



Procesos Estocásticos (86.09)

Trabajo Práctico Integrador

Linear predictive coding

2°C 2025

Integrantes		
Nombre y apellido	Padrón	Correo
Emilia Cavalitto	109394	ecavalitto@fi.uba.ar
Ulises Ferrero	105034	uferrero@fi.uba.ar
Francisco Javier Moya	109899	fjmoya@fi.uba.ar

Índice

Índice	1
1. Introducción	2
2. Desarrollo	2
2.1. Ejercicio 1	2
2.1.a. Implementación de la función con prototipo <code>param_lpc(xs, P)</code>	2
2.1.b. Respuesta temporal y periodograma vs. densidad espectral de potencia (PSD)	4
2.2. Ejercicio 2	10
2.2.a. Implementación de la función <code>pitch_lpc</code>	10
2.2.b. Sintetización de fonemas mediante coeficiente LPC	11
2.3. Ejercicio 3: Codificación y reconstrucción de voz LPC	15
2.3.a. Implementación del modelo base y selección de parámetros	15
2.3.b. Justificación de la selección de parámetros	15
2.3.c. Rango de detección del pitch (f_p)	15
2.3.d. Análisis comparativo de los modos de excitación (3a, 3b, 3c, 3d)	15
3. Anexo: Justificación del estimador de autocorrelación	18
3.1. Estimación sesgada vs. no sesgada en el modelo LPC	18
3.2. El problema de la inestabilidad numérica	18

1. Introducción

En este informe se implementaron y analizaron técnicas de procesamiento de señales del habla orientadas al método Linear Predictive Coding (LPC). Para ello, se desarrollaron códigos en Python que permitieron representar la señal de voz mediante un conjunto reducido de parámetros, capturando sus características acústicas esenciales y posibilitando así su estudio y comparación.

2. Desarrollo

2.1. Ejercicio 1

2.1.a. Implementación de la función con prototipo `param_lpc(xs, P)`

En este ejercicio, se tuvo por objetivo implementar una función con prototipo `param_lpc(xs, P)` cuyos parámetros xs y P correspondieron a la señal y al orden del modelo, respectivamente. La misma devolvió los coeficientes LPC y la ganancia G para los distintos audios proporcionados por la cátedra para su análisis.

Para ello, se tomaron los audios de los fonemas provistos para analizar (“`a.wav`”, “`e.wav`”, “`s.wav`” y “`sh.wav`”) se los normalizó en amplitud y se los convirtió a mono en caso de que se encontrasen en estéreo. Luego, se procedió a calcular la matriz de autocorrelación no sesgada¹ para cada señal previamente normalizada, mediante la ecuación (1).

$$\hat{R}_x(k) = \frac{1}{N-k} \sum_{n=o}^{N-|k|-1} x(n)x(n+k) \quad (1)$$

Posteriormente, se armó la matriz de autocorrelación R^2 y se procedió a resolver la ecuación de Yule-Walker (2), donde $a = [a_1, a_2, \dots, a_p]^T$ corresponden a los coeficientes al modelo LPC de las señales de audio analizadas y $r = [R_X(1), R_X(2), \dots, R_X(P)]^T$.

$$R \cdot a = r \quad (2)$$

Por otro lado, la ganancia se determinó mediante la ecuación (3).

$$G = \left(R_X(0) - \sum_{i=1}^P a_i R_x(i) \right)^{\frac{1}{2}} \quad (3)$$

Cada uno de los cálculos se realizaron suponiendo órdenes de $P = \{5, 10, 30\}$.

A continuación, en “Estimación de parámetros LPC”, se presentan los coeficientes LPC y las ganancias obtenidas para cada uno de los fonemas analizados bajo los diferentes valores de P .

Estimación de parámetros LPC

`a.wav`

```
P= 5 | G=0.116239 | a[0:5]=[ 1.15261234 -0.64550264  0.55626919 -0.1833382  -0.24576042 ]
P=10 | G=0.100222 | a[0:10]=[ 1.32202219 -0.76677186  0.40639055  0.27117203 -0.91512263
                  0.69802044 -0.36429236 -0.01112669  0.20860729 -0.01513923 ]
P=30 | G=0.086179 | a[0:30]=[ 1.17502612 -0.55202568  0.38302157  0.30234299 -1.0146622
                  0.5621878 -0.22559462 -0.07750653  0.53726574 -0.05449198
                  -0.17823281  0.03640665 -0.44273197  0.30194123  0.29928481
                  -0.05816929  0.08243032 -0.2343359 -0.17460515  0.1039561
                  0.02504087  0.03878478  0.05274612 -0.09568177 -0.09323858
                  0.0395693   0.01263402  0.02807319  0.07717911 -0.10711206 ]
```

¹Explicación del uso de la autocorrelación no sesgada en el Anexo 3

²La matriz R se construyó con estructura de *Toeplitz*, dado que cada una de sus diagonales contiene valores constantes derivados de la autocorrelación, en concordancia con las ecuaciones de Yule-Walker.

e.wav

```
P= 5 | G=0.141801 | a[0:5]=[ 0.92871435 -0.91513711  1.04631618 -0.45416427  0.24058898 ]
P=10 | G=0.099021 | a[0:10]=[ 0.63526429 -0.85808074  1.179791   -0.11166585  0.39163688
          0.13070296 -0.62079814  0.04647465 -0.44992192  0.18835150 ]
P=30 | G=0.082407 | a[0:30]=[ 3.35212225e-01 -8.39539566e-01  1.06487912e+00  3.51125653e-01
          7.91601600e-01  4.75471943e-01 -5.46249193e-01 -4.54068618e-01
          -8.96148688e-01 -3.32999477e-01 -8.24421260e-03  3.13452282e-01
          3.39603160e-01  3.46402710e-01 -1.31883282e-02 -1.07414257e-01
          -9.93301574e-02 -2.03792184e-01  9.15702673e-02 -5.69457825e-04
          1.49692010e-01  1.42675592e-01 -1.50899235e-02 -2.26309637e-02
          -1.41484484e-01 -1.60775470e-01 -3.97406079e-02 -4.93769543e-02
          4.82507758e-02  4.81324227e-02 ]
```

s.wav

```
P= 5 | G=0.204110 | a[0:5]=[-0.18501515  0.31966370  0.09450498  0.04139111  0.36357809 ]
P=10 | G=0.189926 | a[0:10]=[-0.31588135  0.15072559 -0.03264615 -0.02861978  0.25878718
          0.08869586  0.27520406  0.25378228  0.07744548  0.11067828 ]
P=30 | G=0.170534 | a[0:30]=[-4.17233968e-01  9.51396155e-02 -3.22622101e-02 -1.16432501e-01
          1.45642750e-01  4.69354507e-02  2.59482901e-01  2.75414882e-01
          9.39365221e-02  1.10793199e-01  2.48072253e-01  1.82593475e-01
          2.80306957e-03 -8.08153486e-02  1.18691238e-01  1.21940649e-01
          1.70938088e-01  3.24882343e-02  7.20379501e-02  7.29420583e-02
          2.69118103e-02 -1.90611830e-03 -2.15437869e-04 -1.00066027e-01
          -2.19935047e-01 -2.05957090e-01 -9.16426976e-02  1.43304366e-02
          6.05292150e-02 -2.88119003e-02 ]
```

sh.wav

```
P= 5 | G=0.102113 | a[0:5]=[ 0.96060145 -1.26958536  0.51482359 -0.56787666  0.01858760 ]
P=10 | G=0.093750 | a[0:10]=[ 1.02956915 -1.57775053  0.92810782 -1.25528822  0.72425146
          -0.67471872  0.35053885 -0.15949236  0.07958799  0.01913870 ]
P=30 | G=0.082522 | a[0:30]=[ 1.04688894 -1.71933946  1.14831796 -1.68371750  1.26163157
          -1.41159352  1.28700880 -1.12118317  1.30609707 -1.07086917
          1.25115776 -0.85128975  0.98420843 -0.90262514  1.04146428
          -0.98311207  0.98856512 -0.88481651  0.81419965 -0.67750333
          0.70997816 -0.45275705  0.48883578 -0.30853563  0.31146347
          -0.12766882  0.10356937 -0.07386883  0.01656008 -0.03636515 ]
```

Como se puede observar en los resultados obtenidos, los valores de la ganancia G disminuyeron sistemáticamente al incrementar el orden de P , lo cual resulta consistente con el método estudiado: al aumentar el número de coeficientes, el modelo LPC logró capturar una mayor proporción de la energía de la señal, disminuyendo el error de predicción y así la energía del error de predicción (G^2).

Con respecto a los coeficientes LPC, como era de esperar, las vocales presentaron valores de mayor magnitud y una estructura más marcada, reflejando la presencia de formantes y la naturaleza periódica de las mismas. Por lo contrario, las consonantes fricativas exhibieron coeficientes más irregulares y variables, lo cual resultó coherente con su espectro ruidoso y poco estructurado.

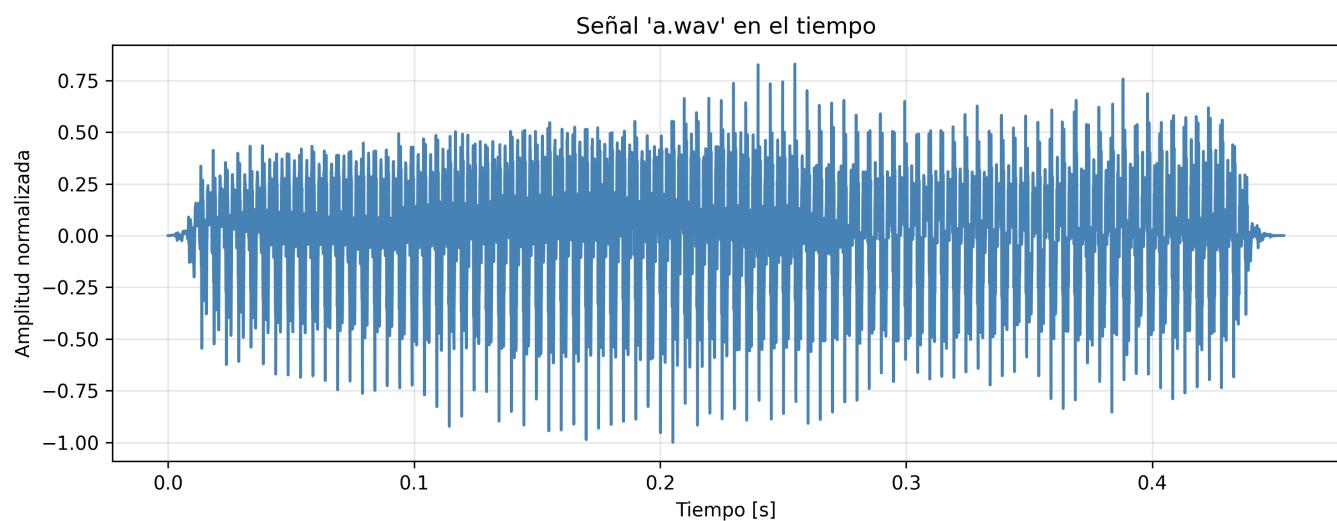
2.1.b. Respuesta temporal y periodograma vs. densidad espectral de potencia (PSD)

En este apartado se buscó obtener gráficamente las respuestas temporales de los audios de la sección 2.1.a, y los periodogramas de cada uno superpuesto a la PSD paramétrica estimada correspondiente.

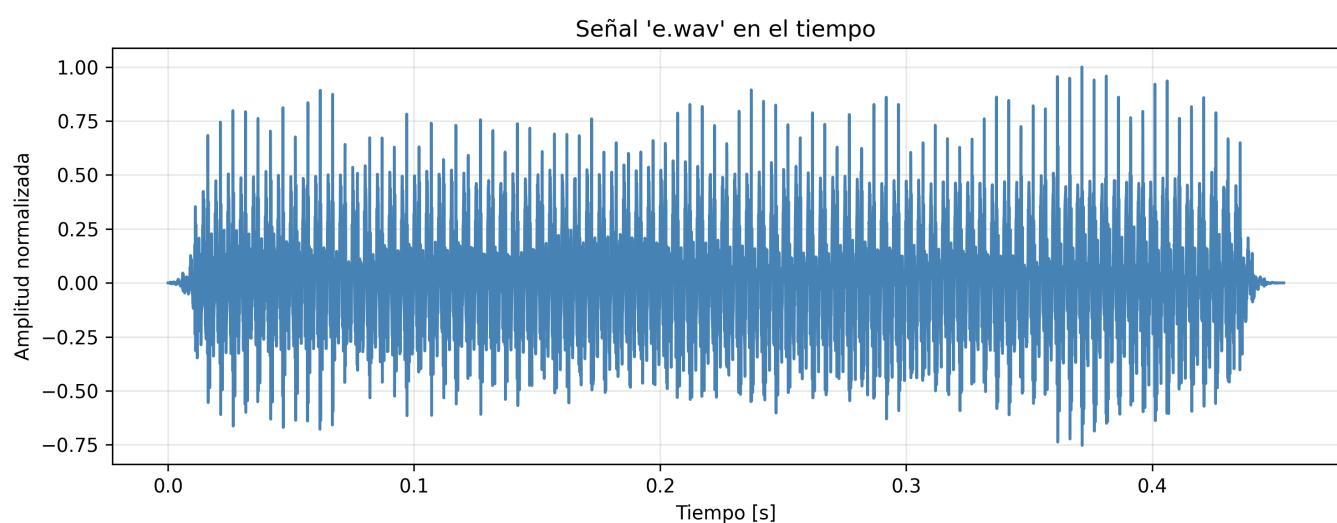
Para ello, se graficaron las señales previamente normalizadas en el dominio temporal (figuras 1 y 2), utilizando su frecuencia de muestreo original para relacionar cada muestra con su instante correspondiente. Luego, se procedió a estimar el periodograma de cada audio y se calculó la PSD paramétrica asociada al modelo LPC según la ecuación (4)³ (para el rango $\omega \in [0, \pi]$).

$$S_X(\omega) = \frac{G^2}{|1 - \sum_{k=1}^P a_k e^{-j\omega k}|^2} S_U(\omega) \quad (4)$$

Posteriormente, se prosiguió a calcular el periodograma de cada fonema mediante la Transformada Rápida de Fourier (FFT) que, junto con los valores determinados por el cálculo de la PSD de las mismas, fueron expresados en decibeles (dB) de modo tal de facilitar su comparación. Finalmente, se graficaron ambas curvas superpuestas (como se observa en las figuras 3, 4, 5 y 6).



(a) Fonema “a”.



(b) Fonema “e”.

Figura 1: Respuestas temporales de los fonemas vocálicos.

³En este caso, la PSD de la señal de excitación, $S_U(\omega)$, se tomó igual a uno como una aproximación simplificada, lo cual permitió comparar de manera directa la forma espectral del modelo LPC con el periodograma obtenido.

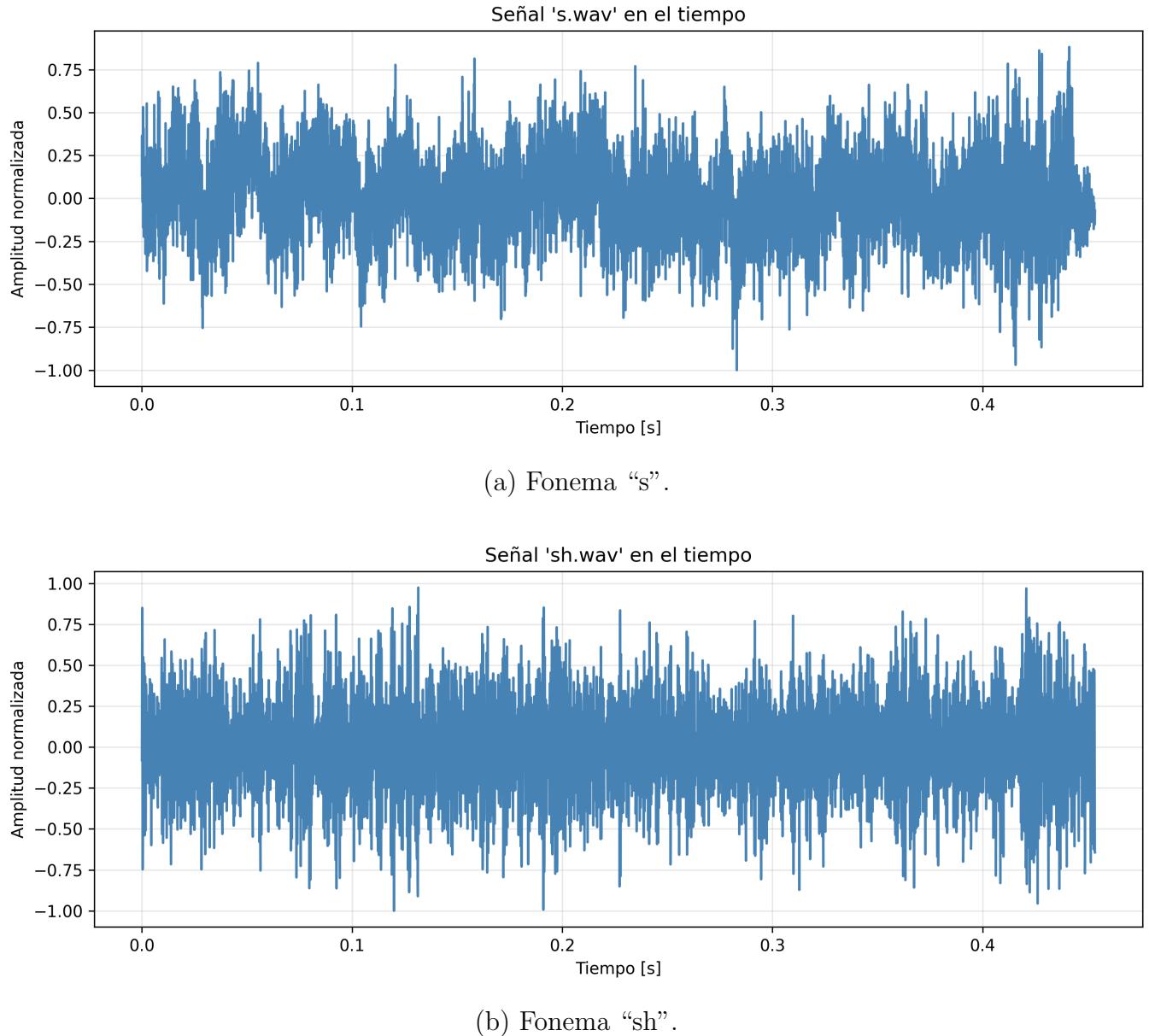


Figura 2: Respuestas temporales de los fonemas fricativos.

A partir de los gráficos de las respuestas temporales de los fonemas vocálicos se observó que las señales correspondientes a los audios “a.wav” y “e.wav” presentaron una estructura aproximadamente periódica y variaciones relativamente suaves en la envolvente de amplitud. Este comportamiento resultó consistente con la naturaleza sonora de estos fonemas, donde la excitación puede modelarse como un tren de impulsos periódico que, al pasar por el filtro del tracto vocal, genera una señal con periodicidad bien definida. En contraste, las respuestas temporales de los fonemas fricativos “s.wav” y “sh.wav” mostraron un comportamiento mucho más ruidoso y sin un patrón periódico evidente. Esto coincidió con el modelo de habla sorda, donde la excitación se approximó mediante ruido blanco gaussiano.

Por otro lado, en las figuras de comparación entre el periodograma y la PSD paramétrica obtenida mediante LPC para cada fonema, se observó que, para los fonemas vocálicos, la PSD capturó los picos principales asociados a los formantes, suavizando las oscilaciones de alta frecuencia presentes en el periodograma. En el caso de los sonidos de consonantes, cuyos espectros presentaron un contenido más amplio en frecuencia y menor definición de los picos, la PSD paramétrica describió de manera razonable la tendencia general del espectro aunque sin seguir todos los detalles finos del periodograma. Esto resultó coherente con la naturaleza más ruidosa de estas señales y con las limitaciones esperables de un modelo sólo de polos para describir procesos de tipo ruido. En ambos casos se registró que al aumentar el valor de P sucesivamente, la envolvente estimada se ajustó progresivamente mejor a la forma general del espectro.

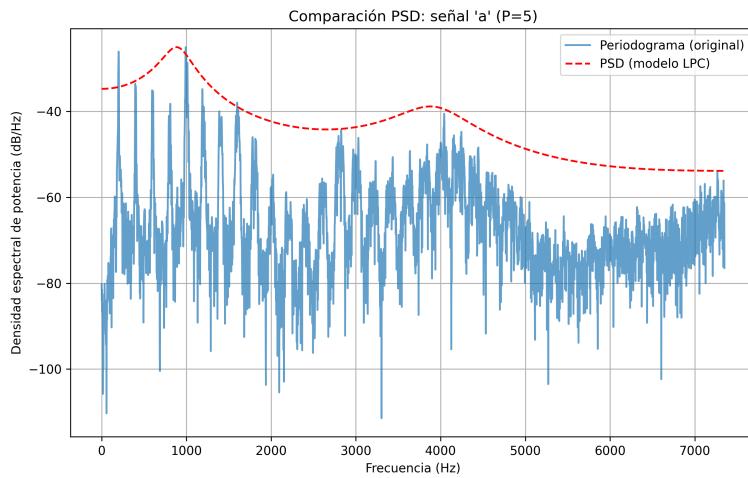
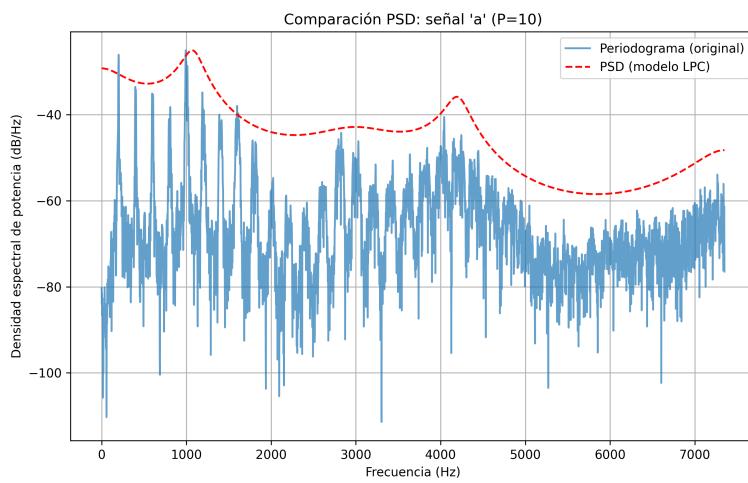
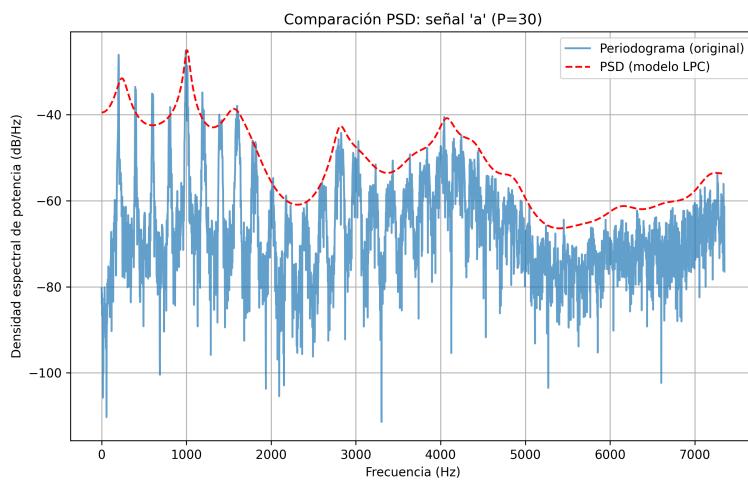
(a) Fonema “a” — Orden $P = 5$.(b) Fonema “a” — Orden $P = 10$.(c) Fonema “a” — Orden $P = 30$.

Figura 3: Comparación entre periodograma y PSD LPC para el fonema “a”.

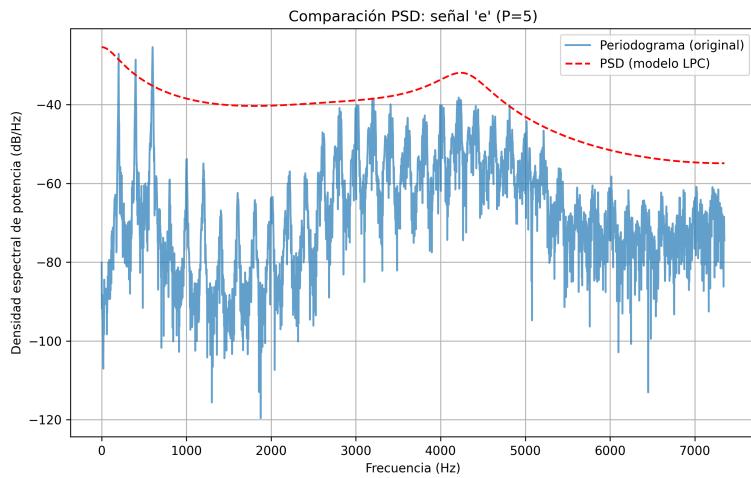
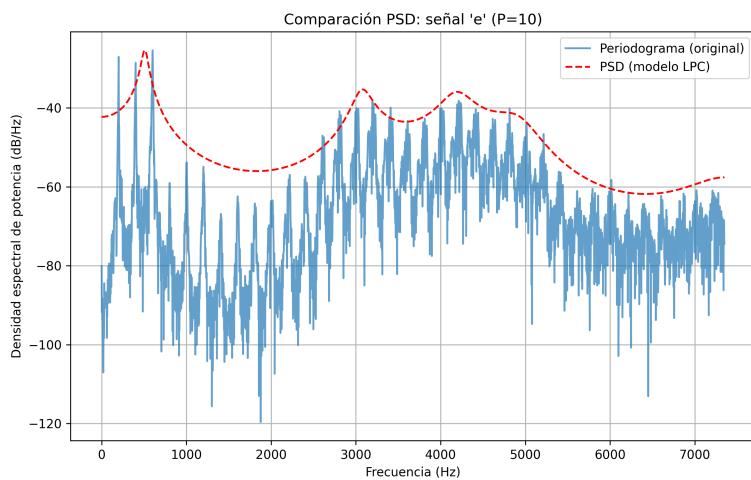
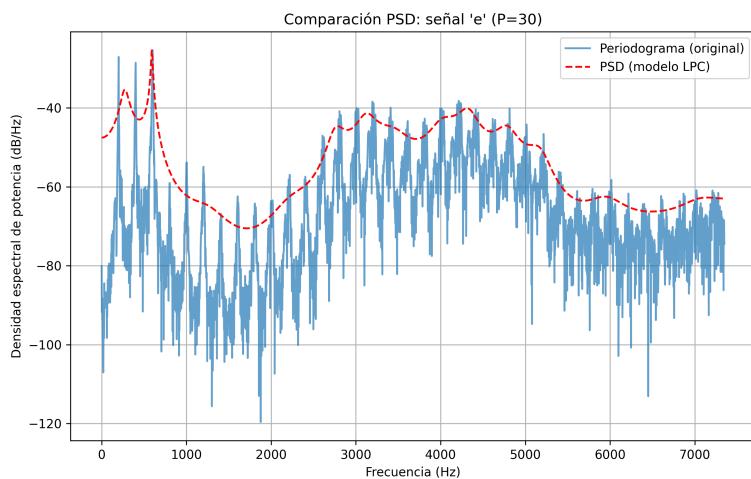
(a) Fonema “e” — Orden $P = 5$.(b) Fonema “e” — Orden $P = 10$.(c) Fonema “e” — Orden $P = 30$.

Figura 4: Comparación entre periodograma y PSD LPC para el fonema “e”.

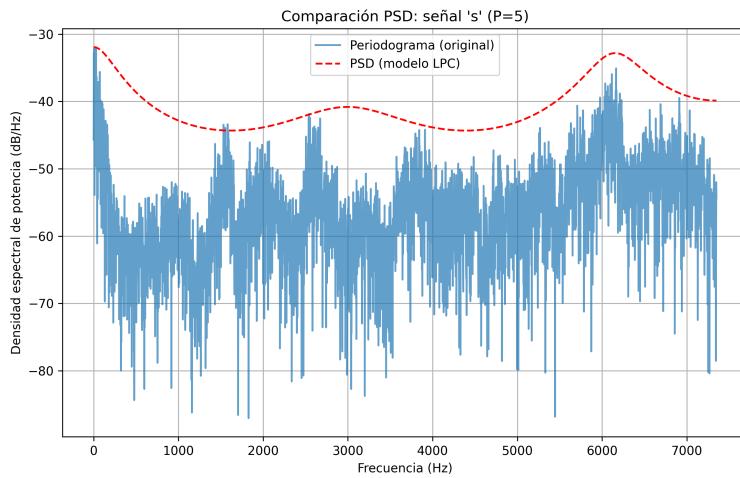
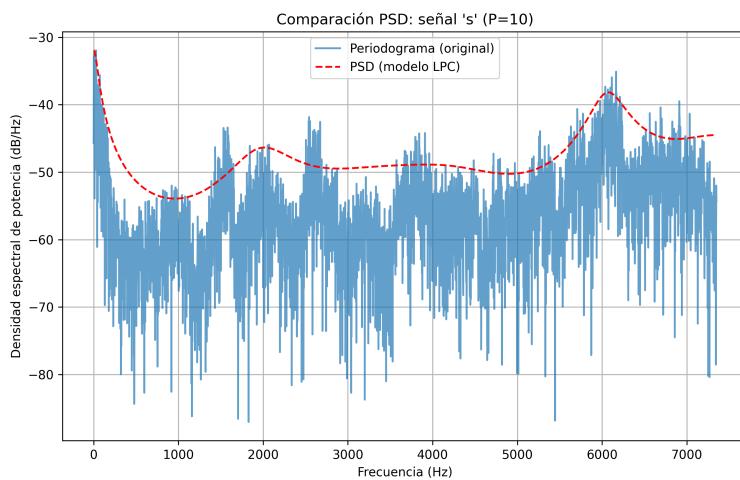
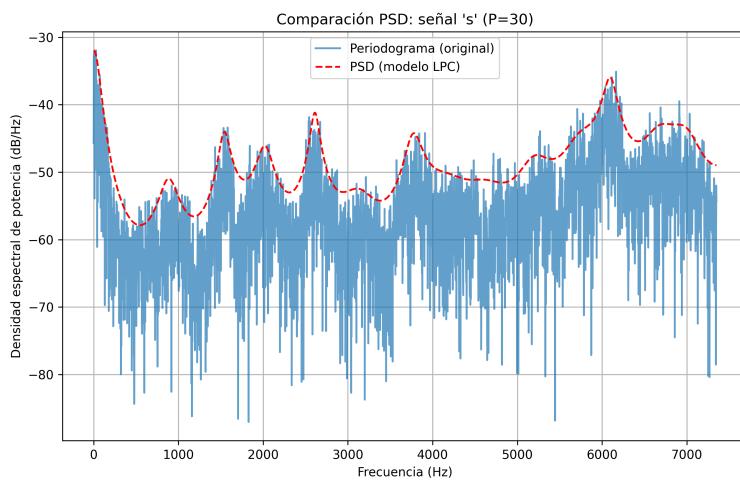
(a) Fonema “s” — Orden $P = 5$.(b) Fonema “s” — Orden $P = 10$.(c) Fonema “s” — Orden $P = 30$.

Figura 5: Comparación entre periodograma y PSD LPC para el fonema “s”.

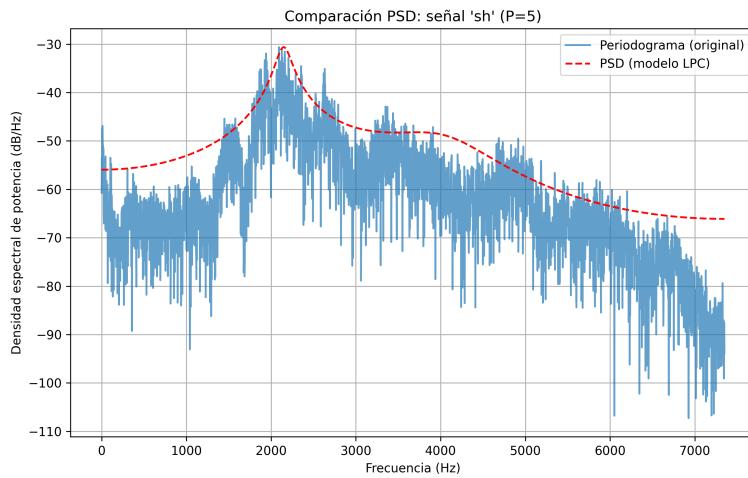
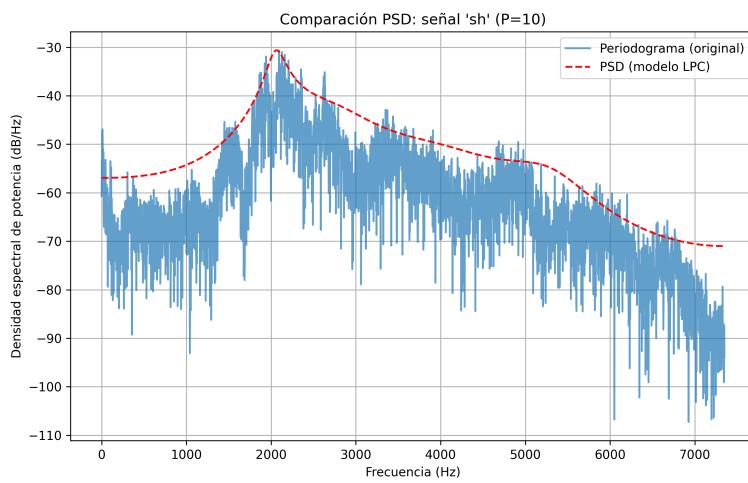
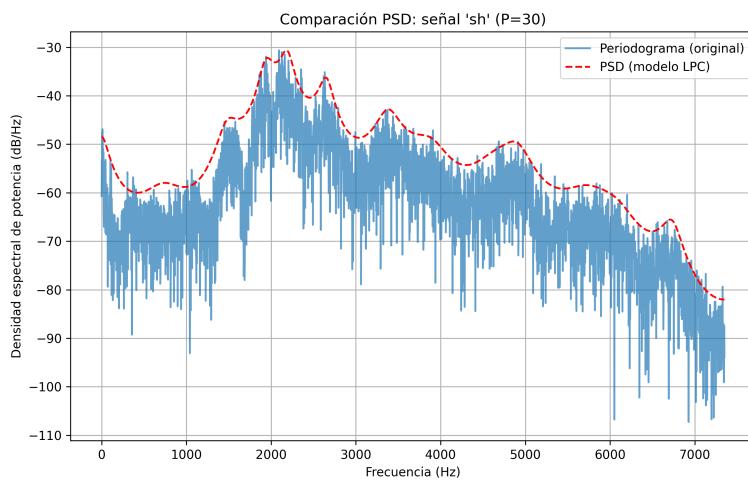
(a) Fonema “sh” — Orden $P = 5$.(b) Fonema “sh” — Orden $P = 10$.(c) Fonema “sh” — Orden $P = 30$.

Figura 6: Comparación entre periodograma y PSD LPC para el fonema “sh”.

2.2. Ejercicio 2

2.2.a. Implementación de la función pitch_lpc

El propósito de este ejercicio fue implementar la función `pitch_lpc(xs, a, alpha, fs)`⁴. La función aplica el método de autocorrelación para identificar la periodicidad presente en el segmento. Si se detecta un máximo significativo por encima del umbral `alpha`, se calcula la frecuencia fundamental asociada; en caso contrario (propio de fonemas fricativos) la función retorna un valor de pitch igual a cero.

Se comenzó de igual manera que en la sección 2.1: inicialmente se normalizaron y transformaron las señales a mono y luego se estimaron los parámetros LPC para un P determinado (en este caso, $P = 10$, valor dentro del rango típico para la voz humana)⁵.

Para determinar la frecuencia de pitch f_p , se calculó la autocorrelación del segmento de la señal y se normalizó respecto de su valor en el retardo cero. A continuación, la búsqueda del periodo fundamental se restringió al rango típico de la voz humana (50 Hz a 500 Hz) y, dentro de dicho intervalo, se identificó el máximo de la autocorrelación (sin incluir el pico en el retardo nulo); el cual se interpretó como el periodo fundamental candidato y , por ende, como la mejor opción para f_p .

La decisión para clasificar a las señales en sonoras/sordas se tomó comparando la amplitud del máximo de la autocorrelación normalizada con el umbral `alpha`⁶. En este caso, se evaluó para $\text{alpha} = \{0,3, 0,5, 0,7\}$ ⁷.

Finalmente, cuando la amplitud del máximo superó al umbral, se calculó la frecuencia de pitch mediante la ecuación 5,

$$f_p = \frac{f_s}{k_{max}} \quad (5)$$

(k_{\max} , índice de retardo donde se alcanzó dicho máximo en el rango de búsqueda); y en caso contrario, la función devolvió $f_p = 0$.

A continuación, se muestran los resultados obtenidos considerando un orden $P = 10$ y una frecuencia de muestreo $f_s = 14700$ Hz:

a.wav

```
Orden P=10 (G=8.204475):
alpha=0.3 | Pitch: 201.37 Hz (SONORO)
alpha=0.5 | Pitch: 201.37 Hz (SONORO)
alpha=0.7 | Pitch: 201.37 Hz (SONORO)
```

e.wav

```
Orden P=10 (G=8.118968):
alpha=0.3 | Pitch: 201.37 Hz (SONORO)
alpha=0.5 | Pitch: 201.37 Hz (SONORO)
alpha=0.7 | Pitch: 201.37 Hz (SONORO)
```

s.wav

```
Orden P=10 (G=15.516826):
alpha=0.3 | Sin pitch detectado (SORDO)
alpha=0.5 | Sin pitch detectado (SORDO)
alpha=0.7 | Sin pitch detectado (SORDO)
```

sh.wav

```
Orden P=10 (G=7.692935):
alpha=0.3 | Sin pitch detectado (SORDO)
alpha=0.5 | Sin pitch detectado (SORDO)
alpha=0.7 | Sin pitch detectado (SORDO)
```

⁴En este contexto: `xs` representa un fragmento de una señal aproximadamente estacionaria; `a` corresponde al vector de coeficiente LPC; `alpha` al umbral de correlación normalizada que determina si un segmento es sonoro o sordo; y `fs` a la frecuencia de muestreo.

⁵Los coeficientes LPC se calcularon siguiendo exactamente el procedimiento explicado en el ejercicio 1a, ya que la función a implementar en este ejercicio los recibe como argumento.

⁶Valores de `alpha` bajos hacen el criterio más permisivo (aumentan los falsos positivos de sonoridad), mientras que valores altos “endurecen” la decisión (aumentan los falsos negativos).

⁷Típicamente se seleccionan valores de umbral en el rango de 0.3 a 0.7 porque ese intervalo separa de forma fiable segmentos con periodicidad clara (vocales) de segmentos ruidosos. La elección concreta depende del balance deseado entre sensibilidad y especificidad en la detección sonora/sorda.

Como se observa en los resultados, los fonemas “a” y “e” fueron correctamente identificados como sonoros en los tres valores de `alpha` evaluados, con una estimación consistente de $f_p \approx 201,4$ Hz. La coincidencia del pitch para `alpha = {0,3, 0,5, 0,7}` sugiere que la periodicidad de estos segmentos es pronunciada y robusta frente al umbral de decisión. Por contraste, los fonemas fricativos “s” y “sh” no presentaron picos secundarios relevantes en la autocorrelación dentro del rango de búsqueda, por lo que fueron clasificados como sordos en todos los casos. Además, la ganancia G resultó notablemente mayor en `s.wav`, consistente con la mayor energía dispersa característica.

2.2.b. Sintetización de fonemas mediante coeficiente LPC

En este apartado, se tuvo por objetivo sintetizar una vocal y una consonante empleando sus respectivos coeficientes LPC seleccionados con un orden P apropiado (en este caso, $P = \{10, 30, 50\}$ para evaluar los resultados ante diferentes órdenes del predictor), con el fin de comparar sus características espectrales a partir de los resultados obtenidos.

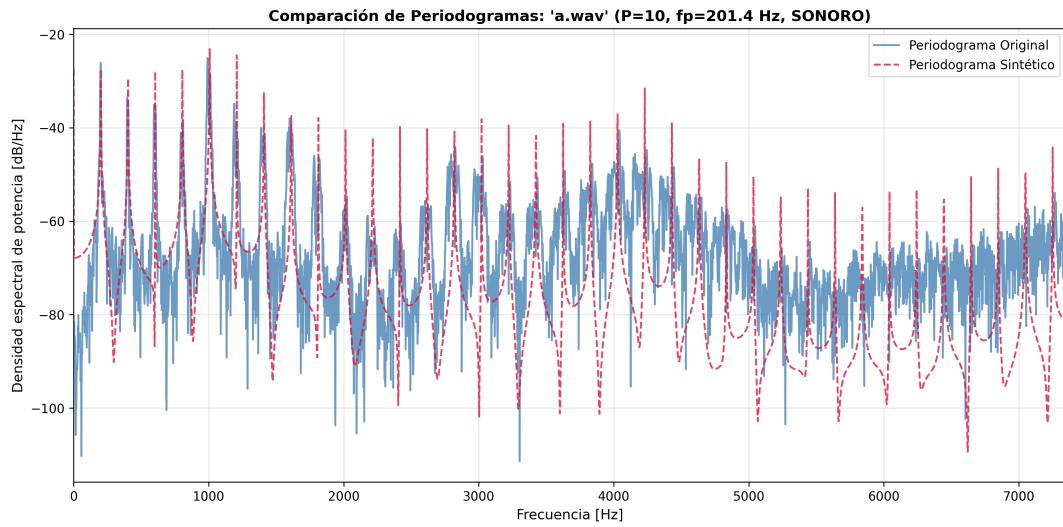
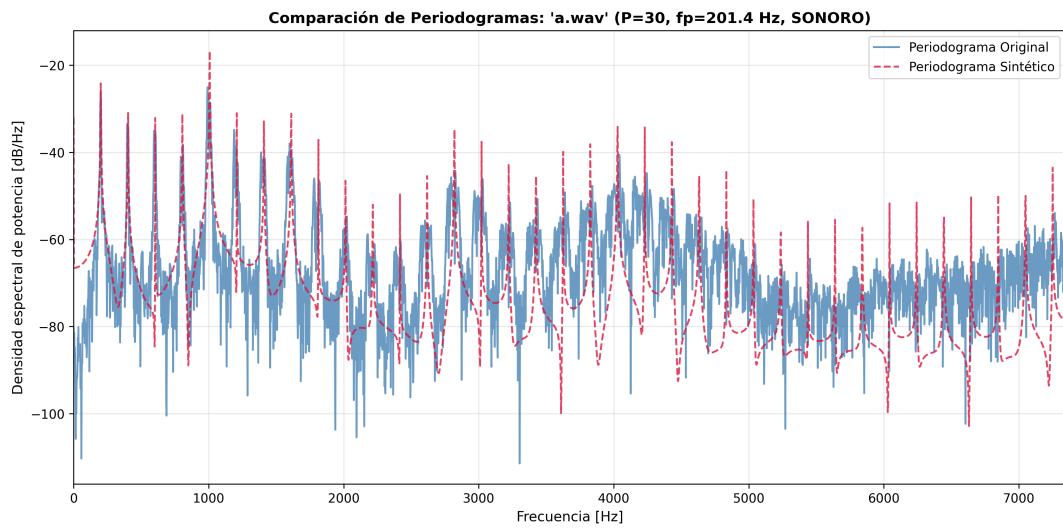
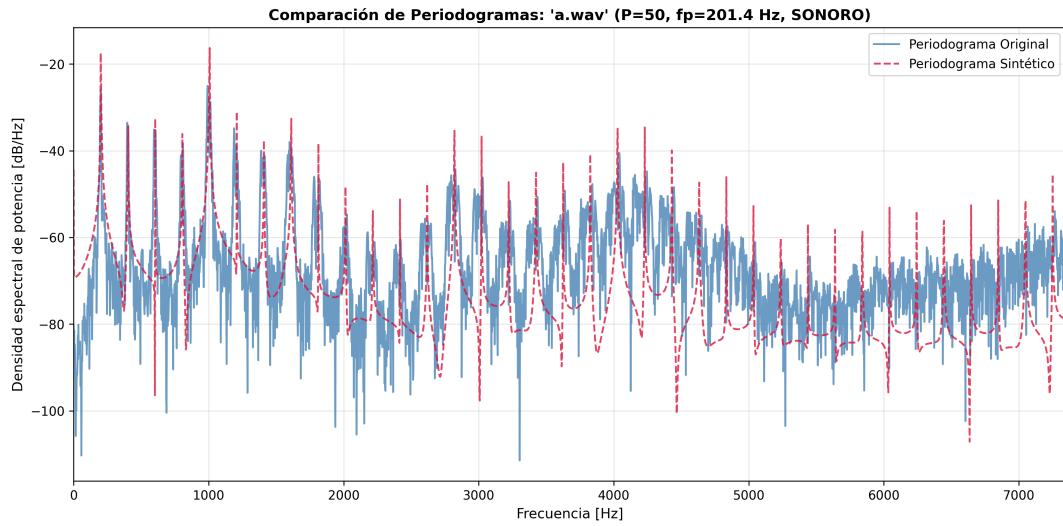
Para la vocal, se utilizó la función `pitch_lpc` (con `alpha = 0,3`) previamente implementada en la sección 2.2.a, con el propósito de estimar su frecuencia de pitch y generar una excitación periódica coherente con su naturaleza sonora. Por otro lado, para la consonante, se asumió una excitación de tipo sorda.

Se pusieron bajo estudio la vocal “a” y la consonante “s” y, para la generación del fonema sintético donde se empleó la ecuación recursiva (6), se utilizó una señal de excitación $u(n)$, donde, en el primer caso correspondió a un tren de impulsos, mientras que, en el segundo, a un ruido blanco gaussiano⁸.

$$x_{sintetico} = G \cdot u(n) + \sum_{k=1}^P a_k x_{sintetico}(n - k) \quad (6)$$

Finalmente, para cada fonema seleccionado se sintetizó una señal con igual duración que el audio original y se compararon ambas mediante la superposición de sus periodogramas (figuras 7 y 8), a fin de evaluar la fidelidadpectral de la síntesis en cada caso.

⁸En el modelo fuente-filtro del habla, los sonidos sonoros se producen por la vibración periódica de las cuerdas vocales, lo que se modela mediante un tren de impulsos con período igual a la f_p estimada. En contraste, las consonantes fricativas son sonidos sordos: no presentan vibración laríngea y su fuente de excitación es turbulenta. Por ello se modelan mediante ruido blanco gaussiano, que reproduce la naturaleza aperiódica y ruidosa de estos fonemas.

(a) Fonema “a” sintético — Orden $P = 10$.(b) Fonema “a” sintético — Orden $P = 30$.(c) Fonema “a” sintético — Orden $P = 50$.Figura 7: Comparación de espectrogramas para vocal “a” sintética para distintos órdenes de P .

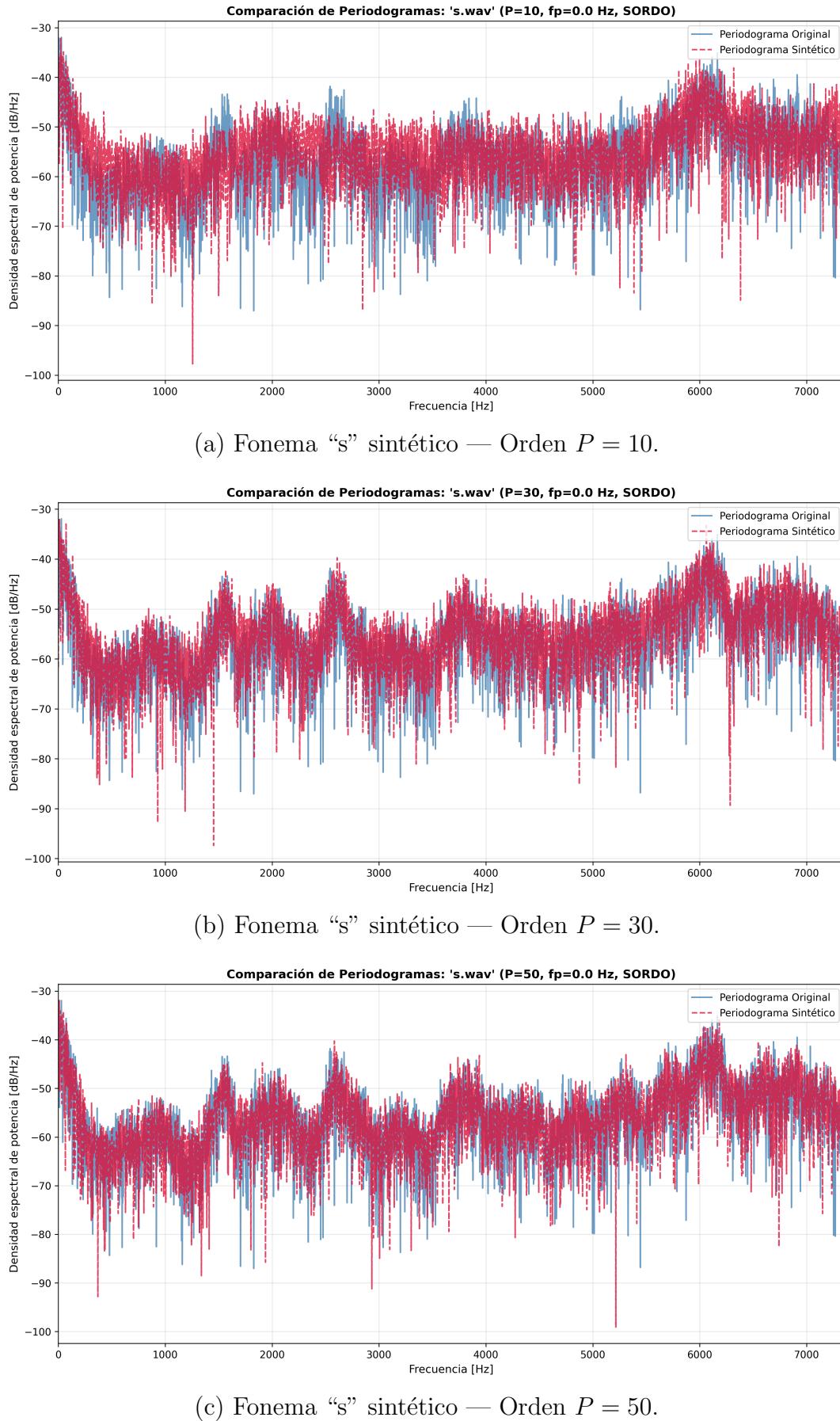


Figura 8: Comparación de espectrogramas para vocal “e” sintética para distintos órdenes de P .

Procesando: s.wav

- Frecuencia de muestreo: 14700 Hz
- Duración: 0.454 s (6667 muestras)

>> Analizando con P = 10

- Ganancia G = 8.204475
- Pitch detectado: 201.37 Hz → SONORO
- Señal sintetizada

>> Analizando con P = 30

- Ganancia G = 7.074920
- Pitch detectado: 201.37 Hz → SONORO
- Señal sintetizada

>> Analizando con P = 50

- Ganancia G = 6.678642
- Pitch detectado: 201.37 Hz → SONORO
- Señal sintetizada

Procesando: s.wav

- Frecuencia de muestreo: 14700 Hz
- Duración: 0.454 s (6667 muestras)

>> Analizando con P = 10

- Ganancia G = 15.516826
- Sin pitch detectado → SORDO
- Señal sintetizada

>> Analizando con P = 30

- Ganancia G = 13.949055
- Sin pitch detectado → SORDO
- Señal sintetizada

>> Analizando con P = 50

- Ganancia G = 13.720407
- Sin pitch detectado → SORDO
- Señal sintetizada

Como se observa en los resultados, para la sintetización del fonema “a” la frecuencia de pitch se detectó correctamente. En el caso de la consonante “s” sintética se determinó nuevamente, mediante la función `pitch_lpc`, que la señal corresponde a un sonido sordo.

En ambos casos, se puede apreciar que, partiendo de $P = 10$ se llega a una estructura general en los formantes de la señal original. A medida que se incrementa el orden del predictor, la definición aumenta y la señal se aproxima a la original. Esto ocurre porque un mayor número de coeficientes permite modelar más resonancias (dado que la envolvente espectral se vuelve más precisa) y ajustar con más detalle la forma espectral del tracto vocal. No obstante, el aumento indefinido de P no resulta siempre ventajoso, ya que esto conduce a un sobreajuste, captando variaciones no correspondientes a la señal (como por ejemplo, ruido), volviendo al modelo más inestable.

2.3. Ejercicio 3: Codificación y reconstrucción de voz LPC

El objetivo central de este ejercicio fue implementar el sistema completo de codificación y decodificación por Predicción Lineal Coding (LPC), que es una aplicación directa del modelo fuente-filtro. El análisis se centró en observar cómo la calidad de la señal sintetizada está determinada por la precisión del filtro (coeficientes **a**) y la naturaleza de la fuente excitadora (pitch o ruido).

2.3.a. Implementación del modelo base y selección de parámetros

La codificación y decodificación LPC opera en segmentos de audio, donde se extraen los coeficientes LPC (**a** y **G**) y la fuente (f_p). El objetivo es lograr el balance óptimo entre la inteligibilidad y la compresión.

2.3.b. Justificación de la selección de parámetros

La selección final de los parámetros se basó en una optimización perceptiva (mediante escucha) que buscó mitigar los artefactos robóticos inherentes al modelo. Los valores utilizados fueron: $f_s = 8000 \text{ Hz}$, $L = 180$ muestras ($\approx 22,5 \text{ ms}$), $P = 15$, $\alpha = 0,2$, y $hop = L/2$.

- **Largo del segmento ($L = 180$ muestras):** Se eligió un segmento corto (22,5 ms) como compromiso entre la estabilidad y la estacionariedad. Minimizar la longitud es crucial, ya que se asume que las características del tracto vocal son constantes dentro de la ventana de análisis.
- **Orden del predictor ($P = 15$):** El orden **P** determina cuántos polos tendrá el filtro. Este valor es ligeramente superior a la regla empírica $P \approx f_s/1000 + 2 \approx 10$ para acomodar formantes adicionales y proveer una mejor resolución espectral⁹. Se evitó un **P** excesivamente alto para prevenir el sobreajuste (overfitting), donde el filtro intentaría modelar los picos individuales del *pitch* en lugar de la envolvente, causando inestabilidad.
- **Umbral ($\alpha = 0,2$):** Se utilizó un valor bajo para maximizar la sensibilidad del detector de periodicidad, asegurando que la mayoría de los segmentos vocálicos fueran clasificados como sonoros y mitigar el susurro en las vocales.
- **Solapamiento (hop):** El solapamiento estándar del 50 % garantizó una transición suave y continua de la energía entre segmentos (*overlap-add*).

2.3.c. Rango de detección del pitch (f_p)

La función de detección de pitch utiliza el Método de Autocorrelación (ACF) y establece su búsqueda en el rango 50 Hz a 500 Hz, que cubre la totalidad del rango fisiológico de la voz humana durante el habla .

- **Límite inferior (50 Hz):** Garantiza la detección de las voces masculinas más graves (cuyo f_p promedio está entre 85 Hz y 155 Hz).
- **Límite superior (500 Hz):** Cubre adecuadamente las frecuencias fundamentales de mujeres y niños. Este límite es crucial para evitar la detección de *pitch* por octava, previniendo que el algoritmo confunda el f_p real con el primer armónico.

Excitación de la fuente: Es importante recordar que el *pitch* (tren de impulsos periódicos) se usa para sonidos sonoros (vocales: /a/, /e/, etc., y consonantes sonoras: /m/, /n/, /z/, /v/) que son producidos por la vibración de las cuerdas vocales. Por otro lado, el *ruido blanco* se emplea para sonidos sordos (consonantes como: /s/, /f/, /t/, /ch/) que se producen por la turbulencia del aire.

2.3.d. Análisis comparativo de los modos de excitación (3a, 3b, 3c, 3d)

La principal conclusión del ejercicio es que la calidad perceptual del habla sintetizada depende críticamente de la precisión de la fuente (f_p y ruido), no solo de la precisión del filtro (**a**).

⁹La regla empírica para **P** se basa en que cada par de polos modela un formante, y los formantes están espaciados aproximadamente cada 1000 Hz en el espectro del habla. Los dos coeficientes adicionales cubren la excitación glotal y la radiación labial.

Modo 3a: Pitch estimado (referencia de calidad)

En este modo, el f_p se mide directamente del segmento de audio original, haciendo que la entonación sea variable y real.

- **Observación perceptual:** El audio es inteligible, pero consistentemente robótico o "zumbante" (buzzy), a pesar de tener una entonación fiel al orador.
- **Conclusión técnica:** Este modo es la referencia de calidad del vocoder LPC. El sonido robótico es la limitación inherente del modelo: la fuente que excita al filtro es un tren de impulsos perfectamente periódico (espectro plano ideal). La voz humana real, en cambio, tiene variaciones sutiles de frecuencia y amplitud (jitter y shimmer) y ruido turbulento, lo que se pierde en esta idealización. La limitación está en la naturaleza simplista de la fuente, no en la estimación de los parámetros del filtro.

Modo 3b: Pitch fijo (200 Hz)

Se utiliza un f_p constante para todos los segmentos sonoros.

- **Observación perceptual:** El habla se vuelve muy robótica y siniestra (monotonía). Las palabras son reconocibles, pero la componente humana desaparece.
- **Conclusión técnica:** Este modo demuestra la importancia de la prosodia (f_p variable). Aunque el filtro ($\mathbf{A}(\mathbf{z})$) modela correctamente la boca (las palabras), la entonación constante (200 Hz) anula la componente emocional y lingüística del habla. El cerebro interpreta esta monotonía forzada como artificial.

Modo 3c: Ruido blanco (todo sordo)

Se destruye toda la periodicidad forzando a la excitación a ser siempre ruido blanco.

- **Observación perceptual:** El resultado es una voz ronca o un susurro siniestro sin tono alguno.
- **Conclusión técnica:** El objetivo es aislar la función del filtro $\mathbf{A}(\mathbf{z})$. Lo que se escucha es el ruido blanco filtrado por el tracto vocal ($\mathbf{A}(\mathbf{z})$). Esto demuestra que los coeficientes LPC son suficientes para modelar las vocales y la identidad del orador. La percepción de voz ronca se debe a que la fuente de ruido simula una fuga de aire constante en las cuerdas vocales. Las vocales, que deberían ser tonales, se pronuncian con la fuente de energía incorrecta.

Modo 3d: Pitch sintético

La entonación se genera artificialmente mediante una función sinusoidal predefinida, desincronizada del habla.

- **Observación perceptual:** La voz suena más robótica y desincronizada que el modo 3b, pero con variación de tono (más neutra).
- **Conclusión técnica:** Demuestra que no solo se necesita variación en el f_p (lo que corrige el modo 3b), sino que esta variación debe ser lingüísticamente correcta y estar sincronizada con la articulación. El patrón sinusoidal artificial causa una disociación entre el ritmo de las palabras y el tono, generando el efecto de un "robot programado".

Conclusiones

En este trabajo se implementaron y analizaron diversas herramientas basadas en modelos lineales, centrándose principalmente en la autocorrelación, el análisis LPC y técnicas de detección y síntesis. A partir de los ejercicios llevados acabo, se verificó que el modelo LPC es una aproximación útil para describir la envolvente espectral de fonemas tanto sonoros como sordos, y que los resultados dependen fuertemente del orden P elegido: con valores bajos se permiten capturar la forma general del espectro, mientras que para órdenes mayores se mejora la definición y la similitud con la señal original.

La función de detección de pitch permitió diferenciar entre fonemas sonoros y sordos, permitiendo estimar la existencia de periodicidad en vocales y desestimando cuando correspondió, como en el caso de las consonantes fricativas. A partir de estas estimaciones, la síntesis mediante el modelo LPC reprodujo señales considerablemente próximas a las reales.

Entre las dificultades encontradas se destacaron la correcta normalización de los audios, el manejo de los límites al calcular autocorrelaciones, y el correcto ajuste de los parámetros.

Finalmente se puede afirmar que los resultados obtenidos confirman la utilidad del análisis LPC como técnica de modelado, codificación y síntesis.

3. Anexo: Justificación del estimador de autocorrelación

Esta sección detalla la decisión técnica de implementar el estimador de la matriz de autocorrelación (\mathbf{R}) mediante la versión **sesgada** en la función `param_lpc`, a pesar de que la estimación no sesgada es teóricamente superior para señales estacionarias. Esta decisión fue crucial para garantizar la estabilidad del *vocoder* completo (Ejercicio 3).

3.1. Estimación sesgada vs. no sesgada en el modelo LPC

La función `param_lpc` utiliza el método de *Yule-Walker* para extraer los parámetros \mathbf{a} y \mathbf{G} , cuya base es la matriz de autocorrelación \mathbf{R} .

- **Estimación No Sesgada ($\hat{R}_x(k)$):** Divide la suma de productos por la longitud de la superposición ($N - |k|$), lo que proporciona una estimación de mínima varianza. Fue la versión inicialmente utilizada, siguiendo la definición rigurosa.
- **Estimación Sesgada ($\hat{R}_x(k)$):** Divide por el largo total de la señal (N), o simplemente utiliza la suma cruda de productos (`np.correlate`), lo cual es numéricamente equivalente para resolver el sistema $\mathbf{R} \cdot \mathbf{a} = \mathbf{r}$.

3.2. El problema de la inestabilidad numérica

La versión **no sesgada** fue descartada debido a problemas de **inestabilidad numérica severa** en el contexto de la síntesis de voz:

- **Contexto problemático:** El ejercicio 3 requiere segmentar el audio en **ventanas cortas** ($L = 180$ muestras) y aplicar una ventana (*Hamming*) que fuerza los bordes a cero.
- **Efecto de la no sesgada:** La división por un número decreciente de muestras ($N - |k|$) en el estimador no sesgado amplifica el ruido para *lags* grandes, lo que hace que la matriz de *Toeplitz* (\mathbf{R}) se vuelva **mal condicionada** durante la inversión.
- **Consecuencia práctica:** Esto provocaba que el filtro de síntesis fuera inestable o que la estimación de \mathbf{R} fuera errática, resultando en fallos de clasificación V/UV (por ejemplo, el fonema /a/ no era reconocido como sonoro con $\alpha = 0,5$).

Justificación final: La correlación **sesgada** (utilizada finalmente) proporciona mayor **robustez y estabilidad** para la inversión de la matriz de *Toeplitz* en la práctica del LPC segmentado, y es la convención más común en implementaciones prácticas de *vocoders* para garantizar que el filtro de síntesis sea causal y estable.

$$\hat{R}_x(k)_{\text{no sesgada}} = \frac{1}{N - k} \sum_{n=o}^{N-|k|-1} x(n)x(n+k)$$

$$\hat{R}_x(k)_{\text{sesgada}} \approx \sum x(n)x(n+k) \quad (\text{Usado en } \text{param_lpc})$$