Intercambio de tracto y pulso

Aplicaciones de la Biometría de la Voz

García Torrecilla, Juan Carlos – 110369

Contenido

Introducción

En este proyecto se plantea la realización de una aplicación que separe el tracto y pulso de señales de voz por filtrado inverso, así como la mezcla de tracto y pulso de nuevo, entre dos hablantes diferentes.

La realización de este proyecto se llevará a cabo en MATLAB.

A continuación, en la sección Background se explicará en qué se basa el método usado (LPC) y cómo funciona. Posteriormente, la sección Desarrollo e implementación describirá el desarrollo del programa, el código y la interfaz gráfica. Y, finalmente, se muestran los resultados y las conclusiones a las que se ha llegado en el apartado Resultados y conclusiones.

Background

Sistema de producción de la voz

Para realizar la separación e intercambio del tracto y pulso de una señal, en primer lugar, es necesario conocer el sistema de producción de la voz.

La voz se produce en el tracto vocal. El proceso se inicia con la expulsión de aire de los pulmones. Este aire atraviesa la glotis donde, en el caso de sonidos sonoros, las cuerdas vocales se abren y cierran haciendo vibrar el aire generando una señal vibratoria. En el caso de sonidos sordos, las cuerdas vocales se mantienen abiertas dejando pasar el aire; lo que produce una señal similar al denominado ruido blanco [1]. A esta señal vibratoria se le denomina señal de excitación, y va a estar caracterizada por la frecuencia fundamental y sus armónicos.

Posteriormente, la señal de excitación pasa por el tracto vocal, compuesto principalmente por la cavidad bucal y la cavidad nasal [1]. El tracto va a actuar como caja de resonancia, favoreciendo unas frecuencias determinadas (formantes), dando a cada voz unas características propias que nos permiten diferenciar individuos.

Encontraremos una diferencia especialmente notable entre pulsos de hombre y mujer, donde la frecuencia fundamental del hombre será menor, principalmente a causa del tamaño generalmente mayor de la cuerda vocal, debido al desarrollo de la nuez. De igual manera, el tracto vocal de hombre será generalmente más grande que el de mujer, por lo que favorecerá tonos más graves.

Con todo esto, la idea es modelar el tracto como un filtro, que recibe el pulso en su entrada, y genera la señal de voz a la salida. Para ello, se va a emplear el modelo de predicción lineal LPC.

Predicción lineal de la voz

La predicción lineal aplicada a la voz se basa en la idea de que la voz se puede modelar como una combinación lineal de valores previos observados (modelo de predicción lineal), teniendo en cuenta que la señal de voz puede ser quasiperiódica y que, físicamente, las modificaciones del tracto vocal no pueden ser muy rápidas.

Esto nos va a permitir elaborar un modelo de predicción mediante la obtención de un conjunto de parámetros de la siguiente manera.

Se parte de la idea ya comentada de que el tracto vocal actúa como un filtro, de forma que la voz se modelaría como un sistema lineal según la convolución:

$$s(t) = (v * u)(t) = \int_{-\infty}^{\infty} v(\tau) \cdot u(t-\tau)\, \partial\tau \qquad (1)$$

Donde:

- $u(t)$ es la señal de excitación de entrada (el pulso)
- $v(t)$ es la función de respuesta al impulso del sistema (el tracto)
- $s(t)$ es la señal de salida (la voz)

Dado que vamos a trabajar con señales de voz muestreadas, es decir discretas, $s(t)$ pasará denominarse $s(n)$. Se asume que el sistema es causal, ya que la voz producida no dependerá de las muestras futuras, solo de las pasadas. Y, además, se asume que no todas las muestras pasadas influirán en la salida actual, sino solo un conjunto reciente de muestras, por lo que se tendrá en cuenta sólo las $L + 1$ muestras previas. De forma que el valor de salida actual sea:

$$s_n = \sum_{l=0}^{L} v(l) \cdot u(n-l) = v_0 \cdot u_n + v_1 \cdot u_{n-1} + \cdots + v_L \cdot u_{n-L} \quad (2)$$

Desconocemos la señal de entrada $u(n)$, pero no la señal de salida $s(n)$, por lo que se transforma la expresión anterior para expresarlo en función de la salida finalmente en (8) mediante el siguiente proceso.

Asumiendo que el sistema es invariante en el tiempo, el valor previo de salida será:

$$s_{n-1} = v_0 \cdot u_{n-1} + v_1 \cdot u_{n-2} + \cdots + v_L \cdot u_{n-L-1} \quad (3)$$

Si ahora despejamos en (3) el término u_{n-1}, que también aparece en (2):

$$v_0 \cdot u_{n-1} = s_{n-1} - v_1 \cdot u_{n-2} - \cdots - v_L \cdot u_{n-L-1} \quad (4)$$

Dividimos por v_0 y multiplicamos por v_1:

$$v_1 \cdot u_{n-1} = \frac{v_1}{v_0} \cdot s_{n-1} - \frac{{v_1}^2}{v_0} \cdot u_{n-2} - \cdots - \frac{v_1}{v_0} \cdot v_L \cdot u_{n-L-1} \quad (5)$$

Y podemos sustituir $v_1 \cdot u_{n-1}$ de (5) en (2). De forma que:

$$s_n = v_0 \cdot u_n + \left(\frac{v_1}{v_0} \cdot s_{n-1} - \frac{{v_1}^2}{v_0} \cdot u_{n-2} - \cdots - \frac{v_1}{v_0} \cdot v_L \cdot u_{n-L-1}\right) + \cdots + v_L \cdot u_{n-L} \quad (6)$$

Se agrupan términos:

$$s_n = v_0 \cdot u_n + \frac{v_1}{v_0} \cdot s_{n-1} + \left(v_2 - \frac{{v_1}^2}{v_0}\right) \cdot u_{n-2} + \cdots + \left(v_L - \frac{v_1 \cdot v_{L-1}}{v_0}\right) \cdot u_{n-L} - \frac{v_1 \cdot v_L}{v_0} \cdot u_{n-L-1} \quad (7)$$

De esta manera introducimos s_{n-1} en la expresión de s_n. Siguiendo esta metodología, para el resto de términos, obtenemos s_n como combinación lineal de términos anteriores de $s(n)$ multiplicados por un valor a, y el término actual de entrada $v_0 \cdot u_n$. Igual que antes, podemos asumir que los valores muy antiguos de salida no tendrán efecto en la salida actual, por lo que se limita a K valores.

$$s_n = v_0 \cdot u_n + a_1 \cdot s_{n-1} + a_2 \cdot s_{n-2} + \cdots + a_K \cdot s_{n-K} \quad (8)$$

Se despeja $v_0 \cdot u_n$ en (8)

$$v_0 \cdot u_n = a_0 \cdot s_n - a_1 \cdot s_{n-1} - a_2 \cdot s_{n-2} - \cdots - a_K \cdot s_{n-K} \quad (9)$$

Siendo $a_0 = 1$. Lo que tenemos en (9) es ahora la convolución de los vectores $a(n)$ y $s(n)$ de la forma.

$$v_0 \cdot u_n = a(n) * s(n) \quad (10)$$

Para algunas aplicaciones, podemos prescindir del valor de escala v_0, que toma siempre el mismo valor para todo n. Quedándonos finalmente con

$$u_n = a(n) * s(n) \quad (11)$$

lo que nos indica que podemos obtener por convolución el valor de la señal de entrada u_n a partir de la señal de voz que tenemos $s(n)$, si conocemos los parámetros del vector $a(n)$.

Además, a partir de la convolución podemos obtener el filtro inverso que corresponda al tracto vocal, así como analizar formantes. Otro uso que permite el vector $a(n)$ es la codificación de señales mediante la función de estimación (14) acompañada con un vector de errores de estimación (16) de la señal original.

Estimador LPC y síntesis de voz

LPC (*Linear Predictive Coding*) es un mecanismo matemático que permite estimar los próximos valores de una señal discreta como una función lineal, basándose en los valores de muestras previas [2, 3].

Partiendo de la fórmula (9), si despejamos $a_0 \cdot s(n)$ (recordando que $a_0 = 1$), tenemos:

$$a_0 \cdot s_n = v_0 \cdot u_n + a_1 \cdot s_{n-1} + a_2 \cdot s_{n-2} + \cdots + a_K \cdot s_{n-K} \quad (12)$$

$$s(n) = v_0 \cdot u_n + \sum_{k=1}^{K} a_k \cdot s(n-k) \quad (13)$$

Dado que $u(n)$ va a ser un tren de impulsos, tomará valor 0 para la mayoría de los valores, por lo que podemos denominar el sumatorio como como un estimador de $s(n)$ [4] en función de las muestras precedentes:

$$\hat{s}(n) = \sum_{k=1}^{K} a_k \cdot s(n-k) \quad (14)$$

$$s(n) = v_0 \cdot u_n + \hat{s}(n) \quad (15)$$

Despejando $v_0 \cdot u_n$ se calcula la diferencia $s(n) - \hat{s}(n)$, que no es más que el error del estimador $e(n)$ respecto del valor real, también llamado residuo.

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{K} a_k \cdot s(n-k) \quad (16)$$

Si aplicamos la Transformada Z en (16), tenemos el siguiente desarrollo:

$$TZ[e(n)] = TZ[s(n) - \hat{s}(n)] = TZ\left[s(n) - \sum_{k=1}^{K} a_k \cdot s(n-k)\right] \quad (17)$$

$$E(z) = S(z) - \hat{S}(z) = S(z) - \sum_{k=1}^{K} a_k \cdot S(z) \cdot z^{-k} \quad (18)$$

Sacando factor común $S(z)$:

$$E(z) = S(z) \cdot \left(1 - \sum_{k=1}^{K} a_k \cdot z^{-k}\right) \quad (19)$$

Si llamamos “filtro” al sumatorio de (19):

$$F(z) = \sum_{k=1}^{K} a_k \cdot z^{-k} \qquad (20)$$

Tendremos que el término $\hat{S}(z)$ (que aparece en (18)) corresponde a:

$$\hat{S}(z) = F(z) \cdot S(z) \qquad (21)$$

Sustituyendo este último término en la fórmula del residuo (18):

$$E(z) = S(z) - \hat{S}(z) = S(z) - \big(F(z) \cdot S(z)\big) = S(z) \cdot \big(1 - F(z)\big) \qquad (22)$$

Siendo la entrada de un filtro la señal $S(z)$ y la salida el error $E(z)$, la función de transferencia del filtro se puede definir como [2]:

$$H(z) = \frac{E(z)}{S(z)} = \frac{S(z) \cdot \big(1 - F(z)\big)}{S(z)} = 1 - F(z) = 1 - \sum_{k=1}^{K} a_k \cdot z^{-k} \qquad (23)$$

El sistema es un filtro inverso, que se corresponde con el siguiente diagrama:

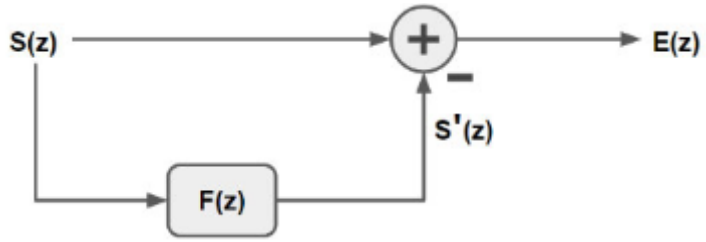

FIGURA 1 - DIAGRAMA DESARROLLADO DEL FILTRO INVERSO [2]

Este filtro inverso, al ser aplicado a la señal, devuelve el residuo de dicha señal. Aparece simplificado en la siguiente figura:

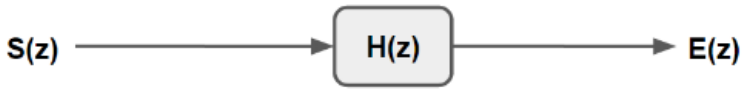

FIGURA 2 – DIAGRAMA SIMPLE DEL FILTRO INVERSO [2]

Si se parte de la señal de excitación, que es equivalente a la señal residual $E(z)$, aplicando el inverso del filtro inverso $H(z)$, se puede obtener la señal original $S(z)$. Este proceso, descrito en la siguiente figura, corresponde a la síntesis de voz:

FIGURA 3 - FILTRO INVERSO DE H(Z) [2]

Recapitulando, conociendo el vector $a(n)$ podemos elaborar un estimador $\hat{s}(n)$ que nos permite modelar el tracto vocal como un filtro. Aplicando Transformada Z en las ecuaciones, convertimos convolución en multiplicación, facilitando revertir la operación, lo que nos permite obtener el pulso glótico a partir de la señal de voz original.

De esta manera, podemos hacer la operación en ambos sentidos mediante el filtro que modela el tracto vocal y su inverso.

Todo esto nos va a permitir separar señales en pulso glótico y tracto vocal, y mezclarlas de nuevo, de forma que tengamos el pulso de una señal y el tracto de otra.

Cálculo del vector a(n)

Sobre (16) se calcula el error cuadrático medio [4]

$$E = \sum e^2(n) \tag{24}$$

Y con esto, se calculan los valores de $a(n)$ que minimicen ese error cuadrático medio. Para ello se calcula la derivada igualando a 0, de forma que:

$$\frac{\partial E}{\partial a_j} = \frac{\partial}{\partial a_j} \sum_n \left(s(n) + \sum_k a_k \cdot s(n-k) \right)^2 \tag{25}$$

$$0 = \sum_n 2 \cdot \left(s(n) + \sum_k a_k \cdot s(n-k) \right) s(n-j) \tag{26}$$

$$0 = \sum_n s(n) \cdot s(n-j) + \sum_k a_k \cdot \sum_n s(n-k) \cdot s(n-j) \tag{27}$$

Donde podemos identificar dos sumatorios que corresponden con autocorrelaciones de señal:

$$\sum_k a_k \cdot R(k-j) = -R(j) \tag{28}$$

Desarrollando (28), podemos escribir esto con notación matricial como un modelo autorregresivo (AR) que sigue la siguiente ecuación de Yule-Walker:

$$\begin{pmatrix} R_0 & R_1 & \cdots & R_{K-1} \\ R_1 & R_0 & \cdots & R_{K-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{K-1} & R_{K-2} & \cdots & R_0 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{pmatrix} = - \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_K \end{pmatrix} \tag{29}$$

Por lo que, dada una señal, podemos obtener el vector $a(n)$ calculando las correlaciones del sistema de ecuaciones y resolviéndolo.

Desarrollo e implementación

En base a lo descrito en la sección anterior Background, se va a desarrollar una aplicación en MATLAB que permita separar tracto y pulso de una señal de audio grabada en WAV, y recomponga una señal de audio a partir de un tracto y un pulso dado.

Funciones básicas

División tracto-pulso

Para la división del tracto y el pulso de una señal se ha implementado la función *separa.m*. Esta función recibe como parámetros:

- **`x`:** Vector de la señal de audio original.
- **`n_coef`:** El número de coeficientes del modelo LPC.
- **`polo`:** El polo a emplear en el filtro de preénfasis. En caso de no emplear filtro tomará el valor 0.
- **`l_v`:** Longitud de la ventana a emplear en el proceso.
- **`prop_despl`:** Proporción de la ventana a desplazar (p. ej.: ½ , ¼, …). Se multiplicará a "`l_v`" para obtener el desplazamiento de la ventana.

En primer lugar, se comprueba si la señal "x" tiene varios canales y, de ser así, se convierte en una señal mono (1 canal). Se calculan los datos necesarios para el proceso como el desplazamiento o el número de trozos a iterar, y se filtra la señal con el filtro de preénfasis.

El filtro de preénfasis enfatiza (aumenta en amplitud) las altas frecuencias de la señal, mientras que atenúa (aplana la amplitud de la onda) en las bajas frecuencias de la señal; como viene ilustrado en [5].

Esto permite prevenir el ruido generado en el proceso. El ruido tiene de forma inherente mayor amplitud en las altas frecuencias (y menor en las bajas) de forma que distorsiona el sonido en altas frecuencias. Para contrarrestarlo se aplica un filtro de preénfasis para incrementar las altas frecuencias de la voz y que, si se genera ruido durante el proceso, no distorsione la señal en las altas frecuencias. Posteriormente, es habitual aplicar un filtro de deénfasis para deshacer el proceso, de forma que la voz vuelva a su estado natural y el ruido quede atenuado en las altas frecuencias.

Seguidamente, se itera sobre la señal que es troceada en ventanas, calculándose para cada ventana el vector "a" y se guarda en una matriz "`A`" de dimensiones $(n_{coef} + 1,\ n_{trozos})$.[1] Este vector se calcula mediante la función de MATLAB "`lpc`", que calcula los coeficientes como se ha visto en Cálculo del vector a(n).

Con este vector "a" se realiza el filtrado inverso de la ventana, obteniendo el pulso glótico a partir del trozo de señal original. Parte de estos vectores de pulso se guardan en una matriz "`P`" de dimensión $(desplaza,\ n_{trozos})$, donde "`desplaza`" es el desplazamiento de la ventana.

La función devuelve las matrices de vectores "`A`" y "`P`", que contiene la información necesaria para reconstruir la señal. Adicionalmente, se añadió como valor de salida "`x_fil`", que contiene la señal original filtrada, que es necesaria para uso de otras funciones (gráficas principalmente).

[1] Recordemos que el primer coeficiente es 1, por lo que no se tiene en cuenta. Para 14 coeficientes, la función "`lpc`" devolverá un vector de 15 elementos.

Recomposición tracto-pulso

Para recomponer la señal, se ha implementado la función *junta.m*. Esta función recibe como parámetros:

- **A**: Matriz de dimensión $(n_{coef} + 1,\ n_{trozos})$ que contiene los vectores LPC.
- **P**: Matriz de dimensión $(desplaza,\ n_{trozos})$ que contiene los pulsos de los trozos.
- **`polo:`** El polo a emplear en el filtro de deénfasis (0 si no se filtra).

La función obtiene el número de coeficientes de las dimensiones de "A", y el número de trozos de las dimensiones de "A" y "P" (se coge el menor número, dado que pueden proceder de señales diferentes y tener diferente cantidad de trozos).

Posteriormente se itera filtrando cada vector del pulso con el vector de coeficientes "a", para deshacer el proceso realizado durante la función de separación, convirtiendo el pulso en señal de voz, que se guarda en un vector.

Finalmente, se filtra el vector de la señal recompuesta con el filtro de deénfasis, y se devuelve el vector como variable de salida.

Funciones de gráficos

Para la elaboración de gráficos, así como transformaciones de tracto y pulso para ser guardados en archivos WAV, se han elaborado un conjunto de funciones que se explican a continuación.

Gráficas de tracto

El archivo *graficas_tracto.m* implementa la función que elabora las gráficas relacionadas con el tracto vocal. Se generan espectrogramas del tracto, lo que nos va a permitir visualizar las frecuencias realzadas por el tracto vocal a lo largo del tiempo.

Esta función recibe los parámetros:

- **`index:`** Un índice sin relevancia en el proceso. Se emplea para indicar si los datos corresponden al archivo 1 ó 2 cuando se están mezclando señales, con el fin de no sobreescribir las gráficas.
- **`l_v:`** Nº de muestras de longitud de ventana.
- **`desplaza:`** Nº de muestras de desplazamiento de ventana.
- **`A:`** Matriz de vectores con coeficientes LPC (obtenida con *separa.m*).
- **`fs:`** Frecuencia de muestreo de la señal original.

Con los datos de entrada se calculan otros parámetros como el número de trozos o el solapamiento entre ventanas.

Posteriormente, se itera sobre el número de trozos y se obtiene el vector de coeficientes LPC. Para cada vector LPC, se obtiene la respuesta en frecuencia al filtro digital que forman los coeficientes del vector "a".

Por un lado, se obtiene la respuesta en frecuencia (función `freqz`) en todo el espectro[2] (`H_tracto`), sobre la que se calcula la transformada inversa de Fourier para pasarlo al dominio del tiempo. Con estos vectores parciales se crea un vector "`tracto`" que corresponde al tracto como tal de la señal, y que será devuelto como parámetro de salida para guardar. Sobre este vector se elabora un espectrograma plano (función `specgram`).

Por otro lado, también se obtiene la respuesta en frecuencia de la mitad del espectro (`H`), que se transforma a decibelios para luego elaborar un espectrograma 3d (función `surf`).

Gráficas de los formantes

El archivo *graficas_formantes.m* implementa la función que produce las gráficas relacionadas con el análisis de los formantes de la voz. Esto nos va a permitir identificar los formantes del tracto vocal en un momento temporal concreto de forma clara y visual.

La función recibe como parámetros:

- **`index:`** Índice para diferenciar si los datos corresponden al archivo de audio 1 ó 2.
- **`x:`** El vector de la señal de voz original (filtrada, si se ha empleado filtro de preénfasis), ya que los coeficientes LPC han sido sacados de la señal filtrada.
- **`trama:`** Trama cuyas formantes se va a representar.
- **`l_v:`** Nº de muestras de longitud de ventana.
- **`fs:`** Frecuencia de muestreo.
- **`A:`** Matriz de vectores con coeficientes LPC (obtenida con *separa.m*).

[2] En esta respuesta en frecuencia, una mitad del espectro es una copia en espejo de la otra mitad.

Como en la función anterior, se obtiene para cada vector de coeficientes la respuesta en frecuencia mediante la función `freqz` y, posteriormente, se pasa a decibelios. Con esto, se representa la trama LPC, así como su envolvente (mediante filtrado) y la envolvente de esta última.

Además, a partir de la señal original, se calcula la transformada de Fourier en tiempo discreto, mediante la función `specgram`, que se convierte igualmente a decibelios, y se representa en la gráfica.

Posteriormente, se elabora un espectrograma con la diferencia entre la trama LPC y la trama de Fourier.

Gráficas pulso

El archivo *graficas_pulso.m* implementa la función que produce las gráficas relacionadas con el análisis del pulso glotal. Genera ilustraciones que permitirán visualizar el pulso glotal así como la frecuencia de vibración de la cuerda vocal.

La función recibe como parámetros:

- **`index:`** Índice para diferenciar si los datos corresponden al archivo de audio 1 ó 2.
- **`pulso:`** Matriz de vectores de pulso (obtenida con *separa.m*).
- **`l_v:`** Nº de muestras de longitud de ventana.
- **`fs:`** Frecuencia de muestreo.

Se transforma la matriz de pulso de dimensión $(desplaza,\ n_{trozos})$ en un vector de dimensión $(1, desplaza \cdot\ n_{trozos})$. A continuacion, se genera la gráfica del pulso y un espectrograma del mismo.

Posteriormente, se integra el vector de pulso glótico para obtener la onda glótica, y se elaboran de nuevo gráfica y espectrograma.

Finalmente, se general una gráfica con el pulso y la onda.

La función devuelve el pulso y onda glótica normalizados para posibilitar guardarlo en archivo.

Se valoró hacer una función para generar espectrogramas de señal inicial y final. Pero, debido a su sencillez (una sola línea) se decidió no hacerlas y crearlo manualmente en la ejecución.

Ejecución por script e interfaz gráfica

Además de los archivos anteriormente mencionados, se incluye un script *test.m* a modo de ejecución de prueba y demostración de la funcionalidad.

En este script se indica un conjunto de parámetros de configuración al inicio del mismo, donde se establece el nombre de los archivos de entrada, tamaño de ventana, desplazamiento, polos de los filtros y número de coeficientes.

A continuación, el script lee, separa y recompone los archivos con la configuración indicada. Seguido de una sección (indicada en comentarios como "`[GRÁFICAS]`") donde se generan las gráficas indicadas en la sección Funciones de gráficos para cada uno de los archivos, y los espectrogramas para cada archivo de audio original y cada archivo de audio recompuesto.

Finalmente, se encuentran las líneas de reproducción de los audios recompuestos, así como las sentencias para guardar los archivos de tracto vocal, pulso glótico y onda glótica.

El script no está pensado para ejecutar el código al completo de una sola vez, debido a la gran cantidad de gráficas generadas y la reproducción de múltiples archivos de audio.

Adicionalmente, se ha elaborado una interfaz gráfica mediante la herramienta GUIDE de MATLAB, cuyo *layout* se encuentra en el archivo *gui.fig* y cuya ejecución se lanza ejecutando el archivo *gui.m*.

FIGURA 4 - INTERFAZ GRÁFICA

La interfaz gráfica cuenta con una funcionalidad de selección y lectura de archivos en la parte superior, pudiéndose abrir archivos en cualquier ubicación mediante el botón "*Browse*". Alternativamente, se pueden leer archivos seleccionando un archivo de la lista desplegable superior, la cual lista únicamente los archivos WAV del directorio "*./wav*".

Al cargar un archivo, se mostrará un texto con el nombre del archivo abierto y la frecuencia de muestreo del mismo, a modo de *feedback* para el usuario; y se mostrará un pequeño plot con el archivo leído, para previsualizar (sin mucho detalle) la apariencia gráfica del audio, su duración, y si cuenta con múltiples canales (color de la gráfica).

En las cajas de configuración se puede indicar si se desea usar filtros de preénfasis y deénfasis (así como sus polos correspondientes), el tamaño de ventana y el porcentaje de solapamiento, así como el número de coeficientes LPC a emplear. Automáticamente, al seleccionar dos audios con la misma frecuencia de muestreo, el programa establece como sugerencia el número de coeficientes a la milésima parte de la frecuencia de muestreo $n_{coef} = floor(0.001 \cdot fs)$.

El botón "*RUN*" se habilita cuando los dos archivos tienen la misma frecuencia de muestreo. Por esa razón, se indica la frecuencia en el texto de salida cuando se lee un archivo. Presionando el botón "*RUN*", se realiza la separación y recomposición de los audios.

Los audios resultantes se pueden escuchar presionando los botones que indican las combinaciones de tracto y pulso, y detenerlos con el botón "*Stop*".

Finalmente, los botones inferiores permiten generar las gráficas de tracto, pulso y formantes (en la trama seleccionada) para cada uno de los archivos.

Hay que tener en cuenta que no se ofrece, en cambio, funcionalidad para generar espectrogramas de los archivos originales y resultantes. Tampoco hay funcionalidad para guardar los archivos de tracto, onda y pulso.

Resultados y conclusiones

A pesar del proceso matemático que hay detrás de LPC, la implementación del intercambio de tracto y pulso es relativamente sencilla (empleando las funciones de MATLAB) y los resultados son bastante buenos bajo ciertas circunstancias y tras un ajuste adecuado de los parámetros.

Mezcla de diferentes frases

A lo largo del proyecto se han realizado diversas pruebas de mezcla, donde se juntaban frases diferentes de locutores diferentes. En estos casos, el resultado es habitualmente una mezcla de las dos frases. La frase resultante va a depender en gran medida del número de coeficientes que tengamos.

Cuanto menor sea el número de coeficientes LPC empleados, peor se modelizará el tracto vocal y, en consecuencia, menos información se les quitará a las señales originales (y menos se le añadirá). Por tanto, a menos coeficientes LPC, más se asemejará la frase resultante a la señal original a la que pertenecía el pulso. Siendo, en el caso extremo de 0 coeficientes LPC, la señal original igual que la señal resultante e igual al pulso (en ausencia de filtros).

Por otro lado, a mayor número de coeficientes LPC, mayor será la influencia del tracto vocal de una señal en el pulso de la otra. Sin embargo, al tratarse de frases distintas, los golpes de voz no concuerdan y, cuando se produce una pausa en la señal que proporciona el tracto, se puede distinguir claramente la frase original del pulso.

En general, los resultados son bastante buenos, teniendo en cuenta que las frases originales no coinciden en contenido. Por supuesto, los audios recompuestos van a contar con bastante ruido y distorsión donde los filtros de preénfasis y deénfasis pueden resultar de utilidad para tratar de mejorar la calidad resultante en la medida de lo posible.

Por concretar, y como ejemplo, se incluyen los archivos "*pulp.wav*" (que contiene una grabación de hombre con voz muy grave, extraída de la película Pulp Fiction) que llamaremos "audio 1", y "*sheldon.wav*" (que contiene una grabación de hombre con voz relativamente aguda, extraída de la serie The Big Bang Theory) que llamaremos "audio 2", con los que se han hecho algunas pruebas.

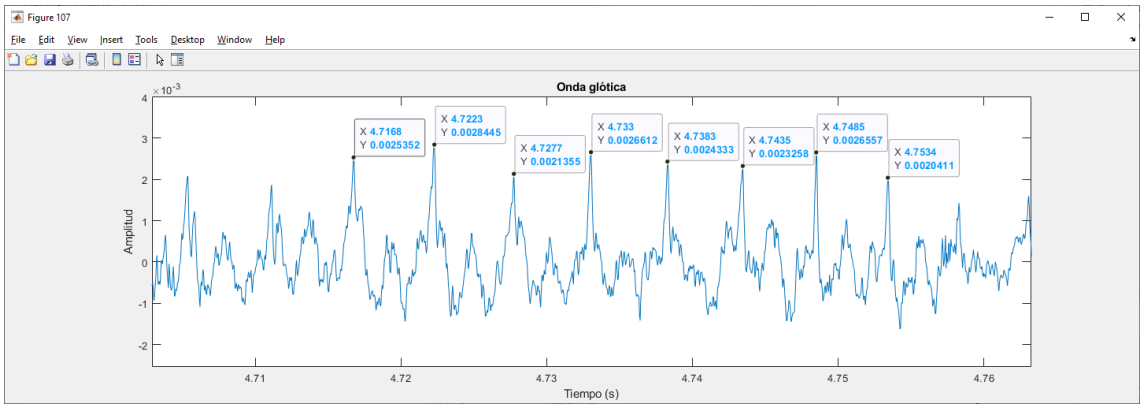

FIGURA 5 - FRECUENCIA DE ONDA GLÓTICA EN PULP.WAV

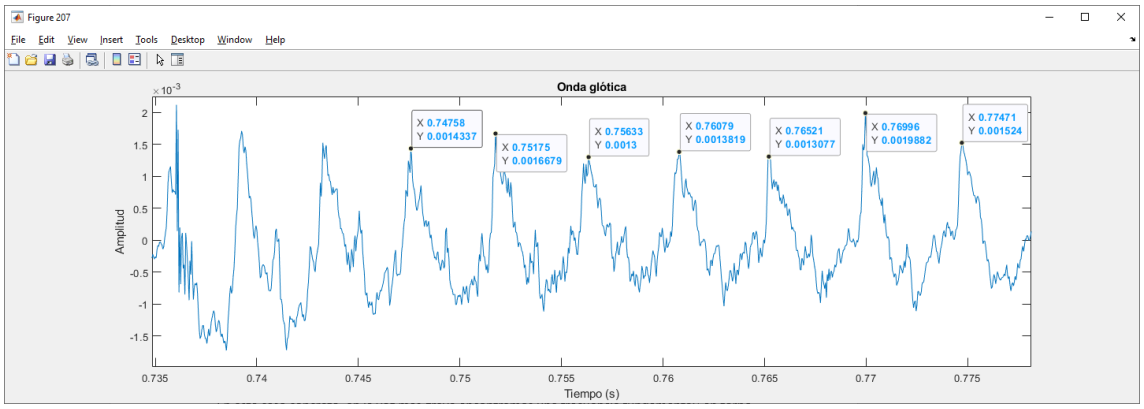

FIGURA 6 - FRECUENCIA DE LA ONDA GLÓTICA EN SHELDON.WAV

En este caso concreto, en la voz más grave encontramos una frecuencia fundamental[3] entre 180 y 200 Hz (Figura 5), y la voz más aguda alrededor de los 210 y 230 Hz (Figura 6), por lo que la diferencia es notable pero no es especialmente grande.

En cuanto al tracto, si observamos los espectrogramas:

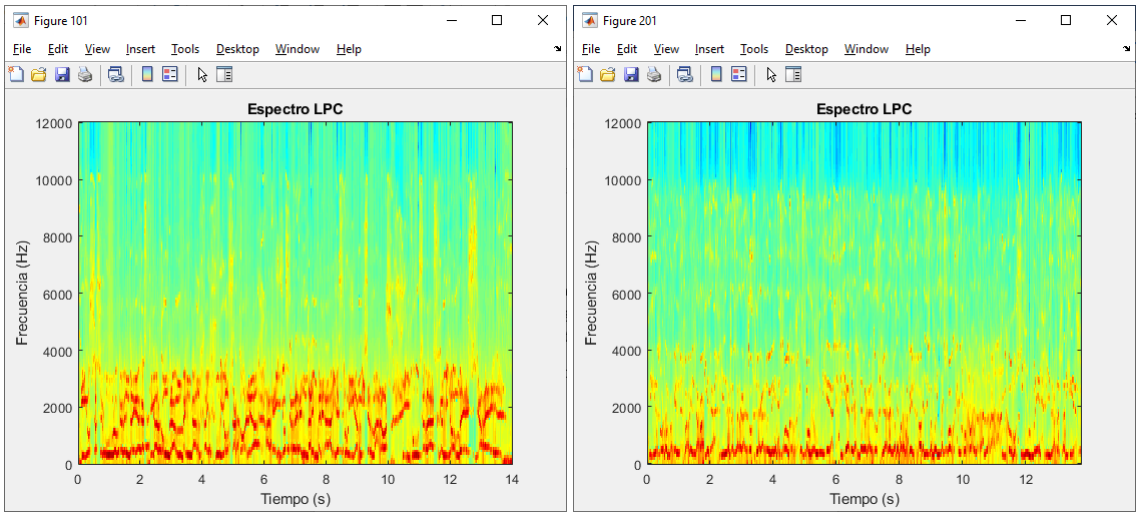

FIGURA 7 – ESPECTROGRAMAS DEL TRACTO: PULP.WAV (IZQ) Y SHELDON.WAV (DER)

Vemos que el segundo audio realza, en general, las frecuencias bajas, encontrando mucha energía en la línea de los 400 Hz. Esta línea también aparece en el audio 1 (de forma más irregular), pero además tenemos muchos más puntos elevados a lo largo de las bandas de frecuencia, hasta aproximadamente los 3000 Hz (un poco más).

De forma que, al mezclar el pulso del audio 1, con el tracto del audio 2, tendremos un pulso ligeramente más grave en el que se van a realzar menos frecuencias. En efecto, al reproducir el audio recompuesto con el pulso 1 y el tracto 2, la voz generada (en las partes en las que se entiende) es muy similar a la original, con una pérdida de calidad cuya causa puede atribuirse al

[3] Al tratarse de un audio complejo hay variación y es difícil detectar la frecuencia fundamental. Se ha tratado de escoger puntos donde más se alarguen las vocales. Se trata de una muestra, el tono puede cambiar a lo largo de la frase. Las frecuencias obtenidas son inesperadamente altas.

ruido generado en el proceso de mezcla de dos audios con frases distintas, y al tracto vocal que hace resonar menos frecuencias.

En el caso contrario (tracto 1 – pulso 2), tendremos un pulso más agudo con un tracto que hace resonar más frecuencias. En este caso, se oye la frase del audio 2 con una voz más aproximada a la del audio 1, debido a que le tracto del audio 1 resalta las frecuencias de dicha voz. En determinados momentos la frase resulta ininteligible debido a que se mezclan ambas frases. Además, en determinados momentos, la voz (tracto) de un audio y otro parecen alternar; lo que, posiblemente, se deba al descuadre que hay entre los audios. Como se trata de frases diferentes, el tracto modelado no está adaptado al pulso del otro audio, por lo que hay momentos en los que la aplicación del tracto 1 no está realzando apenas frecuencias y no influye en el pulso 2, por lo que la voz original reaparece.

En cualquier caso, se puede ver la capacidad y potencia de LPC, que separando y mezclando tracto y pulso de dos frases diferentes, es capaz de recomponer el audio. Con cierta distorsión, pero de forma (en gran medida) inteligible.

Mezcla de secuencia de palabras

Otra de las pruebas realizadas ha sido la recomposición de audios a partir de dos audios de diferentes locutores en los que se dice la misma secuencia de palabras separadas en el tiempo: nevera – aldaba – Lugo – barco.

El archivo "*palabras_m.wav*" contiene la secuencia de palabras producida por una mujer (que llamaremos "audio 1"), y "*palabras_h.wav*" contiene la misma secuencia producida por un hombre (que llamaremos "audio 2").

El análisis en este caso resulta más sencillo ya que es una locución más breve y sencilla, y no está siendo interpretada. Al ser las mismas palabras en ambas señales, el análisis nos dará más información y los resultados serán mejores.

Visualizando el pulso glótico y su onda, podemos diferenciar claramente las silabas pronunciadas. En estas locuciones resulta mucho más sencillo identificar el tono fundamental, siendo (para la sílaba "al") en el audio 1 en torno a 180 Hz (Figura 8), y alrededor de 110 Hz en el audio 2 (Figura 9).

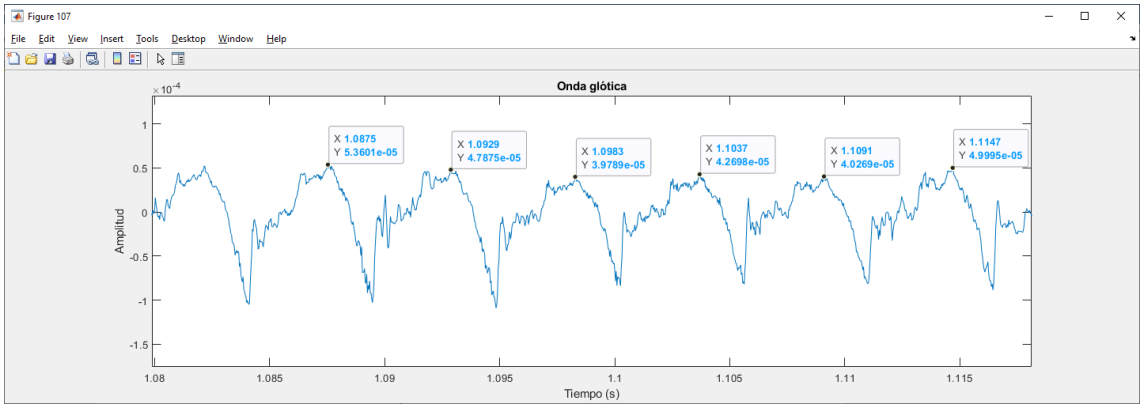

FIGURA 8 – FRECUENCIA DE LA ONDA GLÓTICA EN PALABRAS_M.WAV

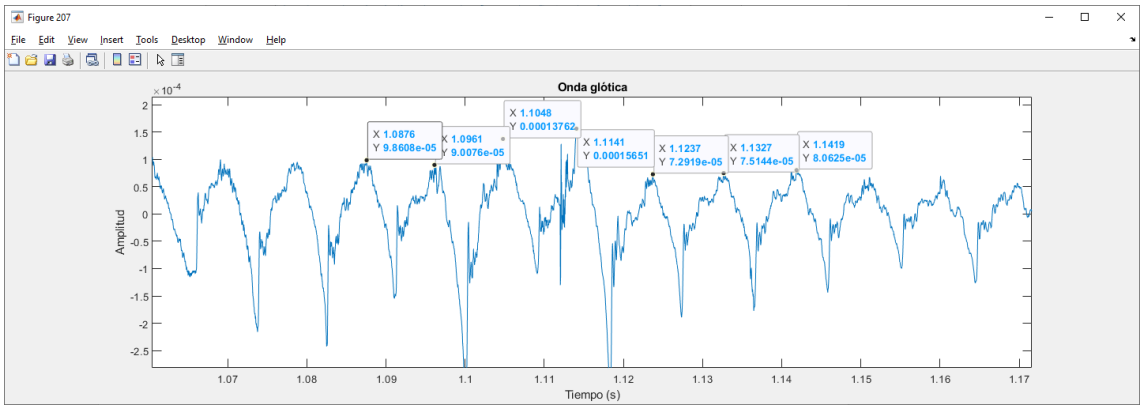

FIGURA 9 - FRECUENCIA DE LA ONDA GLÓTICA EN PALABRAS_H.WAV

Atendiendo al tracto vocal:

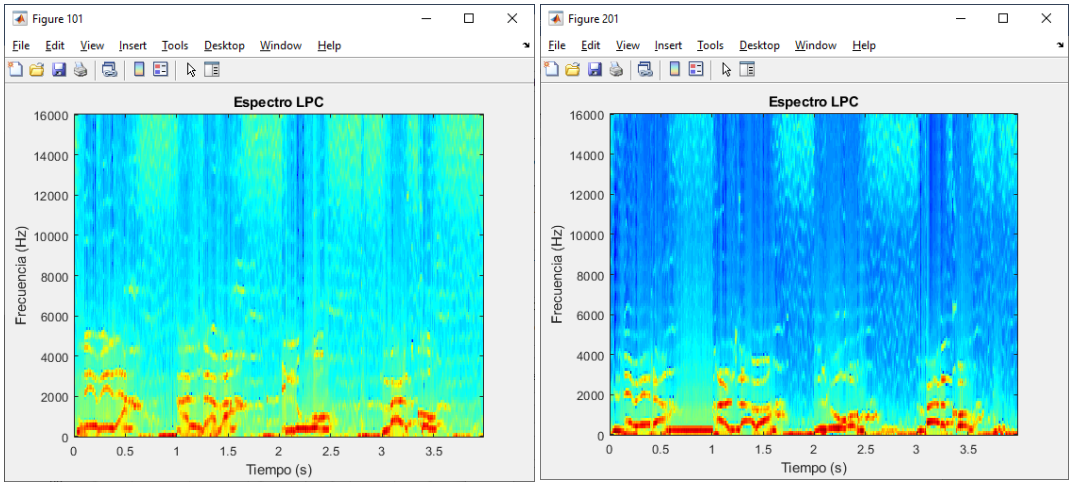

FIGURA 10 – ESPECTROGRAMAS DEL TRACTO: PALABRAS_M.WAV (IZQ) Y PALABRAS_H.WAV (DER)

En primer lugar, con un vistazo genera, vemos que las frecuencias altas, a partir de 5000 Hz están de color azul oscuro en el audio 2, mientras que en el audio 1 están ligeramente más realzadas.

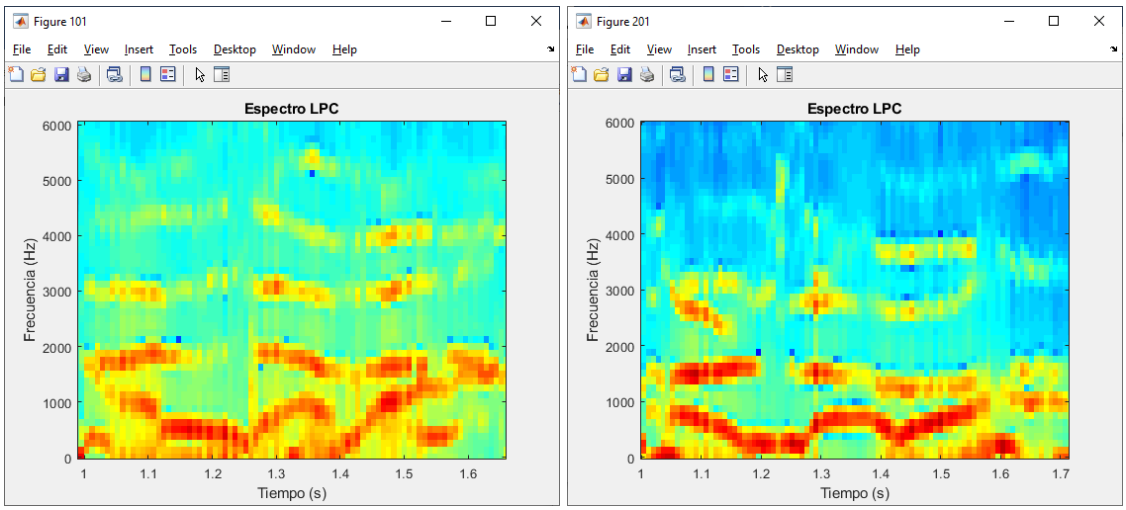

FIGURA 11 – ESPECTROGRAMA DE TRACTO AMPLIADO EN LA PALABRA "ALDABA"

Sin embargo, las frecuencias más importantes se encuentran antes de los 5000 Hz, donde podemos ver que el espectro sigue un patrón muy parecido con ligeras diferencias. En el tracto femenino del audio 1 aparecen formantes (sin demasiada energía) que no tenemos en el tracto masculino (audio 2), o encontramos esas líneas más dispersas. Por otro lado, los formantes de menor frecuencia aparecen más marcados en el tracto masculino que en el femenino, apareciendo en algunos casos a frecuencias ligeramente más bajas.

Un ejemplo de ello aparece en la Figura 11, donde los formantes inferiores aparecen más marcados en el tracto masculino, y los formantes superiores aparecen más dispersos.

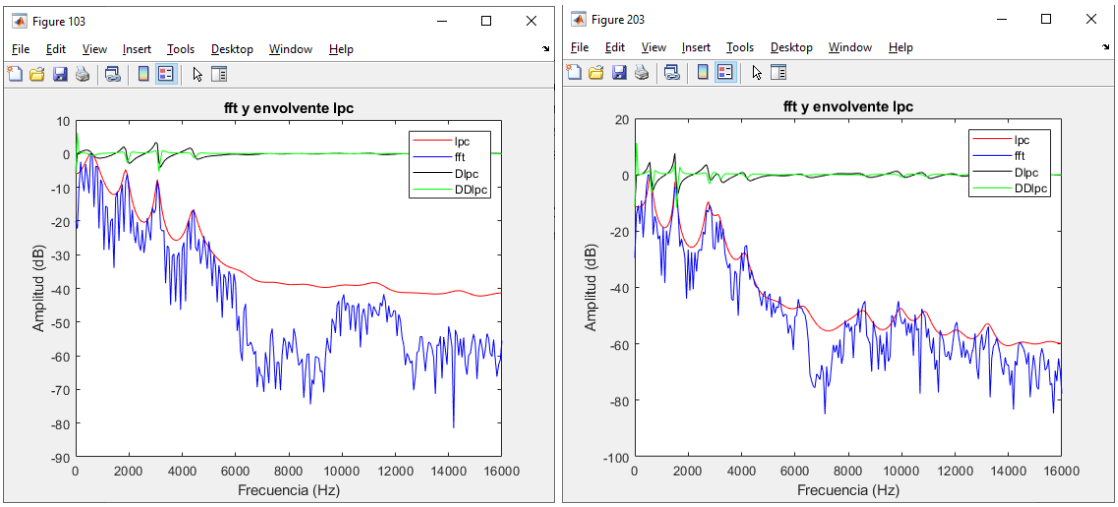

FIGURA 12 - FORMANTES EN EL SEGUNDO 1.3: PALABRAS_M.WAV (IZQ) Y PALABRAS_H.WAV (DER)

TABLA 1 - FORMANTES DEL TRACTO EN EL SEGUNDO 1.3

Tracto femenino		Tracto masculino	
562.5 Hz	0.0000 dB	562.5 Hz	0.0000 dB
1875.0 Hz	-4.7427 dB	1500.0 Hz	-2.5831 dB
3062.5 Hz	-7.6778 dB	2750.0 Hz	-9.5707 dB
4375.0 Hz	-16.8267 dB	4125.0 Hz	-27.6943 dB

La Figura 12 muestra en detalle los formantes de ambas señales en el segundo 1.3[4], cuyos datos concretos vienen recogidos en la Tabla 1. Podemos ver una primera línea que coincide en frecuencia (562.5 Hz) e intensidad. Sin embargo, los dos formantes siguientes se encuentran a menos frecuencia en el tracto masculino, con una diferencia de aproximadamente 300 Hz. Y, finalmente, vemos un formante pasando los 4000 Hz, que además de tener mayor frecuencia en el tracto femenino, tiene una amplitud mucho más alta.

En este punto, es interesante comentar la importancia de elegir adecuadamente el número de coeficientes LPC. Un número muy bajo resultará en una gráfica subajustada y poco o nada

[4] Las gráficas corresponden a la trama número 162 (sin emplear filtro de preénfasis y empleando 32 coeficientes LPC) producto del cálculo: $1{,}3\ s \cdot \frac{32000\ muestras}{1\ s} \cdot \frac{1\ trama}{256\ muestras}$

representativa de los formantes; mientras que un número alto de coeficientes provocará un sobreajuste, haciendo que nos indique formantes que realmente no hay.

Como consecuencia, reconstruir la señal con el tracto femenino (1) y el pulso masculino (2), tendremos un pulso más grave, con un tracto vocal que potencia las frecuencias un poco más agudas, impulsando frecuencias que inicialmente pasaban desapercibidas. El resultado (con 32 coeficientes LPC) es completamente inteligible, aunque la voz generada resulta extraña y un poco robótica. El proceso genera algo de distorsión, que es eliminada prácticamente al completo al activar los filtros de preénfasis y deénfasis con polos de 0.95.

Al mezclar el tracto masculino (2) con el pulso femenino (1) el resultado final resulta más natural al oído, asemejándose más al de una persona. Hay que tener en cuenta que, en este caso el tracto está eliminando algunas resonancias altas (y desplazando otras a frecuencias más bajas), pero no añade nuevas resonancias como sí ocurría en el caso anterior. Por ello, obtenemos una voz algo más grave que la del pulso original, pero que sigue resultando natural al oído.

Mezcla con instrumentos

Si mezclamos una de las señales anteriores (*palabras_h.wav*, por ejemplo) con una nota musical, también obtenemos resultados interesantes.

En este caso, contamos con 6 cortes que corresponden a las cuerdas de una guitarra al aire, y cuyas frecuencias deben ser:

TABLA 2 - ARCHIVOS Y FRECUENCIAS DE GUITARRA

Archivo	Nota	Frecuencia
nota_E_82-42Hz.wav	Mi1 (E2)	82.4069 Hz
nota_A_110Hz.wav	La1 (A2)	110.0000 Hz
nota_D_146-83Hz.wav	Re2 (D3)	146.8320 Hz
nota_G_196Hz.wav	Sol2 (G3)	195.9980 Hz
nota_B_246-94Hz.wav	Si2 (B3)	246.9420 Hz
nota_e_329-63Hz.wav	Mi3 (E4)	329.6280 Hz

En el caso de la guitarra, el pulso lo determinará la cuerda que se toque, que estará afinada para una frecuencia concreta como se ha visto en la Tabla 2; equivalente a la cuerda vocal humana. El tracto, corresponde a la caja de la guitarra, que reforzará unas frecuencias u otras dependiendo de su forma y tamaño. Como se ha mencionado anteriormente, cuando mayor es el tracto vocal, más bajas serán las frecuencias que se refuerzan y más grave el resultado.

Al contrario que en el caso anterior en el que se mezclaban voces de hombre y mujer, el “tracto” de una guitarra (la caja) es mucho más grande que el tracto vocal humano. Esto va a determinar claramente el resultado.

En primer lugar, se mezcla el pulso humano (archivo *palabras_h.wav*), con el tracto de guitarra en el archivo de la nota La1, nota que tiene una frecuencia de 110 Hz (similar al pulso de la señal humana) como se ha comprobado en la Figura 13.

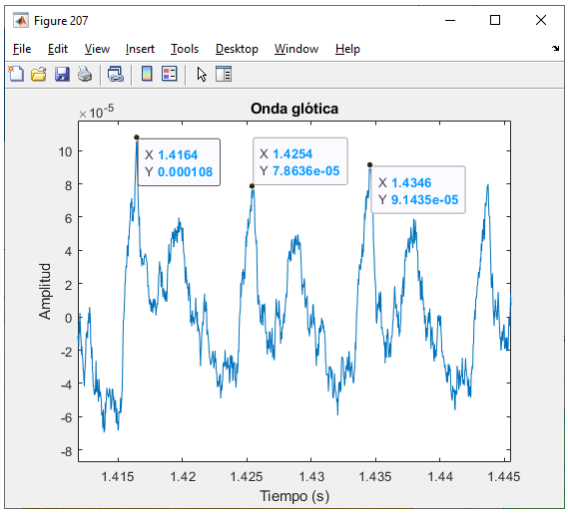

FIGURA 13 - FRECUENCIA DE ONDA DE 110 HZ EN LA1

Este tracto de guitarra (Figura 14), realza las frecuencias 500 Hz y, en menor medida, 1250 Hz (se aprecian al principio otras frecuencias por el golpe inicial hasta que la cuerda se estabiliza). Podemos ver estas frecuencias en el gráfico de formantes de la Figura 14, que corresponde al segundo 1.1 (trama 137).

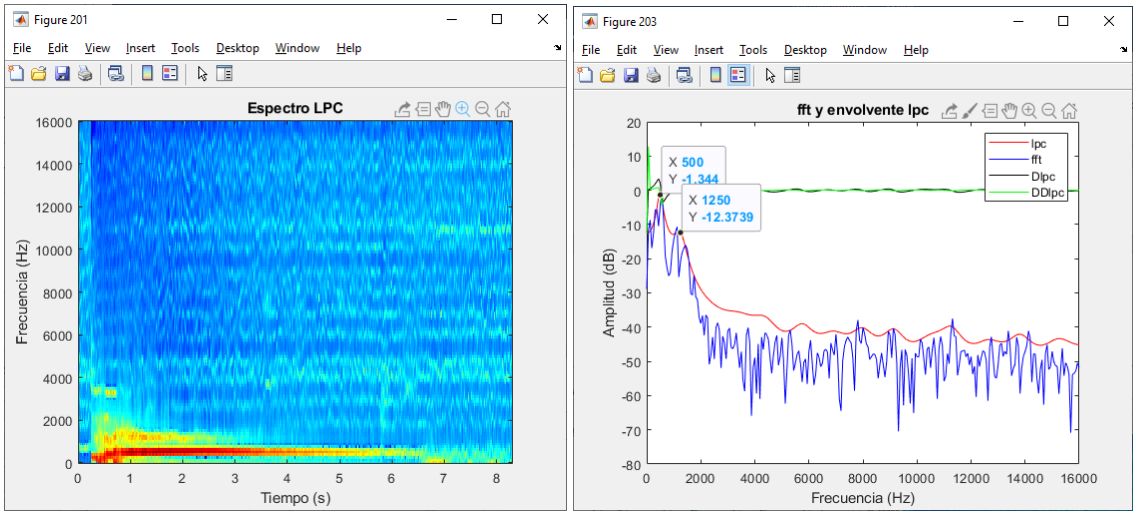

FIGURA 14 – TRACTO DE GUITARRA Y FORMANTES (LA1 - 110HZ)

Unos formantes claros y definidos son un factor positivo en una guitarra, ya que nos va a dar la nota con un sonido más limpio y claro. Sin embargo, al juntarlo con el tracto de voz, apenas va a realzar frecuencias, y esto se traduce en un sonido poco inteligible. El pulso cuenta con parte de la información vocalizada, siendo mayor cuantos menos coeficientes LPC empleemos (como se explicó al inicio de Mezcla de diferentes frases), y es por ello que se distinguen las palabras vocalizadas. En un caso ideal, el pulso no tendría información vocalizada y el resultado sería similar a cuerdas vibrando a ciertas frecuencias resonando en una caja de guitarra.

En el caso contrario (pulso de guitarra a 110 Hz y tracto vocal masculino), el sonido producido es similar al original con un efecto extraño y artificial ya que habitualmente el pulso de voz tiene

una frecuencia aproximada alrededor de la que oscila según entonemos, mientras que el pulso de la guitarra en este caso tiene una frecuencia exacta de 110 Hz casi constantes a lo largo de toda la grabación.

Además, el principio del audio de guitarra, resuena fuertemente con el golpe de la cuerda, y el pulso se atenúa rápidamente a lo largo del tiempo.

Para el resto de notas, juntar el pulso glótico humano con el tracto de la guitarra tiene un resultado similar[5], donde predomina el formante de 500 Hz.

La recomposición de pulso de guitarra y tracto vocal humano funciona de la misma manera, pero el resultado es completamente diferente. En última instancia, lo que está ocurriendo es que la voz queda afinada en la frecuencia de una nota concreta, similar al funcionamiento del autotune. Lo que nos permite incluso mezclar una melodía (como la del archivo *guitarra.wav*) con una frase (como *prat.wav*) y obtener, de la manera que hemos visto, una frase resultante en la que el locutor parece que está cantando.

En definitiva, como hemos visto es posible separar y recomponer el tracto y el pulso (con mayor o menor acierto) mediante LPC. Que constituye un mecanismo de modelado del tracto vocal sencillo y a la vez potente con múltiples usos como hemos visto (ingeniería inversa para obtener y separar el pulso, análisis de formantes, compresión de señal…), y cuyas aplicaciones de ingeniería inversa son, a su vez, diversas como cambio de locutor, síntesis de voz o autotune, entre otras.

[5] En algunos casos, como el archivo *nota_e_329-63Hz.wav* se aprecia la voz de forma muy diferente al principio; esto se debe a un tramo de silencio antes de que empiece a sonar la nota.

Referencias

[1] Á. Madrid Lara, «Modelo de fuente glotal para extraer características de la identidad del locutor,» Septiembre 2015. [En línea]. Available: http://oa.upm.es/39931/.

[2] C. Sanz Moreno, «Diseño de un sistema de análisis de formantes de voces,» Abril 2017. [En línea]. Available: http://oa.upm.es/52735/.

[3] M. Arjona Ramírez, «A Levinson algorithm based on an isometric transformation of Durbin's,» *IEEE Signal Processing Letters,* vol. 15, pp. 99--102, 2008.

[4] L. Rabiner, B. Atal y S. MR, «LPC Prediction Error—Analysis of Its Variation with the Position of the Analysis Frame,» *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 25, pp. 434 - 442, 1977.

[5] D. Kalogiros, «Pre–emphasis - Signal Processing,» Matlab Answers - Matlab Central, 11 Agosto 2018. [En línea]. Available: https://es.mathworks.com/matlabcentral/answers/414488-pre-emphasis-signal-processing#answer_332419.