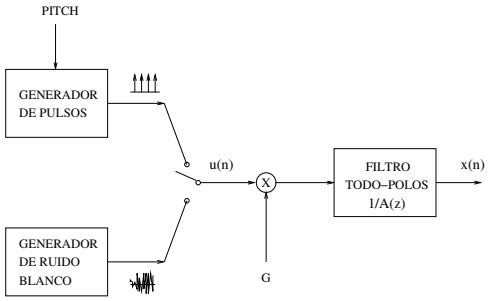
PRÁCTICA 5: PROCESAMIENTO DIGITAL DE VOZ

1. Introducción: Modelo Digital de Producción de Voz

En la figura de abajo se muestra un modelo digital de producción de voz usualmente conocido como modelo LPC (linear prediction coding). Mediante este modelo se supone que la señal de voz $x(n)$ es producida por un filtro digital todo-polos $h(k)$ (representando al tracto vocal) excitado por una señal $u(n)$, que puede ser de dos tipos:



1. Un ruido blanco (imitando flujo de aire procedente de los pulmones) para los sonidos sordos (/s/, /f/, ...).
2. Un tren de impulsos (imitando la vibración de las cuerdas vocales) para los sonidos sonoros (/a/,/m/,/b/,...).

En el caso de sonidos sonoros, el tren de impulsos tendra una frecuencia igual a la de vibración de las cuerdas vocales, conocida como *frecuencia de pitch*. Su inversa se denomina *periodo de pitch* o, simplemente, *pitch*.

2. Estimación Espectral de Voz

2.1. Periodograma

Como en tantas otras aplicaciones, para el estudio de las señales de voz es muy útil disponer de una estimación de la *densidad espectral de potencia* (PSD). La forma más sencilla de obtener esta estimación es mediante el *periodograma*, que se obtiene a partir de la transformada de Fourier $\mathscr{F}$ de un segmento de señal de N muestras (o, equivalentemente, a partir de las estimas sesgadas de la autocorrelación) como:

$$P_x(\omega) = \mathscr{F}[\hat{r}_x(k)] = |\mathscr{F}[x(n)]|^2/N. \tag{1}$$

La transformada $\mathscr{F}$ puede implementarse mediante una FFT de L puntos, por lo que se obtiene en las frecuencias $\omega_k = 2k\pi/L$ con $k = 0, 1, \ldots, L-1$. Normalmente se usará un valor alto para L para reflejar la continuidad de la variable ω (usaremos $L = 1024$).

En esta parte de la práctica se aplicará el método del periodograma para la estimación del espectro de potencias de un segmento de voz. Este segmento está formado por 256 muestras de la vocal /e/ muestreada a $F_s = 8$ KHz. Para cargar y visualizar la señal se ejecutará:

Python:
```
e=np.loadtxt('voc_e.asc')
plt.plot(e)
```

MatLab:
```
e=load('voc_e.asc')
plot(e)
```

La señal (almacenada en la variable e) presenta un aspecto cuasi-periódico. El periodo de la señal vocal se denomina pitch, definido anteriormente.

Realizaciones prácticas:

1. Medir el pitch de la vocal directamente sobre la señal.
2. Posteriormente se obtendrá el periodograma de la vocal aplicando la expresión (1). Para la visualización de espectros es conveniente usar la escala decibélica ($10\log_{10} P_x(\omega)$) y bastará con mostrar el espectro en el intervalo de frecuencias $[0, \pi]$.
3. El espectro proporciona información sobre la señal que sería difícil o imposible extraer en el dominio del tiempo:
   - Podemos observar una estructura fina o rizado (armónicos) que es debida a la excitación. Podemos medir la frecuencia de pitch como la diferencia entre dos picos consecutivos del rizado y comparar con el valor del pitch obtenido directamente sobre la señal.
   - Obviando el rizado, se pueden distinguir una serie de picos suaves que corresponden a las frecuencias de resonancia del tracto vocal, también conocidas como *formantes*. Medir de forma aproximada dichos formantes.

2.2. Espectro LPC/AR

Diversas aplicaciones requieren la determinación del espectro de voz obtenido del filtro todo-polos del modelo LPC de voz (prescindiendo de la excitación). Para sonidos sonoros, este espectro se obtiene de la respuesta en frecuencia de un modelo determinista todos-polos, mientras que en el caso de sonidos sordos se trataría de un espectro basado en un modelo AR de proceso aleatorio. En ambos casos la solución es idéntica y se conoce normalmente como *espectro LPC* o AR de voz.

Realizaciones prácticas:

1. Obtener el espectro AR(p=12). Para ello es necesario obtener los parámetros σ^2 y a_k ($k = 0, \ldots, p$) del modelo AR. Pueden obtenerse como en la práctica P2 (a partir de los parámetros AR estimados a partir de la autocorrelación sesgada) o, directamente, usando el comando *aryule*:

   ```
   a,sig2,k=aryule(e,p)
   ```

   donde `a` es el vector con los coeficientes a_k (coeficientes LPC), `sig2` es la potencia σ^2 del ruido de excitación del modelo AR, y `k` el vector con los coeficientes de reflexión correspondientes (el comando aplica resolución eficiente por método de autocorrelación y algoritmo de Levinson-Durbin). MatLab: comando homónimo (`aryule`).

2. Comentar las diferencias entre la PSD no paramétrica del periodograma y la PSD paramétrica obtenida con el modelo AR (superponer gráficas). Comparar los valores de los formantes obtenidos en cada método.

3. Análisis Homomórfico

Hemos visto que el periodograma nos proporciona información acerca de los parámetros del modelo de producción de voz, tanto de la excitación (pitch) como del filtro digital que representa el tracto vocal (formantes). Sin embargo, estas informaciones aparecen bastante entremezcladas. Mediante el *análisis homomórfico* es posible obtener una separación más clara de la excitación $U(\omega)$ y el filtro $H(\omega)$. Para ello basta aplicar una función logarítmica sobre el periodograma $P_x(\omega)$ de la señal. El logaritmo consigue descomponer el espectro en dos sumandos, uno debido al tracto vocal y otro a la excitación, como se demuestra a continuación:

$$P_x(\omega_k) = |H(\omega_k)|^2 P_u(\omega_k) \tag{2}$$

$$\log P_x(\omega_k) = \log |H(\omega_k)|^2 + \log P_u(\omega_k) \tag{3}$$

El análisis homomórfico se completa con la aplicación de la DFT inversa:

$$c_x(n) = IDFT[\log P_x(\omega_k)] \tag{4}$$

El resultado de esta operación sobre el periodograma es una nueva señal que recibe el nombre de *cepstrum FFT*. El dominio n del cepstrum es de nuevo el tiempo (ya que hemos aplicado una IDFT), pero se suele denominar *cuefrencia* para diferenciar claramente el cepstrum de la señal original. El cepstrum $c_x(n)$ tiene dos componentes, $c_h(n)$ y $c_u(n)$, debidas al filtro vocal (correlaciones de retardo corto) y a la excitación (correlaciones de retardo largo), respectivamente:

$$c_x(n) = IDFT[\log P_x(\omega_k)] = IDFT[\log |H(\omega_k)|^2] + IDFT[\log P_u(\omega_k)] = c_h(n) + c_u(n) \tag{5}$$

En la práctica es sencillo separar visualmente ambas componentes de $c_x(n)$, ya que la excitación contribuye esencialmente a componentes de alta cuefrencia, mientras que el filtro contribuye a las bajas. Otra ventaja de la representación cepstral es que permite medir distancias entre espectros logarítmicos como distancias euclídeas.

Realización práctica:

1. Obtener y visualizar el cepstrum FFT de la misma vocal de la sección anterior. Para mayor claridad dibujar únicamente los 100 primeros coeficientes excluyendo el $c_x(0)$, cuyo rango dinámico es muy superior al del resto.

2. Determinar de nuevo el pitch de la vocal como el valor de cuefrencia al que se produce el mayor pico del cepstrum para valores superiores a $n = 20$ (este valor de cuefrencia correspondería a una frecuencia de pitch de $f = F_s/n = 400$ Hz, que está por encima de los valores típicos del pitch de una persona adulta).

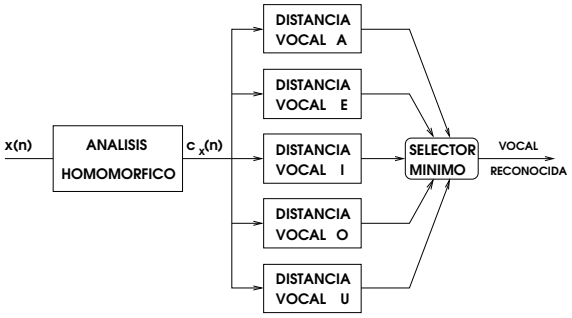
3. El cepstrum también puede obtenerse a partir del espectro LPC (AR), en lugar del periodograma. En este caso recibe el nomre de *cepstrum LPC*, y solo contiene información relativa al tracto vocal. Obtener este nuevo cepstrum y superponer al anterior. Comentar el resultado.

4. Reconocimiento de Vocales

Para el reconocimiento de señales de voz, la información relevante es la relativa al tracto vocal, ya que es la que define el tipo de sonido que se ha emitido. Por el contrario, la información relativa a la excitación no es útil, ya depende de factores altamente variables como la entonación, sexo del locutor, estado emocional del locutor, etc. Por ello, una buena manera de representar la información relativa exclusivamente al tracto vocal es mediante un vector de parámetros que contenga los primeros L coeficientes cepstrales $(c(1), c(2), \ldots, c(L))$, siendo L un número pequeño (típicamente entre 12 y 20). El primer coeficiente cepstral $c(0)$ tampoco se suele incluir en el vector, ya que está relacionado con la energía de la señal, que es también un parámetro sometido a una alta variabilidad. Usaremos el **cepstrum LPC** ya que, al proceder de un espectro suavizado que excluye la excitación, suele proporcionar mejores resultados que el cepstrum FFT.

Realización práctica:
Se implementará un reconocedor/clasificador de vocales simple. Para ello se dispone de 5 segmentos vocálicos de 256 muestras, correspondientes a los 5 fonemas vocálicos (ficheros voc_a.asc, voc_e.asc, voc_i.asc, voc_o.asc, voc_u.asc). El diagrama de bloques del reconocedor se muestra en la figura de abajo. Utilizando el vector de coeficientes cepstrales LPC ($L = 12$) de cada una de esta vocales como referencia, ha de determinarse a qué vocal corresponde un segmento incógnita (voc_x.asc). Para ello, deben computarse las distancias cepstrales entre este segmento y cada una de las referencias, seleccionando como vocal reconocida aquella para la que se obtenga distancia mínima. La distancia cepstral entre un vector de referencia $\mathbf{c_r}$ y otro incógnita $\mathbf{c_x}$, se obtiene como una distancia euclídea:

$$d_C(\mathbf{c}_x, \mathbf{c}_r) = \sum_{n=1}^{L} (c_x(n) - c_r(n))^2 = \|\mathbf{c}_x - \mathbf{c}_r\|^2 \tag{6}$$



Módulos y funciones para implementación en Python

La práctica requiere la instalación mediante *pip* del módulo *spectrum* (consultar seminario de python). Una vez instalado, se realizarán las siguientes importaciones:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from spectrum import *
```