

PRÁCTICA 04: Detección del inicio-fin de una palabra.

Se va a implementar un algoritmo para detectar el inicio y fin de una palabra basado en el propuesto por Rabiner-Sambur. Este método considera las características de los sonidos:

- Sonidos sonoros: Tienen un alto contenido en energía y ocupan las frecuencias bajas del espectro de la voz.
- Sonidos no sonoros: Con un bajo contenido en energía y ocupan las frecuencias superiores del espectro de la voz.

Para la detección del inicio de la palabra realizamos la siguiente secuencia de pasos:

1. Por cada trama, calcular las funciones cruce por ceros y la magnitud: $Z(n)$ y $M(n)$.
2. Para obtener las estadísticas del ruido ambiental, se considera que las diez primeras ventanas son ruido:

$$M_{S_n} = \{M(1), M(2), \dots, M(10)\}$$
$$Z_{S_n} = \{Z(1), Z(2), \dots, Z(10)\}$$

3. Calcular la media (μ) y desviación estándar (σ) para las características del ruido y obtener los siguientes umbrales:
 - Umbral superior de energía (UmbSupEnrg): $0.5 * \max \{M(i)\}$
 - Umbral inferior de energía (UmbInfEnrg): $\mu_{M_s} + 2 \sigma_{M_s}$
 - Umbral de cruces por cero (UmbCruCero): $\mu_{Z_s} + 2 \sigma_{Z_s}$
4. Recorrer $M(i)$ hacia la derecha desde $i = 11$ hasta que $M(i) > UmbSupEnrg$. En este punto se garantiza la presencia de señal y a la posición que ocupa la denominamos ln .
5. Como el inicio de la señal debe de encontrarse en algún punto anterior a ln , recorrer $M(i)$ hacia la izquierda desde $i = ln$ hasta que $M(i) < UmbInfEnrg$. A esta posición la identificaremos como le y se reconoce como el inicio de la señal determinado por la magnitud.
6. Como existe la posibilidad de que sonido sordo preceda a uno sonoro, decrementar i desde $i = le$ hasta $i = le - 25$ (o $i = 11$ si la diferencia supera ese límite) verificando si tiene lugar alguna de las siguientes condiciones:
 - Si $Z(i) < UmbCruCero$, no hemos encontrado delante ningún fragmento de la señal, por lo que el inicio es le .
 - Si encontramos que se verifica que $Z(i) > UmbCruCero$ menos de tres veces seguidas, sólo se trata de una espiga de ruido, por lo que el inicio es le .

- Si encontramos que se verifica que $Z(i) > UmbCruCero$ al menos tres veces seguidas, se ha encontrado un fragmento sordo. Localizamos el i correspondiente a la primera de las esas veces seguidas y será el inicio de la palabra.

Para la detección del fin de palabra, hacer la misma secuencia de pasos (a partir del punto 4), pero en sentido inverso.

Implementar una función en MATLAB para obtener la señal recortada con la siguiente sintaxis:

```
tramasPalabra = inicioFin (tramas, numTramasRuido, ventana)
```

Aplicar la función anterior a los siguientes fragmentos de la señal:

<i>Pronunciación (utterance)</i>	<i>Desde la muestra</i>	<i>Hasta la muestra</i>
UNO	1	18000
DOS	18000	33000
TRES	33000	47000
CUATRO	47000	64000
CINCO	64000	80000
SEIS	80000	96000
SIETE	96000	113000
OCHO	113000	130000
NUEVE	130000	146800
DIEZ	146800	166000

Comprobar si se ha segmentado la palabra. Realice el proceso inverso al que ha tenido lugar en la función `segmentacion` (ejercicio 3).

```
palabra = invSegmentacion (tramasPalabra,despl)
```

DOCUMENTACIÓN A ENTREGAR

Fichero comprimido con el código implementado.

SCRIPT PRINCIPAL.M

```
%% PRINCIPAL.m
close all
clear all

%% Parámetros
tiempoTrama = 0.03;
tiempoDesplTrama = 0.01;
a = 0.95;
ventana = 'rectangular';
```

```

numTramasRuido = 10;

fragmentos = [      1   18000
                18000   33000
                33000   47000
                47000   64000
                64000   80000
                80000   96000
                96000  113000
                113000  130000
                130000  146800
                146800  166000];

%% Lectura de la señal de audio
[y,Fs] = audioread('numeros.wav');
y = y(:,1); %Nos quedamos con el primer canal

%% Preénfasis
y = preenfasis(y,a);

%% Segmentación
longTrama = round(Fs * tiempoTrama);
longDespTrama = round (Fs * tiempoDesplTrama);

%% Extracción de características en el dominio del tiempo
for i=1:size(fragmentos,1),

    conjTramas = segmentacion (y(fragmentos(i,1):fragmentos(i,2)), ...
                                longTrama, longDespTrama);

    tramasPalabra = inicioFin (conjTramas, numTramasRuido, ventana);

    palabra = invSegmentacion (tramasPalabra, longDespTrama);

    figure, subplot(1,2,1),plot(y(fragmentos(i,1):fragmentos(i,2)));
    subplot(1,2,2),plot(palabra);
    obj_senal = audioplayer(palabra,8000);
    play(obj_senal);
    pause (2);
end

```

RESULTADOS

*Nota: Se ha utilizado como umbral superior de energía $0.3 * \max \{M(i)\}$. ¿Qué ocurre si utilizamos el umbral indicado en el algoritmo anterior? ¿Y si lo subimos? ¿Y si variamos los otros umbrales?*

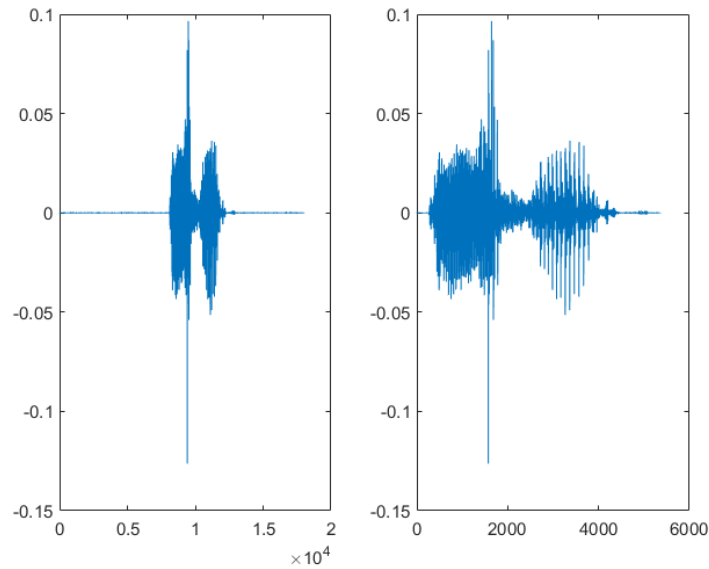


Ilustración 1: Resultado de la iteración 1 (UNO).

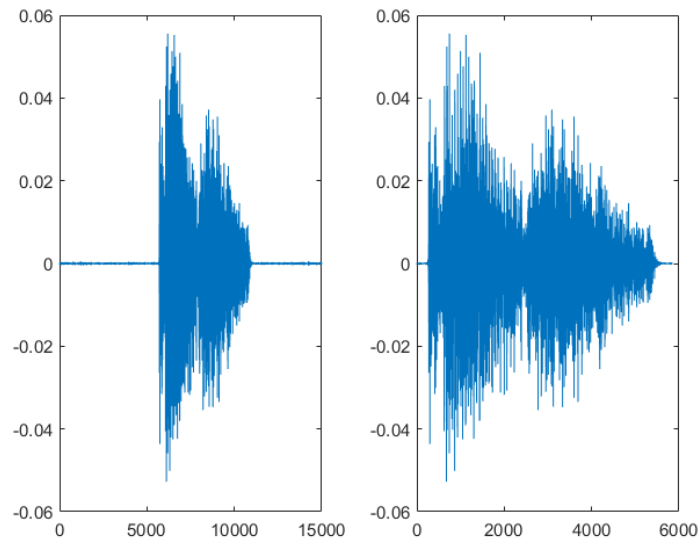


Ilustración 2: Resultado de la iteración 2 (DOS).

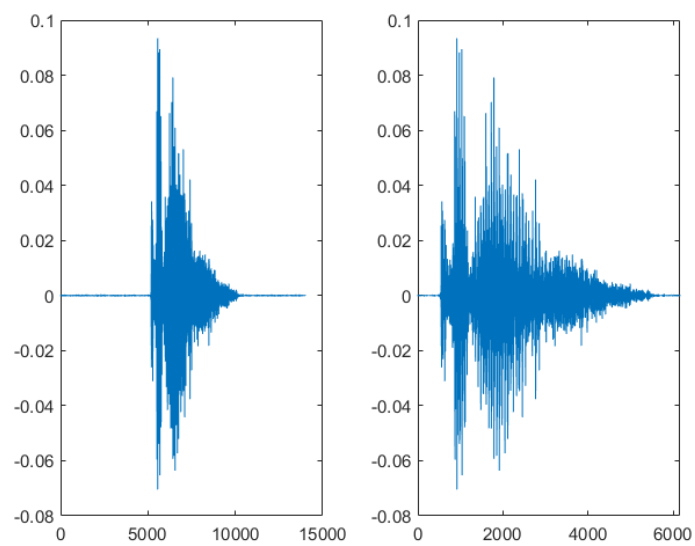


Ilustración 3: Resultado de la iteración 3 (TRES).

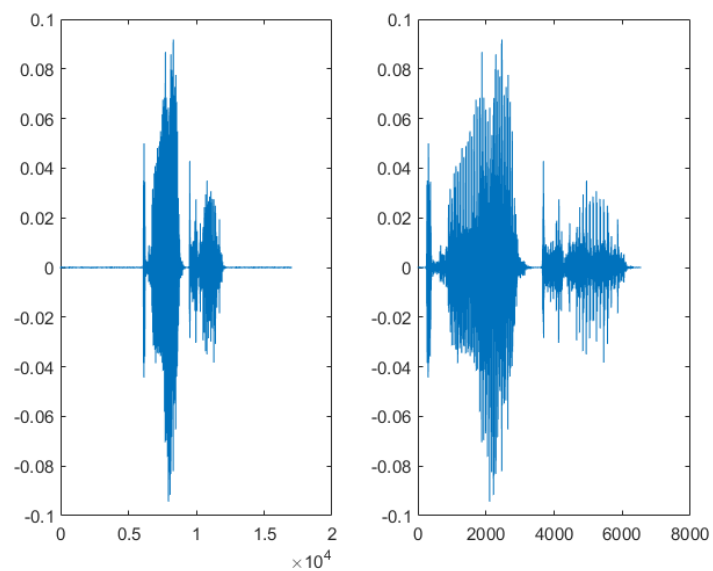


Ilustración 4: Resultado de la iteración 4 (CUATRO).

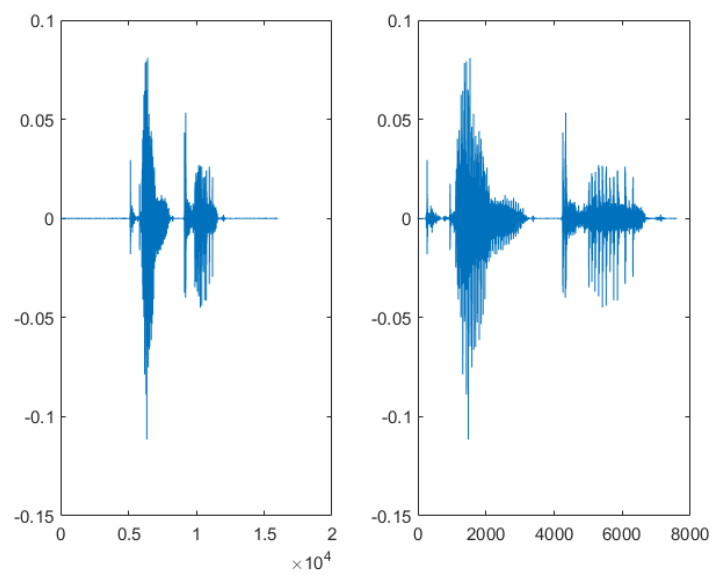


Ilustración 5: Resultado de la iteración 5 (CINCO).

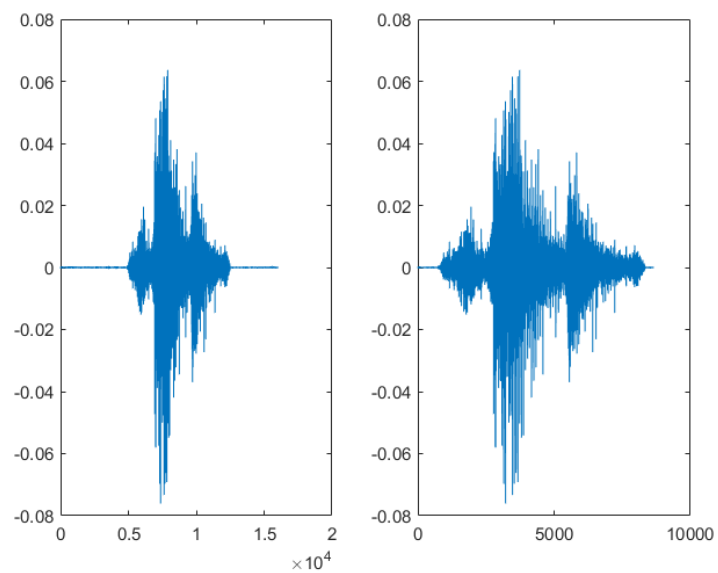


Ilustración 6: Resultado de la iteración 6 (SEIS).

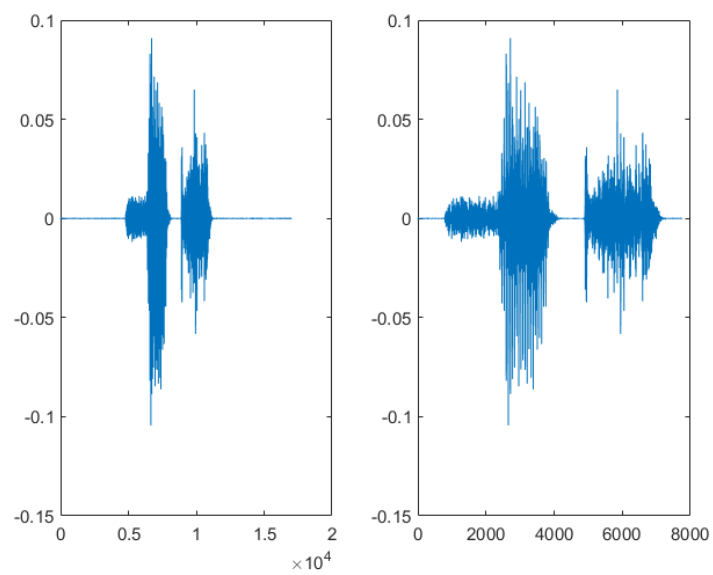


Ilustración 7: Resultado de la iteración 7 (SIETE).

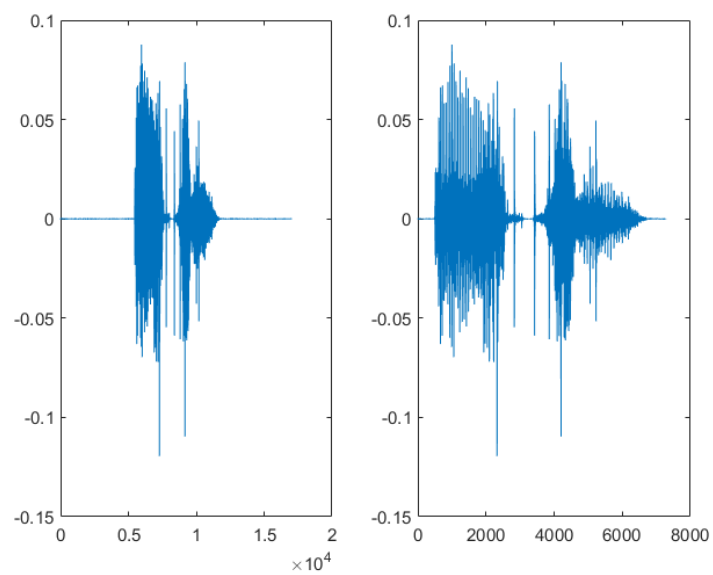


Ilustración 8: Resultado de la iteración 8 (OCHO).

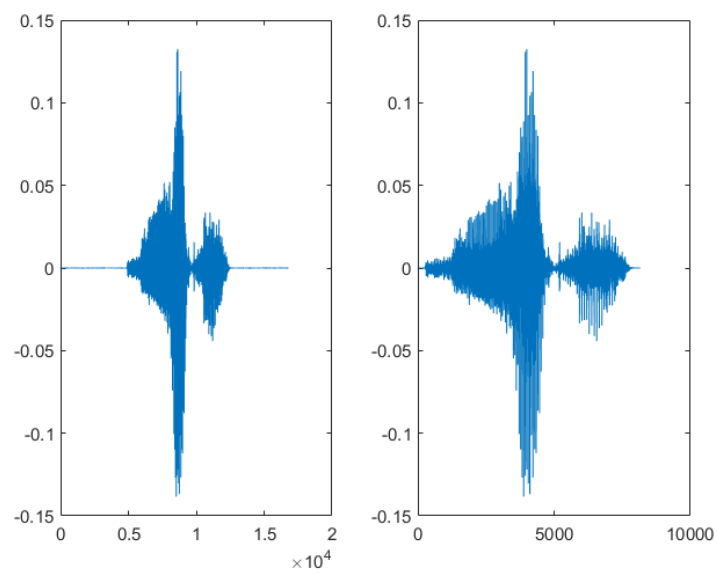


Ilustración 9: Resultado de la iteración 9 (NUEVE).

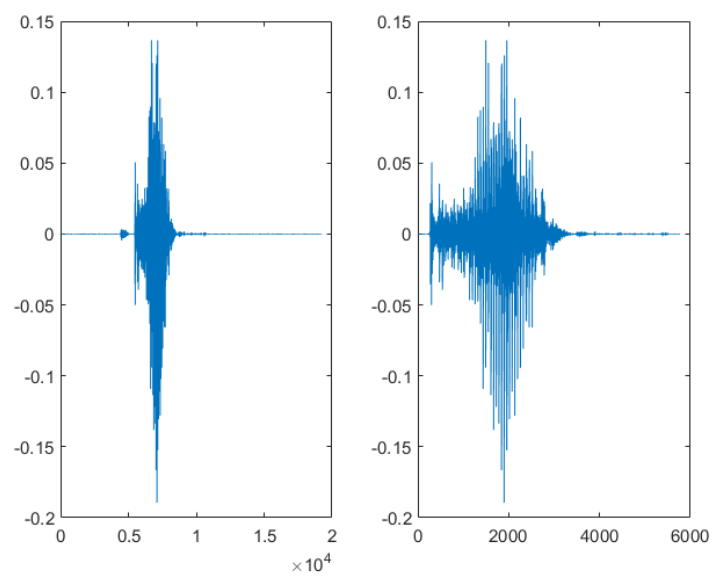


Ilustración 10: Resultado de la iteración 10 (DIEZ).