la transformada de Fourier se supone un conocimiento de la señal para todo instante de tiempo y que cualquier propiedad o característica que estamos buscando mediante la Transformada de Fourier (espectro, resonancias, etc) se mantiene invariante para todo instante de tiempo. Como ya hemos visto, la señal de voz es una señal variante en el tiempo. Efectivamente, cada pocos milisegundos existe un cambio en la señal de voz. Esto hace que en la mayoría de las aplicaciones estemos mas interesados en conocer las propiedades de la señal de voz en intervalos pequeños de tiempo que en toda la señal. Este tipo de análisis lo denominaremos ***análisis localizado***.

En esta práctica estudiaremos el análisis localizado en el dominio temporal y frecuencial, aplicándolos al problema de la estimación de la frecuencia de pitch y la posición de los formantes. Esta práctica es un complemento al capítulo IV (Procesado digital de la señal de voz) del temario. Al finalizar esta práctica, el alumno debe consolidar:

1. Concepto de ***trama***.
2. Energía y Cruces por cero localizados en sonidos sonoros y sordos
3. Autocorrelación localizada en sonidos sonoros y sordos
4. Estimación de pitch y alisado no lineal
5. Transformada de fourier localizada. Espectrogramas de banda ancha y banda estrecha
6. Predicción lineal de la señal de voz
7. Estimación de la frecuencia central de formantes
8. Cepstrum

II. Estudio previo

Nota importante

El estudio previo consiste en una serie de ejercicios que hay que realizar **antes** de iniciar la práctica. Una copia del mismo hay que entregarla al iniciar la práctica en el laboratorio. Los conceptos teóricos necesarios para realizar la práctica han sido desarrollados en el tema IV de las clases teóricas.

1. Sonoridad y pitch

La figura 1 muestra la forma de onda de una realización por parte de un locutor masculino de la frase **"Los bárbaros"**. (La frecuencia de muestreo es 16000 Hz)

a. Definir los segmentos sonoros y los segmentos sordos.

b. Estimar, de la forma más precisa posible, la frecuencia de pitch, dibujando su evolución en una gráfica.

c. Repetir el apartado anterior con la forma de onda de la figura 2. Comentar las diferencias. (Se trata de la misma frase)

d. La figura 3 muestra la forma de onda y su transformada de Fourier de un segmento sonoro. Calcular el número de cruces por cero por muestra y relacionarlo con la distribución del espectro de la señal obtenido mediante una transformada de Fourier.

2. Alisado no lineal

En los sistemas de estimación de pitch automáticos suelen aparecer errores de estimación relacionados con frecuencias armónicas de la fundamental o con la frecuencia del primer formante. Estos errores pueden ser interpretados como un ruido impulsional, pues normalmente son errores que se cometen en una sola trama. Una forma de eliminar ruido impulsional es mediante la utilización de filtros de alisado no lineal.

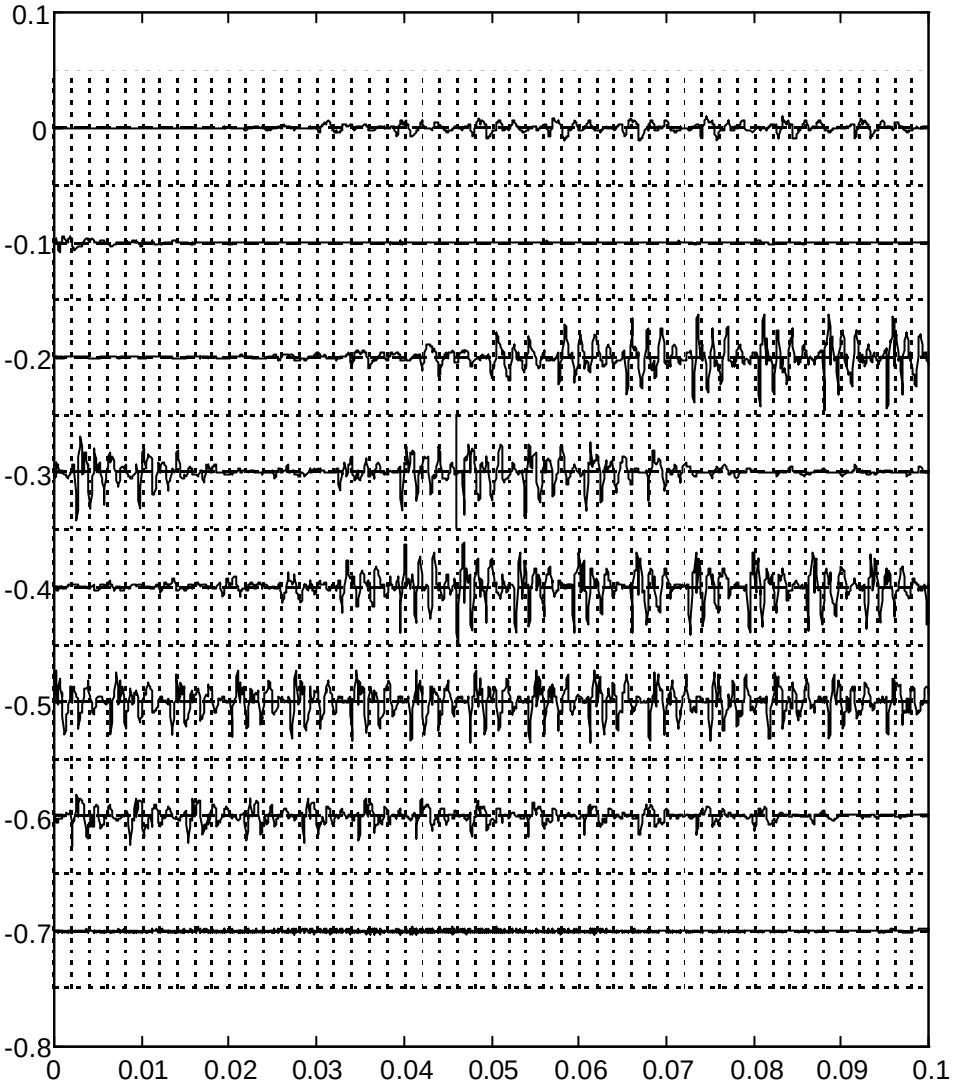


Figura 1. "**Los bárbaros**" apartado 1 (marcas cada 2 ms)

Los sistemas de alisado lineal se basan en la separación del contenido frecuencial no solapado. Para trabajar con sistemas de alisado no lineal es mejor considerar la señal como formada por dos tipos de señales, una alisada y otra ruidosa

$$x(n) = S[x(n)] + R(x(n)]$$

donde S[.] representa la parte alisada y R[.] la parte ruidosa de la señal x(n).

Un filtro no lineal que intenta separar ambas partes es el filtro de mediana, definido como :

$$y(n) = med\{x_{n-V}, \ldots, x_n, \ldots, x_{n+V}\}$$

donde el operador $med\{X_n\}$ es

$$med\{X_n\} = \begin{cases} x_{V+1} & si\ N = 2V+1 \\ (x_V + x_{V+1})/2 & si\ N = 2V \end{cases}$$

donde las muestras se han ordenado en orden ascendente de magnitud

$$x_1 \leq x_2 \leq \cdots \leq x_N$$

Normalmente estos filtros no dan un alisado suficiente de las componentes ruidosas y se suele combinar con un filtro lineal de alisado.

a. Dado un filtro de mediana de orden 3 (N=3), obtener la señal de salida cuando la entrada es la secuencia x(n)=[1 , 1 , 1 , 2 , 1 , 0.5 , 0.25 , 1 , 1 , 4 , 0.5 , 0.5 , 0 , 0 ,.....].

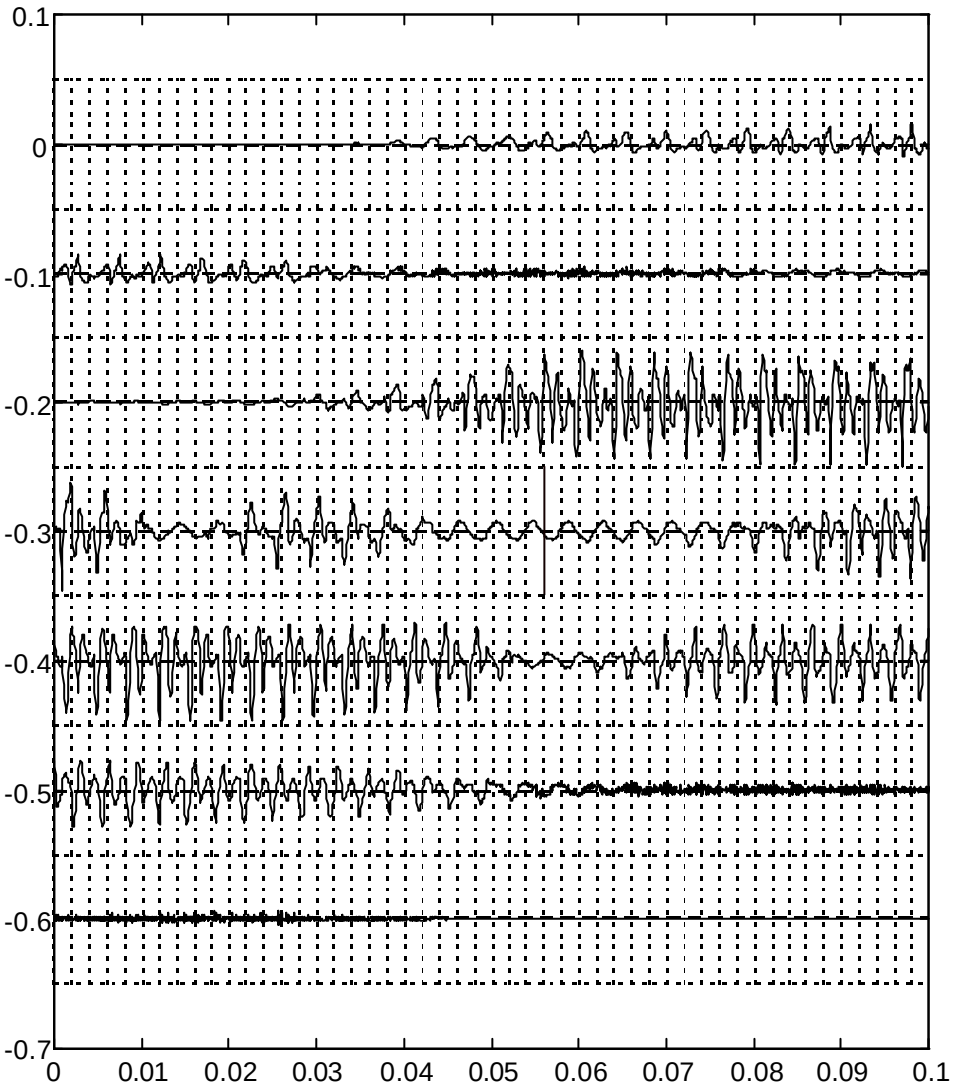


Figura 2. "**Los bárbaros**" apartado 2 (marcas cada 2 ms)

3. Autocorrelación y estimación de la frecuencia de pitch

Preparar una función en Matlab que permita estimar la evolución de la frecuencia de pitch a partir de la forma de onda, utilizando el método de la autocorrelación con center clipping. Para ello se dispone de las siguientes funciones :

R= **corloc** (muestras, N, M, CC, NFFT)
V= **voice** (muestras, N, M)
S= **pitch** (R, V, thV, fmin, fmax, fs)
Y= **medfilt1** (muestras, orden)

para mas información sobre las funciones ver apéndice A.

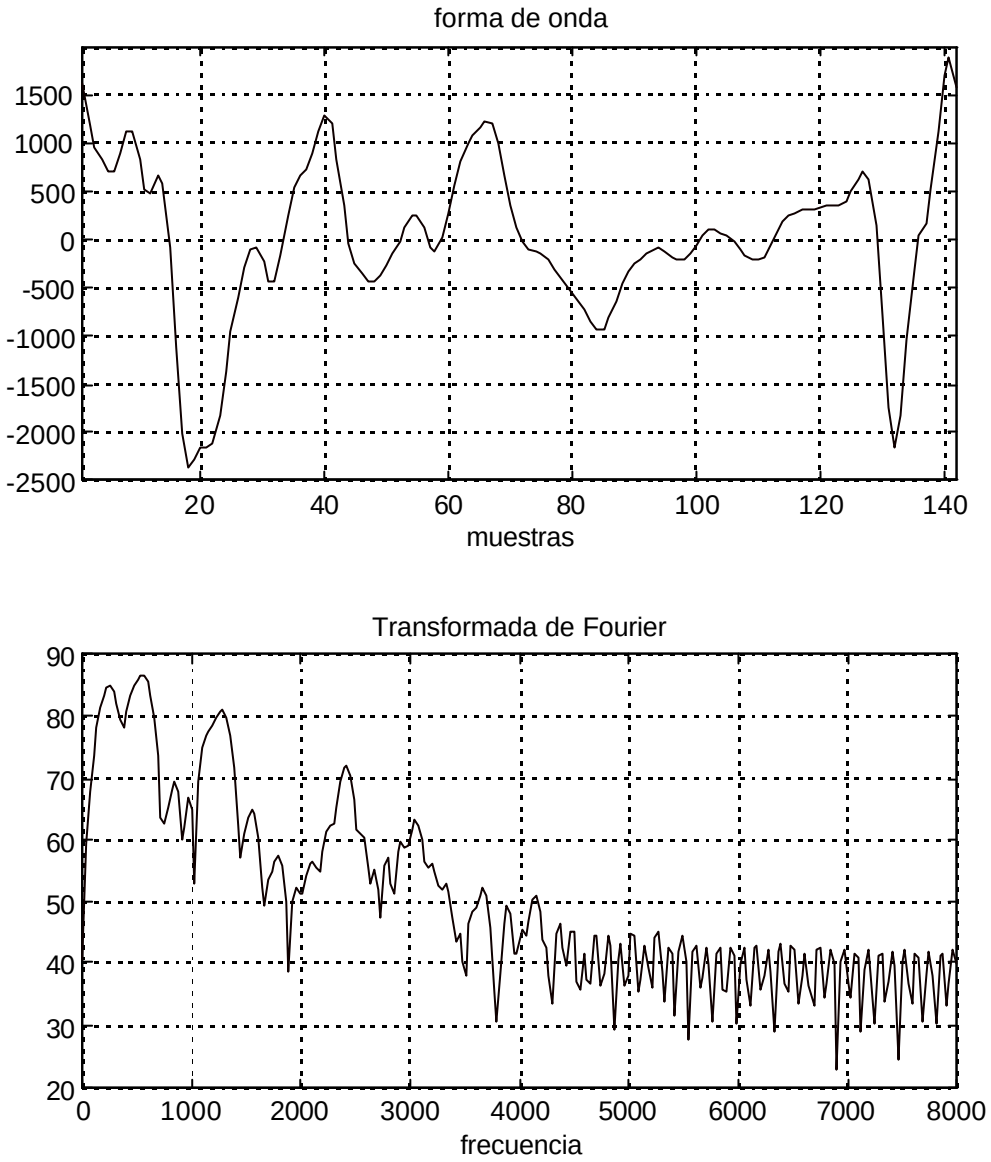


Figura 3. Forma de onda y transformada de Fourier. Apartado 1.d

4. Filtrado LPC y estimación de pitch

Como es conocido, la señal de error obtenida en el proceso de filtrado inverso obtenido a partir del análisis LPC de la señal de voz, tal y como se define en el diagrama de bloques de la figura 4 permite hacer una estimación de la frecuencia de pitch, al mismo tiempo que los coeficientes del filtro nos dan una estimación de la respuesta frecuencial del tracto vocal.

a. Indicar el proceso a seguir para determinar la frecuencia de pitch a partir del error de predicción.

b. En la figura 5 se muestra el proceso de reconstrucción de la señal de voz a partir del error de predicción y los coeficientes LPC. Explicar como funciona el proceso de reconstrucción.

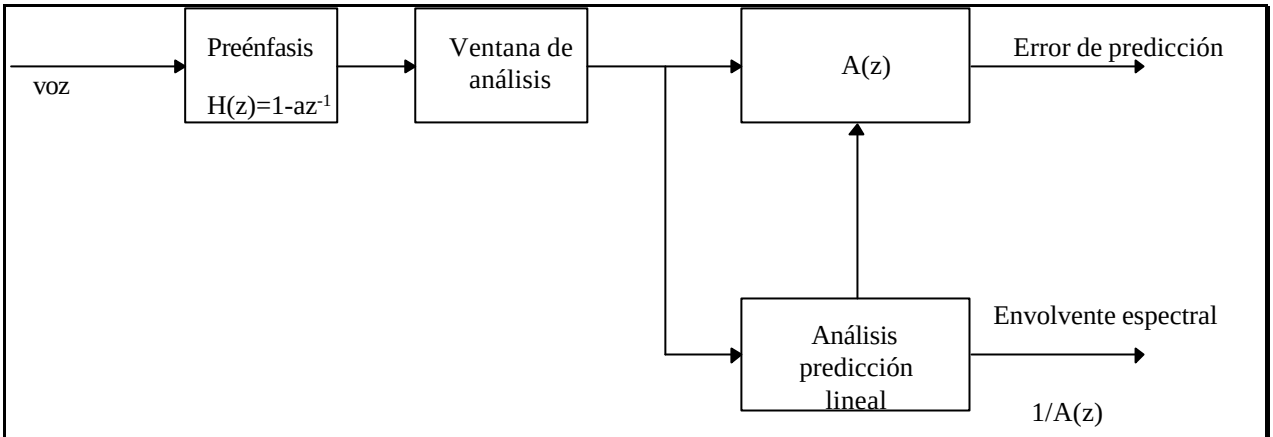


Figura 4

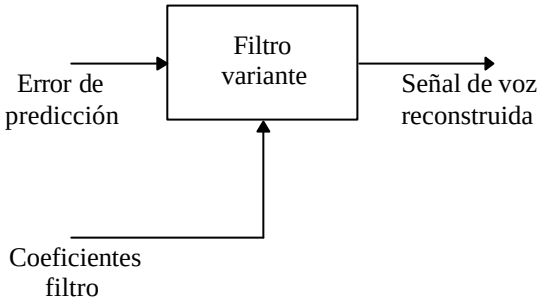


Figura 5

5. Estimación frecuencia central de los formantes.

A la hora de interpretar la información contenida en el filtro del predictor lineal, comúnmente se asume que las raices del polinomio A(z) (los polos del filtro del tracto vocal) son representativos de las frecuencias de los formantes para el segmento de voz con el que se han calculado los coeficientes del predictor. Así pues, los ángulos de las raices, expresados en frecuencia analógica, se utilizan en muchas ocasiones como una estimación de la frecuencia central de los formates. Utilizando las funciones **abs(), angle()** y **roots()** de Matlab, indicar el procedimiento a seguir para calcular la evolución de la frecuencia de los formantes asociados a la evolución del tracto vocal. Para tal fin, se ha construido la siguiente función que permite calcular los formantes entre las frecuencias 0 y 4000 Hz, considerando como formantes los ángulos de las raices complejas conjugadas con una magnitud mayor que un valor **0< m < 1**. La función que se dispondrá en el laboratorio es

F = forman(A, r, fs)

donde **A** es una matriz que contiene la evolución de los coeficientes del predictor a lo largo de la señal de voz, **r** umbral de la magnitud de las raices para se consideradas como formantes, **fs** la frecuencia de muestreo de la señal y **F** será una matriz con la frecuencia asociada a los ángulos de las raices.

III. Realización de la práctica

Nota importante

Para la realización de la práctica en el laboratorio es imprescindible haber realizado y presentado al inicio de la misma una copia del estudio previo.

Parte 1. Análisis temporal localizado

1. Utilizando la grabadora de sonidos, digitalizar la pronunciación de una frase de vuestra invención que contenga tanto sonidos sonoros, sordos y oclusivos. Realizar la digitalización a 8 kHz, 16 bits. Digitalizar la misma frase 2 veces dejando un intervalo de tiempo entre grabaciones de 1 minuto. Guardar estos ficheros en formato **.wav**.

2. Utilizando las funciones **eneloc( )** y **cerloc( )** representar la evolución de la energía y cruces por ceros localizados. (Representar los cruces por cero en términos de frecuencia).

3. Determinar los inicios y finales de los segmentos sonoros y sordos.

4. Comparar las evoluciones de estos dos parámetros sobre las dos señales digitalizadas. ¿En qué segmentos hay mayor variación?.

5. Aislar un segmento de 30 ms de voz sonora y otro segmento de voz sorda. Utilizando la función **corr( )** dibujar la función de autocorrelación de ambas señales. Comparar ambas y comentar las diferencias.

6. Utilizando la función definida en el estudio previo para la estimación de la evolución de la frecuencia de pitch, dibujar en una gráfica la evolución de la frecuencia de pitch de las dos señales digitalizadas. Utilizar segmentos de 30 ms con un desplazamiento de 15 ms. Fijar el umbral de center clipping en 0.75 y el umbral de sonoridad en 0.3. Comentar la diferencias entre ambas y los posibles errores de detección de pitch.

7. Estudiar la variación de la estimación de la frecuencia de pitch para diversos valores de center clipping y de umbral de sonoridad.

8. Leer las señales **nino.mat** y **roma.mat** (variables sam y samples respectivamente). El segmento **"los bárbaros"** utilizado en el estudio previo, apartados 1 y 2, se corresponden con un segmento de estas señales. Verificar la estimación de la frecuencia de pitch con la realizada en el estudio previo.

Parte 2. Análisis frecuencial localizado

Transformada de Fourier localizada.

9. Utilizando una de las señales digitalizadas en el apartado 1, dibujar el espectrograma para longitudes de la ventana de análisis de 10 ms, 30 ms y 50 ms y desplazamiento 10 ms en todos los casos. Para el cálculo del espectrograma utilizar la función **tfl( )** e **imagesc( )** para la representación (con **colormap( )** se pude cambiar la tabla de colores). Comentar las diferencias más notables entre ellas.

10. Repetir el apartado anterior con la señal del fichero roma.mat con longitud de ventana de análisis de 30 ms y desplazamiento de 10 ms con la señal original y la misma señal diezmada por un factor 2 (frecuencia de muestreo analógica 8000 Hz). Comentar las diferencias más notables.

Análisis LPC localizado.

11. Utilizando la misma señal del apartado 9, representar la evolución de la estimación de la respuesta frecuencial del tracto vocal utilizando la función **lpcloc( )** e **imagesc( )**. Utilizar un coeficiente de preénfasis de 0.95 y estudiar la variación de la estimación de la respuesta frecuencial con el orden del análisis LPC (utilizar orden 8, 12 y 16). Dejar fija la longitud de la ventana en 30 ms y un desplazamiento de 10 ms.

12. Comparar el espectrograma y la estimación de la envolvente para un sonido sonoro y otro sordo.

13. Para la señal utilizada en el apartado 11, calcular el error de predicción utilizando la función **erlpc( )** y representarla mediante **plot( )**. Comparar el error de predicción en segmentos sonoros y sordos. ¿Donde aparecen los picos del error de predicción ?. Escuchar la señal de error de predicción con la función **sound( )**. ¿Es inteligible ?.

14. Utilizando la función **sinlpc( )** y los datos adecuados, reconstruir la señal de voz y escucharla.

15. Representar la evolución de los formantes existentes entre 0 y 4000 Hz, calculados con la función definida en el apartado 7 del estudio previo. Comparar visualmente con el espectrograma de banda ancha calculado con la función **tfl( )** y con **lpcloc( )**.

Cepstrum localizado.

16. Calcular el cepstrum localizado mediante la función **ceploc( )** y representarlo utilizando **imagesc( )** para la señal utilizada en los apartados anteriores. Identificar el cepstrum relacionado con el tracto vocal y con la excitación.

17. Definir un procedimiento para estimar la evolución temporal de la respuesta frecuencial del tracto vocal a partir del cepstrum localizado. Dibujar la evolución utilizando **imagesc()** y compararla visualmente con la obtenida con la técnica de predicción lineal.

18. Particularizar para un instante de tiempo concreto y dibujar con **plot()** la respuesta frecuencia asociada al tracto vocal calculada utilizando la técnica del cepstrum para distintas longitudes de un liftado rectangular. ¿Cómo varía la estimación con la longitud de liftado?.

IV. Apendices

A. Funciones de Matlab

ANGLE Phase angle.

ANGLE(H) returns the phase angles, in radians, of a matrix with
complex elements.

See also ABS, UNWRAP.

ROOTS Find polynomial roots.

ROOTS(C) computes the roots of the polynomial whose coefficients
are the elements of the vector C. If C has N+1 components,
the polynomial is C(1)*X^N + ... + C(N)*X + C(N+1).

See also POLY.

FILTER Digital filter.

Y = FILTER(B, A, X) filters the data in vector X with the filter described by vectors A and B to create the filtered data Y. The filter is a "Direct Form II Transposed" implementation of the standard difference equation:

y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
- a(2)*y(n-1) - ... - a(na+1)*y(n-na)

[Y,Zf] = FILTER(B,A,X,Zi) gives access to initial and final conditions, Zi and Zf, of the delays.

See also FILTFILT in the Signal Processing Toolbox.

MEDFILT1 One dimensional median filter.

Y = MEDFILT1(X,N) returns the output of the order N, one dimensional median filtering of vector X. Y is the same length as X; for the edge points, zeros are assumed to the left and right of X.

For N odd, Y(k) is the median of X(k-(N-1)/2 : k+(N-1)/2).
For N even, Y(k) is the median of X(k-N/2 : k+N/2-1).

If you do not specify N, MEDFILT1 uses a default of N = 3.

MEDFILT1(X,N,BLKSZ) uses a for-loop to compute BLKSZ ("block size") output samples at a time. Use this option with BLKSZ << LENGTH(X) if you are low on memory (MEDFILT1 uses a working matrix of size N x BLKSZ). By default, BLKSZ == LENGTH(X); this is the fastest execution if you have the memory for it.

%function S=corloc(samples,N,M, th, NFFT)
%
% samples ..... muestras de la señal
% N ..... longitud de la ventana en muestras
% M ..... desplazamiento de la ventana en muestras
% th ..... umbral center clipping %
% NFFT ..... numero de puntos de la FFT
%
% S ..... autocorrelacion localizada
%
% calcula la autocorrelación localizada de una señal utilizando la
% ventana de hamming de N puntos desplazando M muestras
% con una FFT de NFFT muestras y aplicando un center clipping de CC %

% function S=voice(x,N,M)
%
% x ..... muestras de la señal
% N ..... longitud de la ventana de analisis en muestras
% M ..... desplazamiento de la ventana de analisis en muestras
%
% S ..... grado de sonoridad 0<= S <= 1

```
%  function S=pitch(A,sono,th,fmin,fmax,fs)
%
%       A ........ Autocorrelación localizada
%       S ........ grado de sonoridad
%       th ....... umbral de sonoridad 0<=th<=1
%       fmin ..... frecuencia de pitch minima a detectar
%       fmax ..... frecuencia de pitch maxima a detectar
%
%       S ........ evolución de la frecuencia de pitch

%function [A,G,S]=lpcloc(samples,preenf,N,M,P)
%
%       samples ........ muestras de la señal
%       preenf .......... coeficiente del filtro de preenfasis
%       N ............... longitud de la ventana en muestras
%       M ............... desplazamiento de la ventana en muestras
%       P ............... orden de análisis
%
%       A ............... coeficientes lpc localizados
%       G ............... Ganancias del predictor localizados
%       S ............... módulo respuesta frecuencial asociada (dB)
%
%       Calcula los coeficientes lpc localizados utilizando una
%       ventana hamming de longitud N, desplazandola M muestras y
%       utilizando un orden P.

%function s=sinlpc(e,A,G,N)
%
%       e ....... señal de excitacion del filtro
%       A ....... matriz con la evolución de los coeficientes del filtro LPC A(1:orden,1:num_tramas)
%       G ....... Ganancia del predictor
%       N ....... numero de muestras a procesar por cada filtro
%
%       s ....... señal sintetizada

%function e=erlpc(samples,A,preenf,N)
%
%       samples ..... muestras de la señal
%       A ........... matriz con la evolución temporal de los coeficientes LPC
%       preenf ...... coeficiente de preénfasis
%       N ........... Desplazamiento
%
%       e ........... error de predicción
%
```