



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2024 - 2^{do} cuatrimestre

PROCESOS ESTOCÁSTICOS (TA132)

TRABAJO PRÁCTICO 3: SINTETIZACIÓN DE FONEMAS MEDIANTE PROCESOS AR

Fecha de entrega: 28/11/2024

ESTUDIANTES: Grupo 5

Giani, Tomás	107629
<code>tgiani@fi.uba.ar</code>	
Lalía, Manuel	107979
<code>mlalia@fi.uba.ar</code>	
Perassi, Felipe	107630
<code>fperassi@fi.uba.ar</code>	
Rodríguez, Maximiliano	107604
<code>masrodriguez@fi.uba.ar</code>	

Índice

1. Introducción	2
2. Desarrollo	2
2.1. Ejercicio 1	2
2.1.1. Ítem A	2
2.1.2. Ítem B	3
2.1.3. Ítem C	4
2.1.4. Ítem D	4
2.2. Ejercicio 2	5
2.2.1. Ítem A	5
2.2.2. Ítem B	6
2.3. Ejercicio 3	8
2.4. Ejercicio 4	9
2.4.1. Ítem A	9
2.4.2. Ítem B	11
3. Conclusiones	13

1. Introducción

En el presente trabajo práctico, se aborda la sintetización de fonemas mediante procesos autorregresivos, utilizando la técnica de *Linear Predictive Coding* (LPC). Esta técnica permite modelar la producción de fonemas como la salida de un sistema lineal e invariante en el tiempo (LTI) excitado por diferentes señales. Las vocales se representan mediante un tren de pulsos periódico, mientras que los fonemas sordos se modelan como ruido blanco gaussiano, reflejando así las características de cada tipo de fonema.

En primer lugar, se analiza el origen de las expresiones matemáticas a utilizar en todo el trabajo de estudio a fin de comprender lo que se está realizando. Luego, teniendo en cuenta lo ya desarrollado matemáticamente, a partir del *software* llamado *Matlab* se implementaron los incisos propuestos por la cátedra, los cuales incluyen la estimación de los parámetros LPC de distintos fonemas. Entre las tareas realizadas, se destacan la generación y comparación de periodogramas, la aplicación del método de *Welch* para estimar densidades espectrales de potencia (PSD) y el análisis de los resultados obtenidos.

Por último, se realizó una sintetización de las señales a partir del método descrito anteriormente, es decir, basándose en los parámetros LPC. También se generó una concatenación de todas las señales, suavizando bordes y cambiando las diferentes frecuencias de *pitch* de las vocales, lo cual influye en el tren de deltas de la entrada. Adicionalmente, se generó una señal concatenada que combina los fonemas sintetizados, aplicando un suavizado de bordes entre cada letra y variando las frecuencias de *pitch* de las vocales, lo cual influye en la modulación del tren de pulsos utilizado como señal de excitación.

2. Desarrollo

2.1. Ejercicio 1

2.1.1. Ítem A

A partir de la ecuación que describe un proceso AR-P, se opera multiplicando por la salida conjugada en un instante de tiempo $k \geq 0$ y, posteriormente, tomando esperanzas. De esta manera, lo que se busca es llegar a una expresión equivalente de la función de autocorrelación.

$$\begin{aligned}
 x(n) &= \sum_{i=1}^p a_i x(n-i) + Gu(n) \\
 x(n)x^*(n-k) &= \left(\sum_{i=1}^p a_i x(n-i) + Gu(n) \right) x^*(n-k) \\
 x(n)x^*(n-k) &= \sum_{i=1}^p a_i x(n-i)x^*(n-k) + Gu(n) x^*(n-k) \\
 \underbrace{E[x(n)x^*(n-k)]}_{R_x(k)} &= \sum_{i=1}^p a_i \underbrace{E[x(n-i)x^*(n-k)]}_{R_x(k-i)} + E[Gu(n) x^*(n-k)] \\
 R_x(k) &= \sum_{i=1}^p a_i R_x(k-i) + G E[u(n)x^*(n-k)] \tag{1}
 \end{aligned}$$

Una vez obtenido esto, se debe analizar el término $E[Gu(n) x^*(n-k)]$. Para esto, se tiene en consideración de que se trata de un sistema LTI y causal. Por lo tanto, la salida $x(n)$ se puede escribir en función de la convolución de la entrada con la respuesta al impulso desde cero a infinito.

$$\begin{aligned}
E[u(n) x^*(n-k)] &= E \left[u(n) \sum_{l=0}^{\infty} h^*(l) G u^*(n-k-l) \right] \\
E[u(n) x^*(n-k)] &= G \sum_{l=0}^{\infty} h^*(l) \underbrace{E[u(n) u^*(n-k-l)]}_{R_u(k+l)} \\
E[u(n) x^*(n-k)] &= G \sum_{l=0}^{\infty} h^*(l) R_u(k+l) \tag{2}
\end{aligned}$$

En función de este último resultado y entendiendo que $u(n)$ es un proceso blanco, se tiene que su función de autocorrelación tiene la siguiente forma.

$$R_u(k) = \sigma_u^2 \delta(k) \Rightarrow R_u(k) = \delta(k)$$

Esto se debe a que al ser $u(n)$ un proceso blanco sus muestras en diferentes instantes de tiempo están descorrelacionadas entre sí. Por ende, el único término que es distinto de cero es $R_u(k=0)$ que corresponde a la varianza del proceso $\sigma_u^2 = 1$, dado que $u(n) \sim \mathcal{N}(0, 1)$.

Volviendo a la ecuación 2, se tiene lo que se observa a continuación.

$$\begin{aligned}
E[u(n) x^*(n-k)] &= G \sum_{l=0}^{\infty} h^*(l) R_u(k+l) \\
E[u(n) x^*(n-k)] &= G \underbrace{\sum_{l=0}^{\infty} h^*(l) \delta(k+l)}_{h^*(-k)} \\
E[u(n) x^*(n-k)] &= G h^*(-k)
\end{aligned}$$

Considerando que se trata de un sistema causal, entonces para todo $k > 0$, la respuesta al impulso $h(-k) = 0$. De esta forma, se puede decir que $h(-k)$ es equivalente a $\delta(k)$.

$$\begin{aligned}
E[u(n) x^*(n-k)] &= G h^*(-k) \\
E[u(n) x^*(n-k)] &= G \delta(k)
\end{aligned}$$

A modo de concluir lo realizado en este punto del trabajo, se reemplaza todo lo analizado en la ecuación 1. Asimismo, se demuestra que se cumple con lo solicitado.

$$\begin{aligned}
R_x(k) &= \sum_{i=1}^p a_i R_x(k-i) + G E[u(n) x^*(n-k)] \\
R_x(k) &= \sum_{i=1}^p a_i R_x(k-i) + G^2 \delta(k)
\end{aligned}$$

2.1.2. Ítem B

Teniendo en cuenta el resultado obtenido en el inciso anterior y evaluándolo en $k=0$, se tiene la siguiente expresión.

$$\begin{aligned}
R_x(0) &= \sum_{i=1}^p a_i \underbrace{R_x(0-i)}_{R_x(i)} + G^2 \\
R_x(0) &= \sum_{i=1}^p a_i R_x(i) + G^2
\end{aligned}$$

En virtud de esto, se puede despejar G .

$$G = \left(R_x(0) - \sum_{i=1}^p a_i R_x(i) \right)^{\frac{1}{2}}$$

Además, como depende de la función de autocorrelación de $x(n)$ y esta se estima a partir de estimaciones del proceso $x(n)$, lo que se obtiene es una estimación de la ganancia G .

$$\hat{G} = \left(\hat{R}_x(0) - \sum_{i=1}^p a_i \hat{R}_x(i) \right)^{\frac{1}{2}}$$

2.1.3. Ítem C

Al igual que en el ítem anterior se parte desde la ecuación de la función de autocorrelación, es decir, de las ecuaciones de Yule Walker. En este caso, se expande el término en el que $k \neq 0$ de forma matricial. Cabe destacar que se debe tener principalmente en cuenta que $R_x(-k) = R_x(k)$ por simetría de la función de autocorrelación.

$$R_x(k) - \sum_{i=1}^p a_i R_x(k-i) = 0, \quad k > 0$$

$$\begin{bmatrix} R_x(1) \\ R_x(2) \\ \vdots \\ R_x(k) \end{bmatrix} - \begin{bmatrix} R_x(0) & R_x(1) & \dots & R_x(p-1) \\ R_x(-1) & R_x(2) & \dots & R_x(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(-p+1) & R_x(-p+2) & \dots & R_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = 0$$

Entonces, despejando el vector de los coeficientes, se tiene lo que se ve a continuación.

$$\underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}}_a = \underbrace{\begin{bmatrix} R_x(0) & R_x(1) & \dots & R_x(p-1) \\ R_x(-1) & R_x(2) & \dots & R_x(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(-p+1) & R_x(-p+2) & \dots & R_x(0) \end{bmatrix}}_{R^{-1}}^{-1} \underbrace{\begin{bmatrix} R_x(1) \\ R_x(2) \\ \vdots \\ R_x(k) \end{bmatrix}}_r \Rightarrow a = R^{-1} \cdot r$$

De igual forma que en el inciso anterior, este resultado depende de la función de autocorrelación de $x(n)$, la cual se estima según estimaciones del proceso. Teniendo en cuenta esto, lo que se obtiene es una estimación de los valores de los coeficientes.

$$\hat{a} = \hat{R}^{-1} \cdot \hat{r}$$

2.1.4. Ítem D

En primer lugar, a fin de hallar la densidad espectral de potencia, se debe tener en cuenta que los procesos AR-P son la salida de un sistema LTI. De esta manera, la PSD se puede determinar a partir de la siguiente expresión.

$$S_x(\omega) = |H(e^{j\omega})|^2 \cdot S_u(\omega) \quad (3)$$

Donde $H(e^{j\omega})$ es la respuesta en frecuencia y $S_u(\omega)$ la PSD de la entrada.

Ahora, se calcula la densidad espectral de potencia de la entrada, considerando que esta es la transformada de Fourier de la función de autocorrelación, mencionada anteriormente.

$$R_u(k) = \delta(k) \xrightarrow{\mathcal{F}} S_u(\omega) = 1$$

También se transforma por Fourier la ecuación que describe un proceso AR-P, con el fin de obtener la respuesta en frecuencia del sistema.

$$\begin{aligned} x(n) - \sum_{i=1}^p a_i x(n-i) &= G u(n) \\ \mathcal{F}\{x(n)\} - \sum_{i=1}^p a_i \mathcal{F}\{x(n-i)\} &= G \mathcal{F}\{u(n)\} \\ X(e^{j\omega}) - \sum_{i=1}^p a_i X(e^{j\omega})e^{-j\omega i} &= G U(e^{j\omega}) \\ X(e^{j\omega}) \left[1 - \sum_{i=1}^p a_i e^{-j\omega i} \right] &= G U(e^{j\omega}) \Rightarrow H(e^{j\omega}) = \frac{X(e^{j\omega})}{U(e^{j\omega})} = \frac{G}{1 - \sum_{i=1}^p a_i e^{-j\omega i}} \end{aligned}$$

Por lo tanto, reemplazando lo obtenido en la ecuación 3, se obtiene el resultado buscado.

$$\begin{aligned} S_x(\omega) &= |H(e^{j\omega})|^2 \cdot S_u(\omega) \\ S_x(\omega) &= \left| \frac{G}{1 - \sum_{i=1}^p a_i e^{-j\omega i}} \right|^2 \cdot 1 \\ S_x(\omega) &= \frac{G^2}{|1 - \sum_{i=1}^p a_i e^{-j\omega i}|^2} \end{aligned}$$

2.2. Ejercicio 2

2.2.1. Ítem A

En base a lo desarrollado en el ejercicio anterior, se crea una función *param_ar* que recibe una realización del proceso \mathbf{x} y el orden del sistema \mathbf{P} y calcula los parámetros para la LPC, es decir, los coeficientes $[a_1 \dots a_p]$ y la ganancia \mathbf{G} .

A continuación se puede observar el pseudocódigo sobre el cual se basa lo implementado en *Matlab*.

Algoritmo 1: Cálculo de los parámetros para un sistema AR-P

Input: Realización del proceso y orden del sistema

Output: Coeficientes y ganancia del sistema de orden indicado

```
/* 1. Se estima la autocorrelación del proceso */

/* 2. Se inicializa la matriz R con los valores de autocorrelación indicados */

/* 3. De la misma manera se define el vector r */

/* 4. Se calculan los coeficientes [a1 ... ap] de acuerdo a la ecuación hallada en el punto 1. Se
   invierte el signo para que coincida con el modo que matlab utiliza los parámetros de un sistema
   LTI */

/* 4. De la misma manera se calcula la ganancia 'G' */

/* 4. Se agrega el coeficiente a0 = 1 al vector de coeficientes para que pueda ser utilizado tal
   como se devuelve */
```

2.2.2. Ítem B

En este ítem, se calculan los parámetros del sistema LPC de orden $P = 20$ para cada pista de audio. Además, se generan los gráficos de la respuesta temporal de los audios utilizados, la estimación de cada una de las autocorrelaciones y sus respectivos periodogramas comparandolos con sus PSD teóricas.

De esta forma, se genera el código que sigue los siguientes lineamientos.

Algoritmo 2: Cálculos y gráficos del punto 2.b

Input: Realización del proceso y orden del sistema

Output: Gráficos mostrando la señal temporal, autocorrelación y PSD de los audios

```
/* 1. Para cada audio: */

/* 1.1 Se estiman los parametros mediante la función creada param_ar */

/* 1.2 Se estima la autocorrelación y la PSD */

/* 1.3 Se calcula la PSD teórica correspondiente a los parámetros obtenidos */

/* 1.4 Se grafica la señal temporal, la autocorrelación y la PSD */
```

A continuación se muestran los resultados generados para las distintas pistas de audio, separados en vocales y consonantes.

■ **Vocales:**

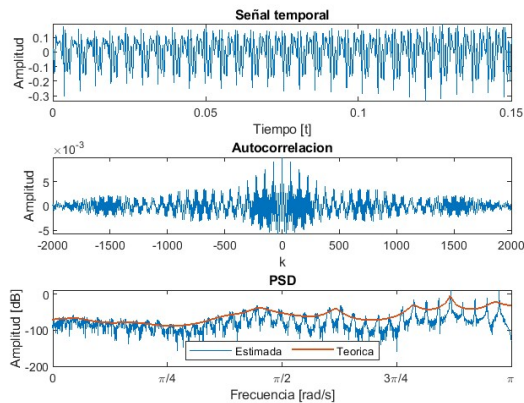


Figura 1: Análisis de la 'a'.

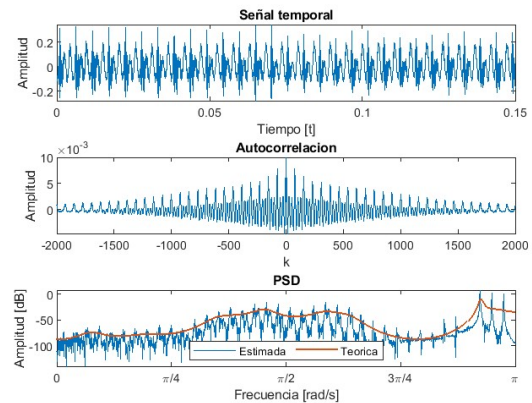


Figura 2: Análisis de la 'e'.

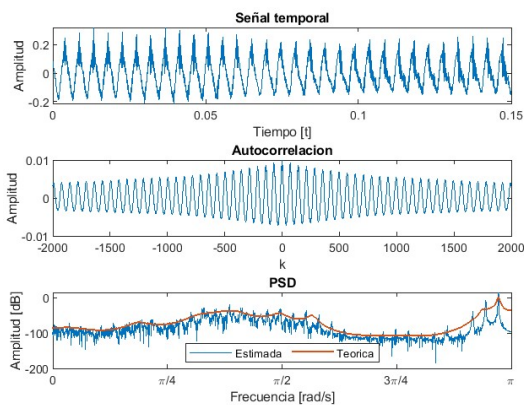


Figura 3: Análisis de la 'i'.

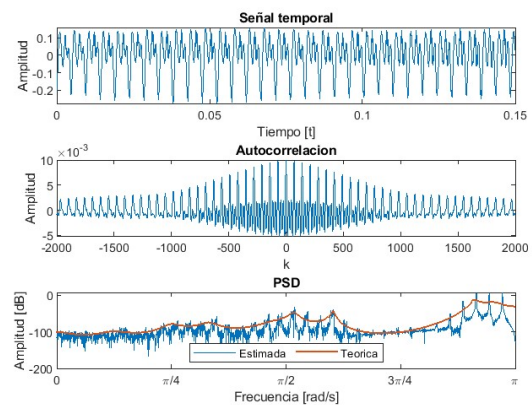


Figura 4: Análisis de la 'o'.

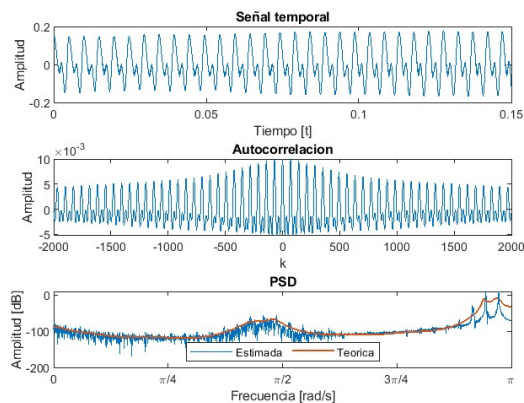


Figura 5: Análisis de la 'u'.

■ Consonantes:

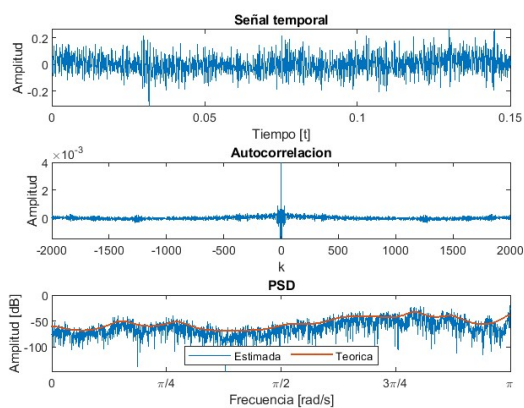


Figura 6: Análisis de la 'f'.

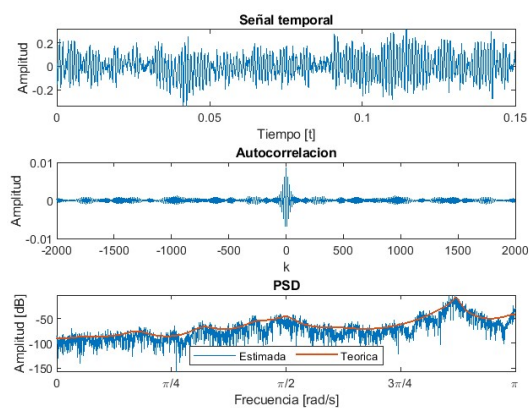


Figura 7: Análisis de la 'j'.

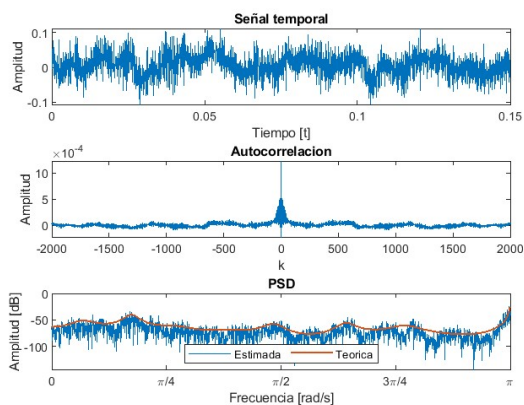


Figura 8: Análisis de la 's'.

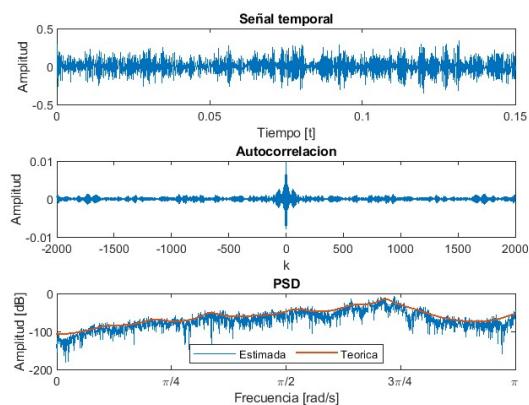


Figura 9: Análisis de la 'sh'.

Se puede ver que las vocales están mucho más autocorrelacionadas que las consonantes, ya que, tienen una amplitud importante para todos los valores de k , mientras que en las consonantes la autocorrelación tiende a cero cuando aumenta k .

Además, en los gráficos de la PSD de las vocales se observan picos distinguidos y equidistantes en el espectro. Esto caracteriza el comportamiento cuasi-periódico de las vocales.

Por otro lado, en las consonantes, se observa un espectro mucho mas 'plano' que contiene amplitud

similar para todo el rango de frecuencias. Esto indica la similitud de las señales sordas de las consonantes con el ruido blanco.

Finalmente, para ambos casos la PSD estimada se corresponde a la curva teórica que se calcula en base al sistema AR-P obtenido.

2.3. Ejercicio 3

Una forma de estimar una PSD es mediante el Método de *Welch*, el cual consiste en dividir el registro original de N puntos, en segmentos más cortos de M puntos, traslapando entre sí L muestras. Esta nueva variable L , se define como el cociente entre N y el paso entre segmentos K .

Posteriormente, se aplicó una ventana temporal a cada segmento. Luego, se calculó el periodograma para cada segmento ya ponderado con la ventana. El periodograma final se obtiene promediando todos los periodogramas parciales.

Las ecuaciones que describen al método de *Welch* son las siguientes.

- Para el segmento i -ésimo del proceso $x(n)$

$$x_i(n) = x((i-1)K + n) \quad (4)$$

$$seg_i(n) = x_i(n) \cdot v(n) \quad (5)$$

- Para el periodograma de i -ésimo segmento

$$P = \frac{1}{M} \cdot \sum_{t=1}^M |v(t)|^2 \quad (6)$$

$$\hat{S}_i(\omega) = \frac{1}{MP} |F[v(n) \cdot x_i(n)]|^2 \quad (7)$$

- Estimador mediante método de Welch

$$\hat{S}_W(\omega) = \sum_{i=1}^L \hat{S}_i(\omega) \quad (8)$$

Para observar los efectos del largo de la ventana de *Hamming*, se realizó el Método de *Welch* para los audios de los fonemas 'e' y 'sh', se comparó el resultado con la PSD teórica y se analizaron los resultados al usar un largo de ventana $M = \{10, 100, 1000\}$.

De esta manera, a partir del pseudocódigo que se ve a continuación, se genera el *script* del ejercicio.

Algoritmo 3: Estimación PSD por Welch para una vocal y una consonante

Input: Realización del proceso y orden del sistema

Output: Gráficos mostrando la señal temporal, autocorrelación y PSD de los audios

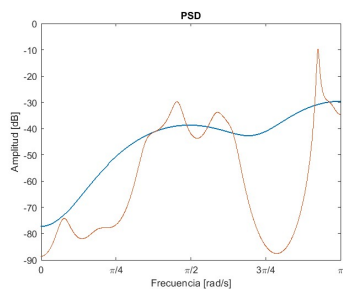
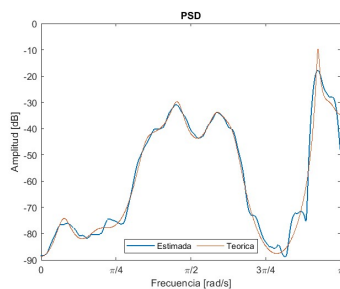
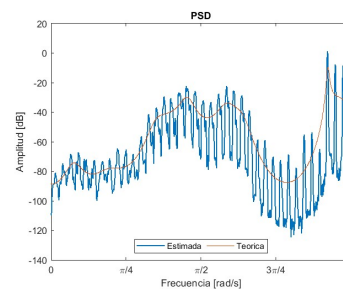
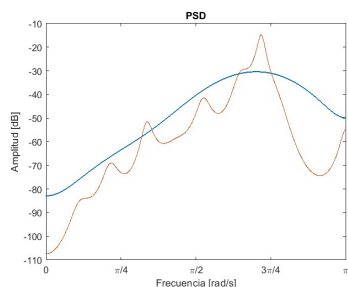
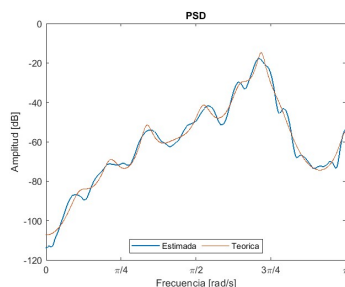
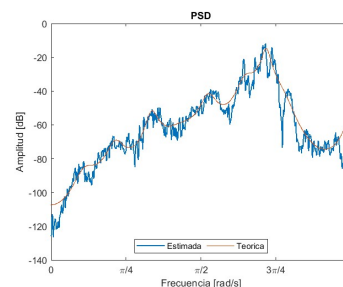
```
/* 1. Se define la ventana de Hamming de largo M */

/* 2. Se obtienen los L segmentos de largo M a partir de la señal de audio y se los multiplica por
la ventana */

/* 3. Se estima la PSD para cada segmento */

/* 4. Se promedian las L estimaciones de la PSD para obtener la estimación por el método de Welch
*/

/* 5. Se compara gráficamente la estimación obtenida con el resultado teórico */
```

Figura 10: $M_e = 10$ Figura 11: $M_e = 100$ Figura 12: $M_e = 1000$ Figura 13: $M_{sh} = 10$.Figura 14: $M_{sh} = 100$.Figura 15: $M_{sh} = 1000$.

Se observó que, a medida que M disminuye, el espectro sigue una tendencia de suavizarse progresivamente, a cambio de un ensanchamiento en frecuencia que provoca falta de resolución, debido a que disminuye la varianza. Aumentar M hará que tengamos una mayor resolución en frecuencia, a cambio de unos picos muy altos debido a la alta varianza.

2.4. Ejercicio 4

2.4.1. Ítem A

En este ejercicio, se sintetizaron señales de fonemas utilizando los parámetros LPC para una duración de 500 ms.

Se definió una matriz para almacenar las señales sintetizadas de cada fonema. Cada fila de esta matriz corresponde a un fonema, y cada columna representa las muestras de la señal en el dominio temporal.

Para las vocales, se generó un tren de pulsos periódicos con una frecuencia de pitch de 100 Hz:

- El periodo del tren de pulsos se definió como $T = \frac{1}{\text{pitch}}$ (es decir, $T = 10$ ms).
- La amplitud de cada pulso se ajustó según la relación $\sqrt{T \cdot F_s}$, donde F_s es la frecuencia de muestreo, para obtener una densidad espectral de potencia adecuada.
- Se construyó un vector de pulsos colocando un valor no nulo cada $\frac{F_s}{\text{pitch}}$ muestras.

En cambio, para las consonantes, se utilizó un vector de ruido blanco como señal de excitación del sistema LTI.

Teniendo esto en cuenta, la señal sintetizada para cada fonema se obtuvo utilizando un filtro LPC, cuya función de transferencia es la siguiente.

$$H(z) = \frac{G}{1 + \sum_{k=1}^p a_k z^{-k}}$$

De esta manera, la señal de salida se calculó filtrando el tren de pulsos (en el caso de las vocales) o el ruido blanco (en el caso de las consonantes) a través del filtro mencionado anteriormente.

A fin de evaluar las señales sintetizadas, se compararon las densidades espectrales de potencia con lo obtenido a través del código. En este instante, se tuvo en cuenta el cálculo de la PSD teórica cuando se trata de un sistema LTI.

$$S_x(\omega) = |H(e^{j\omega})|^2 \cdot S_u(\omega)$$

Las señales generadas se guardaron como archivos de audio (.wav) para poder escucharlas y verificar la validez de los resultados obtenidos. Cada archivo corresponde a un fonema. Esto se entrega en el archivo .zip en conjunto con los códigos implementados.

Por lo tanto, a continuación se puede visualizar el pseudocódigo utilizado en este inciso del trabajo.

Algoritmo 4: Síntesis y estimación de las señales

Input: Coeficientes a , ganancias G , frecuencia de muestreo F_s , longitud de FFT n_fft , número de muestras, frecuencia de pitch y número de fonemas

Output: Señales sintetizadas, gráficos de la PSD teórica y estimada, y archivos de audio generados

```
/* 1. Se crea la matriz para almacenar las señales sintetizadas */

/* 2. Para las vocales  $i = \{1, 2, 4, 6, 9\}$ : */
/* 2.1 Se genera el tren de pulsos con periodo correspondiente al pitch */

/* 2.2 Se filtra el tren de pulsos mediante el filtro LPC definido por  $a$  y  $G$  */

/* 2.3 Se calcula la PSD teórica */

/* 2.4 Se calcula la PSD estimada de la señal sintetizada */

/* 2.5 Se grafica la PSD teórica y estimada para cada fonema */

/* 3. Para las consonantes  $i = \{3, 5, 7, 8\}$ : */
/* 3.1 Se genera una señal de ruido blanco como entrada */

/* 3.2 Se filtra la señal de ruido utilizando el filtro LPC definido por  $a$  y  $G$  */

/* 3.3 Se calcula la PSD teórica */

/* 3.4 Se estima la PSD de la señal sintetizada */

/* 3.5 Se grafica la PSD teórica y estimada para cada fonema */

/* 4. Se guardan los audios sintetizados en archivos correspondientes a cada fonema */
```

A continuación se presentan los gráficos generados para cada fonema.

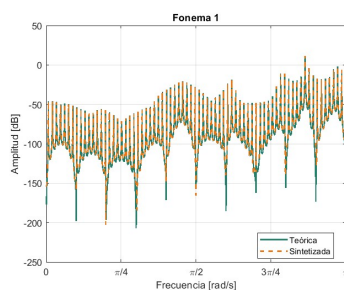


Figura 16: Periodograma 'a'

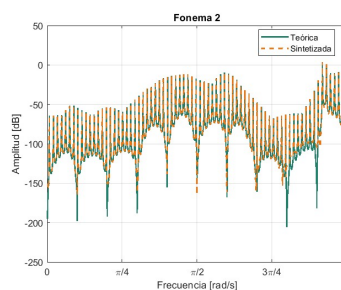


Figura 17: Periodograma 'e'

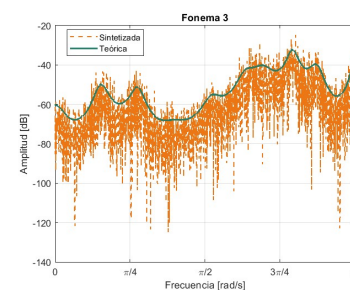


Figura 18: Periodograma 'f'

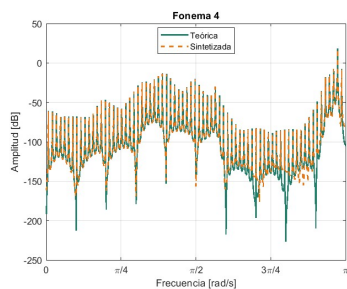


Figura 19: Periodograma 'i'

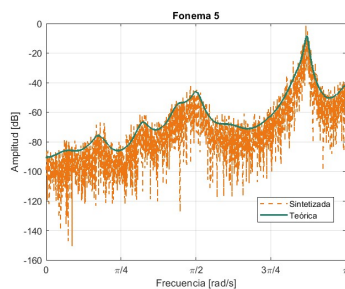


Figura 20: Periodograma 'j'

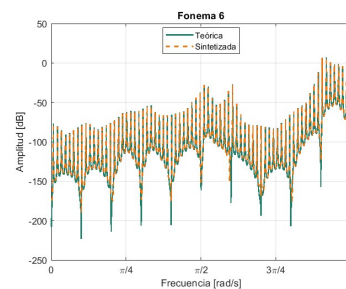


Figura 21: Periodograma 'o'

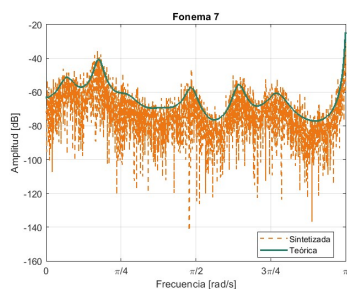


Figura 22: Periodograma 's'

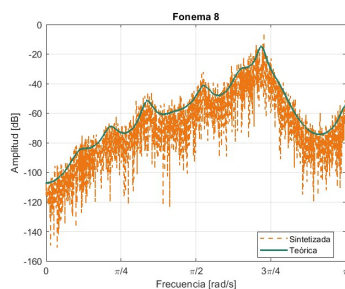


Figura 23: Periodograma 'sh'

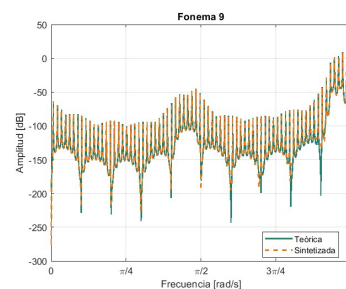


Figura 24: Periodograma 'u'

2.4.2. Ítem B

En este ítem, se generó una señal concatenada a partir de los nueve fonemas sintetizados previamente (cinco vocales y cuatro consonantes). Para esto, se aplicó un suavizado de bordes a cada fonema y se probaron dos configuraciones diferentes: una con el mismo pitch para todas las vocales y otra con un pitch diferente para cada vocal.

Se definió una señal de salida que incluye todas las señales suavizadas, concatenadas entre sí. La longitud total de esta señal se calculó como se puede observar a continuación.

$$\text{longitud total} = \text{muestras por fonema} \times \text{cantidad de fonemas}.$$

Para evitar discontinuidades en los extremos de cada fonema, se utilizó la función `suavizar_bordes()`, extraída del campus. Esta función modula la amplitud en las primeras y últimas muestras de cada fonema para generar transiciones más suaves entre las señales. En este caso, se utilizó un parámetro `fade = 30`, que indica la cantidad de muestras afectadas por el suavizado.

Para concatenar las señales suavizadas, se calcularon los índices de inicio y fin correspondientes a cada fonema en la señal final. Luego, las muestras suavizadas se copiaron en el lugar correspondiente de la señal concatenada.

A fin de analizar los efectos del pitch, en una primera instancia se utilizó un pitch de 100 Hz para todas las vocales.

Luego, se generaron nuevamente los fonemas vocales, pero esta vez utilizando un pitch diferente para cada vocal.

- *a*: 100 Hz
- *e*: 125 Hz
- *i*: 150 Hz
- *o*: 125 Hz
- *u*: 100 Hz

De esta forma, para cada vocal, se generó un tren de pulsos ajustado al pitch correspondiente y se sintetizó la señal mediante el filtro LPC. Mientras que, las consonantes se mantuvieron.

Algoritmo 5: Concatenación de los audios

Input: Los parámetros del LPC , el pitch para cada fonema y el fade para el suavizado

Output: Señal concatenada y gráficos

```
/* 1. Con el fade definido, se suavizan los bordes de las señales generadas en el punto anterior */

/* 2. Se concatenan las muestras en un único vector y se genera un audio */

/* 4. Se definen pitches distintos para cada vocal y se vuelven a generar las muestras de las vocales */

/* 5. Con las nuevas muestras se repiten los pasos 1. y 2. */

/* 6. Se comparan graficamente las señales de igual pitch y con distintos pitch */
```

Nuevamente, las señales concatenadas generadas con ambas configuraciones fueron almacenadas en formato *.wav* para analizarlas de forma auditiva, estos audios serán entregados con el *.zip*.

A fin de realizar un análisis correspondiente, se observaron diferencias notables en el timbre y la percepción de los fonemas debido a los cambios en el pitch (y la elección de distintos valores para cada uno). Esto se puede ver en las siguientes Figs.

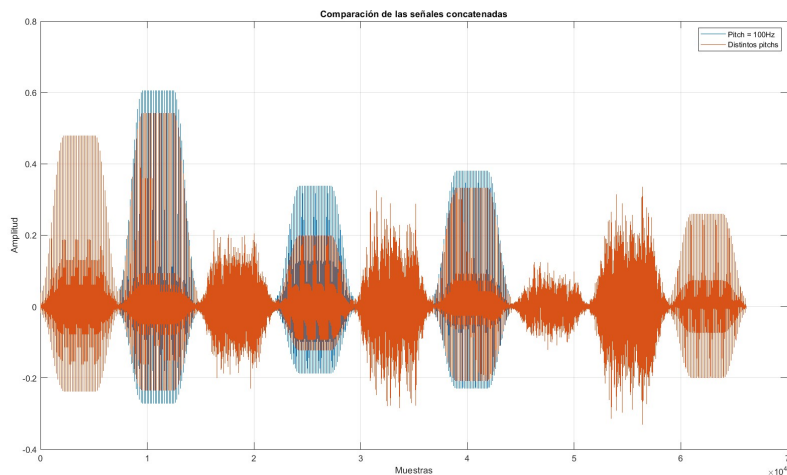


Figura 25: Señales concatenadas con diferentes valores de pitch.

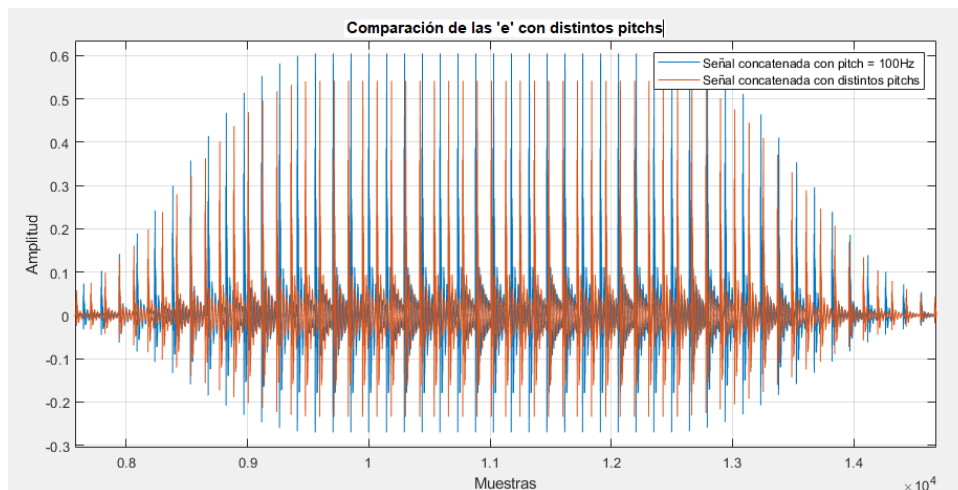


Figura 26: Comparación de las 'e' con distintos pitches.

En base a lo observado en las anteriores Figs. y en función de lo escuchado en los audios almacenados, se puede concluir que se han encontrado algunas diferencias al variar el pitch en cada vocal.

A partir del pitch uniforme, la secuencia concatenada generó una transición fluida entre los fonemas, aunque no se notaron diferencias relevantes entre las vocales debido a la misma frecuencia de pitch. Mientras que, al variar el pitch de las vocales, se observaron diferencias más claras entre ellas, lo que mejoró la percepción de los sonidos. Aún así, algunos fonemas no terminan de distinguirse claramente (como es el caso de la u).

Por último, cabe destacar que el suavizado de bordes con un `fade = 30` de resultó efectivo para evitar cortes bruscos en las transiciones entre fonemas.

3. Conclusiones

Este trabajo permitió analizar diferentes aspectos de la síntesis de fonemas mediante procesos autorregresivos (AR) y la técnica LPC, abordando tanto su base teórica como su implementación práctica. En primer lugar, se derivaron expresiones matemáticas clave, como las ecuaciones de autocorrelación y densidad espectral de potencia de un proceso AR de orden P, lo que permitió comprender cómo los coeficientes y la ganancia se relacionan. Este análisis fue esencial para consolidar el marco teórico del modelo.

Luego, se estimaron los parámetros LPC de distintos fonemas y se analizaron sus periodogramas en comparación con las PSD teóricas. Los resultados mostraron cómo las vocales tienen picos espectrales bien definidos, reflejando su periodicidad, mientras que las consonantes presentan espectros más planos, similares al ruido blanco. Además, con el método de Welch, se evaluaron los efectos del tamaño de la ventana en la estimación de la PSD, destacando la importancia de equilibrar resolución y suavidad dependiendo del análisis.

Por último, se sintetizaron los fonemas utilizando los parámetros estimados y se generaron secuencias concatenadas suavizando las transiciones. Al variar las frecuencias de pitch en las vocales, se observó una mejora, diferenciando mejor los sonidos. Sin embargo, en algunos casos, como la 'u', se presentaron ciertas dificultades para su distinción. Como dato aparte, los resultados de esta última actividad se pueden ver reproduciendo los audios que se encuentran en la carpeta *Audios_sintetizados*.

En conclusión, cabe destacar que este trabajo práctico ayudó al desarrollo de un pseudocódigo, previo a comenzar a generar el *script* en *Matlab*. De esta manera, se pudieron implementar estimaciones y gráficos basados en modelos teóricos y empíricos, lo que demuestra aplicaciones prácticas y contribuye a una mejor comprensión de los conceptos aprendidos en clase.