Práctica 2 - TDS

Segmentar señales en tiempo

Ver señales en frecuencia

Práctica 2

OBJETIVOS

1. Segmentar señales en tiempo: Veremos cómo establecer los índices para recortar un trozo de señal. Este proceso se denomina segmentación.
2. Ver el espectro de un fragmento de una señal. Ver el espectro por tramas -> Espectro tiempo-frecuencia (T-F).
3. Para casa: Empezar a crear los ficheros wav necesarios para la base de datos.

Importante:

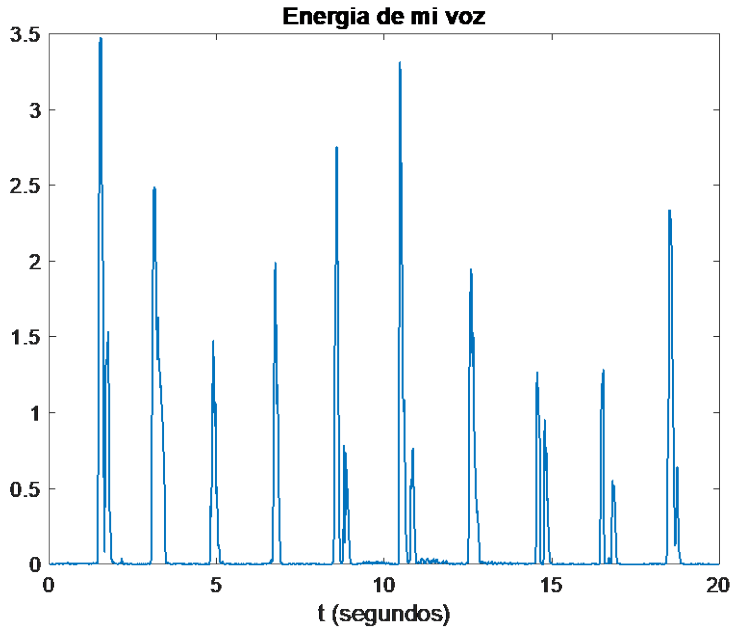
- Se proporciona el fichero mi_practica2.mlx donde escribir el código de Matlab utilizado para cada ejercicio.
- Este **fichero se debe entregar antes de la siguiente sesión junto con su impresión digital en pdf** (se puede hacer desde Matlab). Se valorará que el fichero tenga comentarios explicativos.
- En las siguientes transparencias, el texto de color marrón indica que son variables o instrucciones para usar en Matlab

Ejercicio 2.1

- Carga tu fichero wav donde has grabado los números del ‘cero’ al ‘nueve’ asignándole la variable mi_voz:

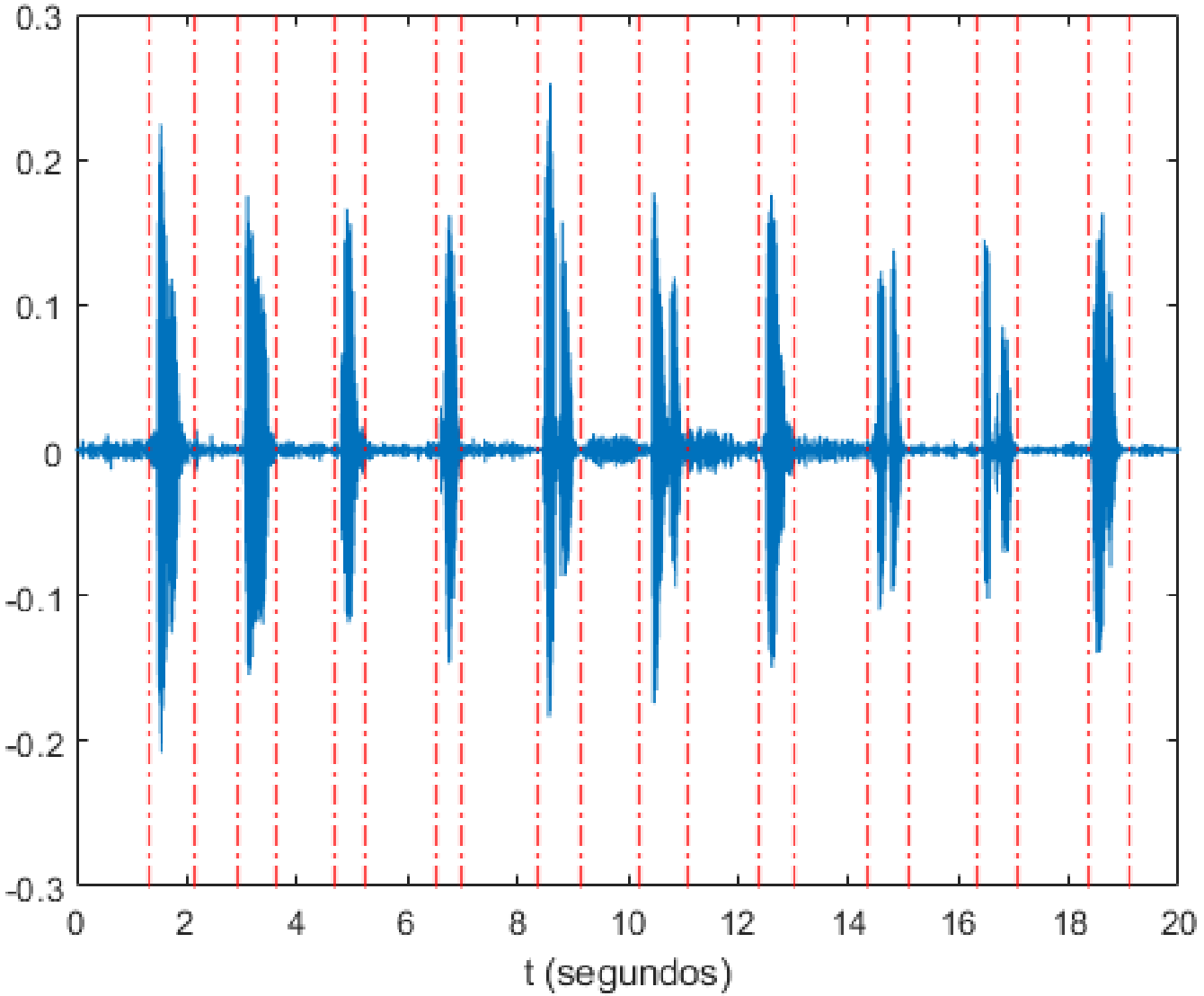
```
>> [mi_voz,fs] = audioread('?.wav');        % Cambia ? por el nombre de tu fichero
```

- Representa la señal completa **en tiempo discreto y en tiempo continuo** en dos figuras distintas.
- Calcula la Energía por tramas de 20ms usando la función del Ejercicio 1.3 ó 1.4 de la Práctica 1 y representa la Energía en tiempo continuo (mira el ejemplo de la figura)



Ejercicio 2.2

- Con la ayuda de la **figura de la energía**:
  - Genera un vector de tiempos T_limites con el tiempo de inicio y de fin de cada dígito en segundos. Ayúdate de la función ginput de Matlab. Selecciona los tiempos con cuidado para que incluya todo el contenido significativo. Aunque haya unas muestras de silencio antes o después no pasa nada.
  - El vector T_ limites debe tener 20 valores correspondientes a los tiempos de inicio y fin de cada dígito hablado **en segundos**.
  - Para comprobar que los valores de T_ limites son correctos, vuelve a dibujar mi_voz con el eje de segundos y usa el comando xline de Matlab para dibujar los valores de T_ limites como líneas verticales. Mira el ejemplo de la figura siguiente.
  - Si no has tomado los valores de forma secuencial, puedes ordenar los elementos de T_ limites de menor a mayor con la función sort de Matlab.
- A partir del vector T_ limites, genera el vector de índices discretos N_limites que serán los que marcan las posiciones de inicio y fin de cada dígito en el vector mi_voz.
  - Para comprobar que los valores de N_ limites son correctos, vuelve a dibujar mi_voz en el eje de tiempos discreto y usa el comando xline de Matlab para dibujar los valores de N_ limites como líneas verticales.


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10
12
14
16
18
20
t (segundos)

Ejercicio 2.3. Segmentar la señal de voz

- Usa N_limites para recortar cada uno de los dígitos del vector mi_voz. Una vez recortado el dígito:
  - Añádele 0.5 segundos de silencio antes y 0.5 segundos de silencio después. Para ello, calcula primero cuantas muestras equivalen a 0.5 segundos para añadirle un vector de 0's de ese tamaño antes y después.
- Almacena cada dígito en una variable distinta dándole como nombre: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve. Estos nombres serán las "etiquetas" del clasificador.
- Salva cada uno de los dígitos en un fichero wav mediante la función audiowrite asignándole como nombre su etiqueta: cero.wav, uno.wav, etc.
- De esta manera hemos empezado a crear la base de datos de dígitos hablados para poder realizar el clasificador.
- Cada uno de nosotros tendremos que conseguir 20 grabaciones para poder tener una base de datos de 2.400 locuciones de cada dígito en español. Para que haya suficiente diversidad, tendremos que grabar a 4-5 personas distintas (incluidos nosotros). Intenta grabar a un número similar de hombres y mujeres.
- Al final de la práctica te diremos como debes nombrar y guardar los ficheros.

Ejercicio 2.4. Visualizar el espectro

- En este ejercicio vamos a visualizar el espectro de algunas tramas de la señal proporcionada, tres3.wav. Para ello, cárgala asignando a su variable el nombre tres3.
- Vamos a seleccionar 2 tramas de distintos fonemas para visualizar el espectro:
  - Almacena en yframe1 la trama de 32ms cuya muestra de inicio está en 0.66 segundos. Corresponde al fonema /e/
  - Almacena en yframe2 la trama de 32ms cuya muestra de inicio está en 0.82 segundos. Corresponde al fonema /s/
- Visualiza en figuras distintas la forma de onda de yframe1 e yframe2 en tiempo, así como sus espectros usando las siguientes instrucciones, siendo L_frame el tamaño de la trama (hay que calcularlo antes):

```
>> Yframe=abs(fft(yframe));
>> ejef=(0:L_frame-1)/L_frame*fs;  % solo hace falta ejecutarlo la primera vez.
>> figure, plot(ejef(1:end/2), 20*log10(Yframe(1:end/2)),'r') % Dibujamos en dB
>> xlabel('Frecuencia (Hz)'), ylabel('Magnitud (dB)')
```

- Identifica los 6 primeros picos en cada espectro. Si están separados el mismo número de Hz (aproximadamente), entonces podemos decir que la señal tiene armónicos, y por tanto es periódica. Para un espectro con armónicos, la separación entre picos es el valor de *pitch* (F0)

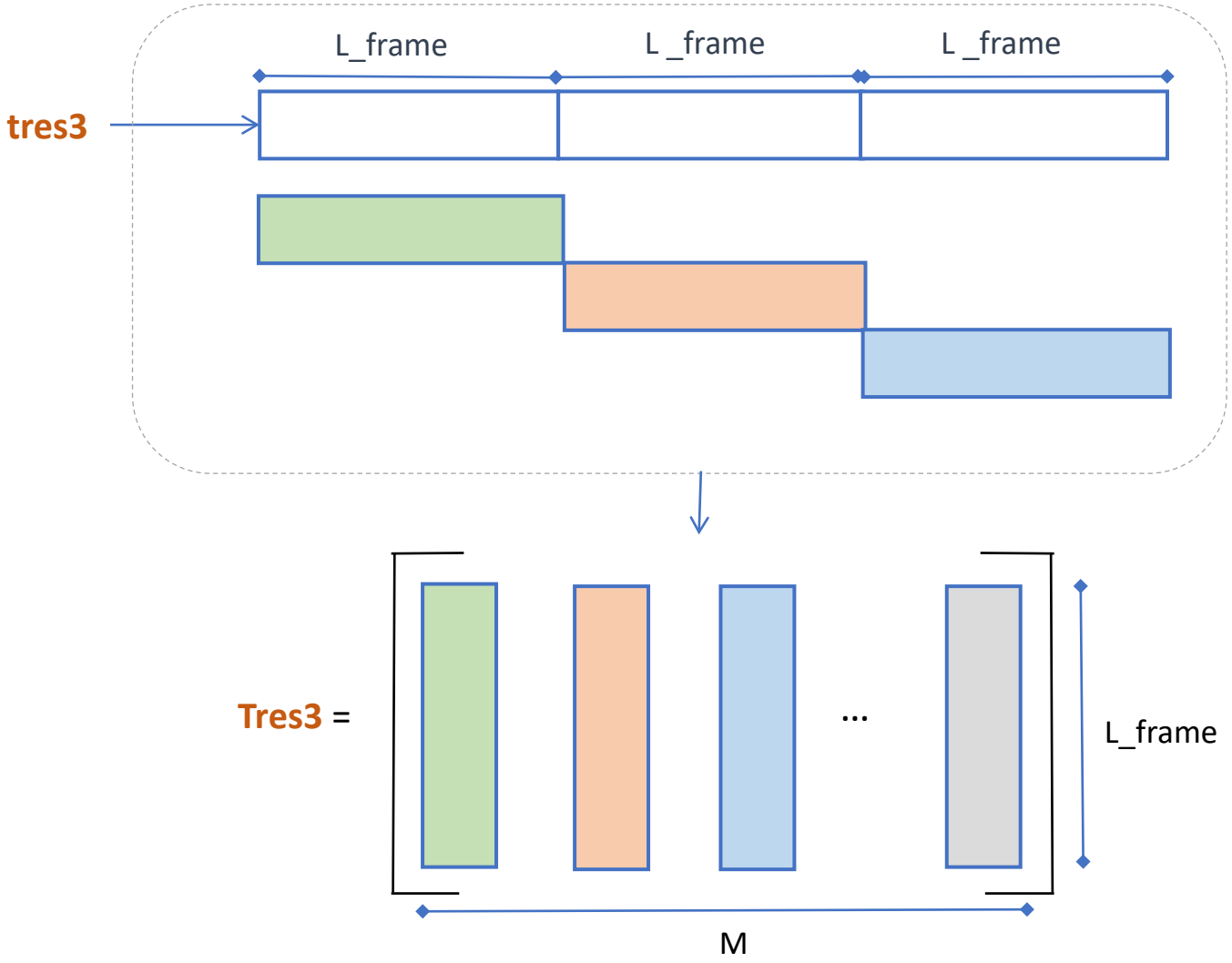
Ejercicio 2.5. Visualizar el espectro T-F (I)

- Hemos visto que el espectro cambia mucho si elegimos una trama u otra de la señal de voz. Por tanto, al visualizar el espectro conviene ver todas las tramas.
- Lo primero que vamos a hacer es eliminar los tiempos de silencio (0.5s al inicio y 0.5 al final).
- A continuación troceamos la señal tres3 en tramas de 32ms como en el ejercicio 1.3 de la Práctica 1 usando la función tramas_sin_solape
- Llama a la salida de la función Tres3 . La salida será una matriz de L_frame filas y M columnas, tal y como muestra la figura.
- A continuación ejecutamos la siguiente instrucción para obtener el espectro de cada columna (trama) de Tres3. Lo almacenamos en S_Tres3:

  >> S_tres3=abs(fft(Tres3));

- Creamos un eje apropiado para el tiempo ya que el espectro se calcula cada trama de 32ms:

  >> ejet_Trama= (0:M-1)*0.032;



Ejercicio 2.5. Visualizar el espectro T-F (II)

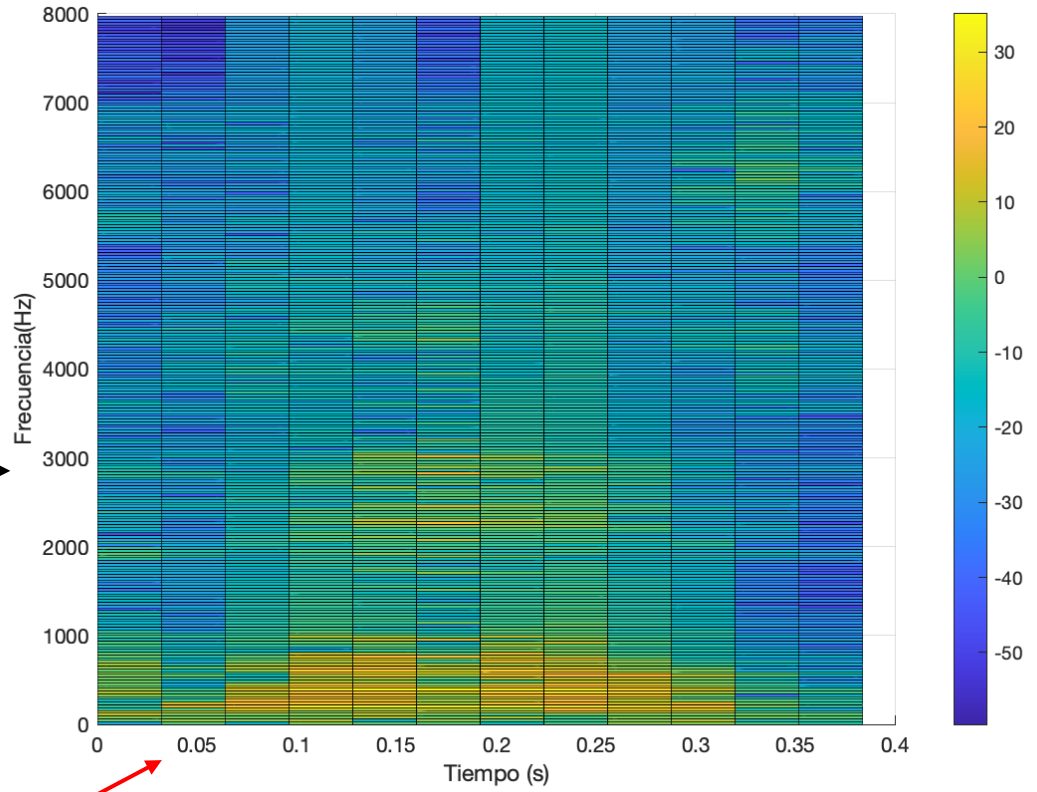
- Visualiza el espectro tiempo-frecuencia (T-F) almacenado en S_Tres3 usando:
  - La función waterfall:

  >> waterfall(ejet_Trama,ejef(1:end/2),20*log10(S_tres3(1:end/2,:)))

  >> xlabel('Tiempo (s)'), ylabel('Frecuencia(Hz)'), zlabel('Magnitud (dB)')
  - la función surf:

  >> surf(ejet_Trama,ejef(1:end/2),20*log10(S_tres3(1:end/2,:)))

  >> xlabel('Tiempo (s)'), ylabel('Frecuencia(Hz)')
- En la figura donde has usado surf, mueve el ángulo de visión hasta que la veas representada como en la figura de la derecha
- ¿Puedes identificar las tramas con el fonema /e/? Recuerda que en su espectro hallado en el ejercicio 2.4 aparecían varios picos muy altos al principio del eje de frecuencias.
- ¿Puedes identificar las tramas con el fonema /s/? Recuerda que su espectro hallado en el ejercicio 2.4 tenía bastante contenido en altas frecuencias



¡El clasificador de dígitos se basará en este tipo de imágenes para extraer las características con las que entrenaremos la red neuronal!

Para la próxima práctica

- Intenta realizar las grabaciones de los dígitos a alguna persona de tu entorno. Recuerda:
  - Procura hacer las grabaciones en un **entorno lo más silencioso posible**.
  - El locutor debe dejar una pausa de al menos 1 segundo entre número y número para poder segmentarlos bien.
  - Comprueba que el rango de valores de las muestras grabadas está comprendido en el rango (-1,1) y que no ha llegado a dichos valores en ningún momento. Si no es así, aleja al locutor del micrófono y vuelve a grabar.
  - Por el contrario, si ves que los valores de las muestras no pasan de 0.1-0.2 en el eje de ordenadas, acerca el locutor al micrófono y vuelve a grabar hasta que alcancen al menos 0.5 (ó -0.5) en su máxima amplitud.
- Debes realizar **20 grabaciones (a 4-5 personas distintas, no grabes más de 5 veces a la misma persona)** de cada dígito. No grabes a compañeros de clase para asegurar que no repetimos personas.
- Guárdalas en ficheros distintos de la siguiente forma: Asigna un número a cada persona y otro número a cada una de sus grabaciones. Utiliza el siguiente formato:

  etiqueta_#locutor_#grabacion.wav

- Por ejemplo, el fichero **cero_3_2.wav** correspondería al dígito ‘cero’ que ha dicho el locutor número 3 en la 2ª grabación de dicho locutor.
- En la Práctica 3 indicaremos las instrucciones para subir todos los ficheros wav a un repositorio común.
- **Fecha tope para tener las 20 grabaciones: Lunes 14 de noviembre**