ELO 314 - Procesamiento Digital de Señales
Lab. 5 - Guía complementaria:
Análisis y Compresión de la voz usando LPC

Preparado por

Dr. Gonzalo Carrasco, e-mail: Gonzalo.Carrasco@usm.cl

INTRODUCCIÓN

En esta guía se presentan actividades que permiten experimentar un poco más con los temas tratados en el laboratorio 5 del ramo usando Matlab. Los objetivos de esta guía complementaria son:

- Comprender más detalles de el filtro auto-regresivo (AR) usado en LPC
- Experimentar con el cambio de orden del filtro AR
- Experimentar con el filtro inverso y analizar el residuo de predicción

IV. FILTRO AUTOREGRESIVO

Rescatando lo aprendido en las experiencias anteriores, y con el fin de madurar aún más los conceptos de funciones de transferencia de sistemas LTI, debe experimentar con cambiar el orden del filtro AR de solo polos, analizando la relación entre la mayor cantidad de grados de libertad (existen p coeficientes a_k que son los grados de libertad) y la forma de la respuesta en frecuencia de magnitud que puede lograrse para aproximar los formantes de un sonido vocal. Recordar que la respuesta de fase en codificación por predicción lineal (LPC) no es relevante para descomprimir (sintetizar) la voz humana para ser escuchada por nuestros oídos.

Recuerde que un filtro de solo polos (filtro AR) de parámetros a_k siempre se puede representar como una función de transferencia cuyo polinomio de denominador se puede escribir como multiplicatoria de los factores $(z - p_i)$ donde $p_i \in \mathbb{C}$ es la ubicación de los polos en el plano $\mathcal{Z}$ (las raices del polinomio):

$$y[n] = x[n] - a_1\, y[n-1] - a_2\, y[n-2] - \cdots - a_p\, y[n-p] \tag{1}$$

$$\frac{Y(z)}{X(z)} = \frac{1}{1 + a_1\, z^{-1} + a_2\, z^{-2} + \cdots + a_p\, z^{-p}} \tag{2}$$

$$\frac{Y(z)}{X(z)} = \frac{z^p}{z^p + a_1\, z^{p-1} + a_2\, z^{p-2} + \cdots + a_p} \tag{3}$$

$$H(z) = \frac{z^p}{\prod_{i=1}^{p} (z - p_i)} \tag{4}$$

Si todos los coeficientes a_k son reales, entonces los p polos del sistema son reales o complejos, y en el último caso siempre se presentan en pares complejos conjugados. Esto implica que si p es impar, existirá siempre al menos un polo real y a lo más $(p-1)/2$ pares de polos complejos conjugados. Son los pares de polos complejos conjugados de particular interés en el análisis de LPC, pues cada uno de los pares representan un *resonador* y que permite la existencia de un máximo local el la respuesta en frecuencia de magnitud del filtro AR.

En consecuencia con el párrafo anterior, si se desea que el filtro AR modele una determianda cantidad de formantes F (máximos locales de la respuesta en frecuencia), entonces existe un mínimo orden para el filtro AR: $2F$. Es recomendable que el orden del filtro AR sea impar, para posibilitar que el predictor lineal entregue un polo real que genere la típica respuesta en frecuencia de -6dB/octava que modele la respuesta pasabajos natural de muchos sistemas reales.

Basado en la experiencia de observar el espectrograma por debajo de los 4 kHz de ancho de banda de la vocalización humana donde se observaron a simple vista 4 formantes para algunas vocales, como primera conclusión para elegir el orden del filtro AR es que debe ser al menos de orden: 2F+1. Esto es orden 9 para nuestra observación. Esto coincide con lo mencionado en la literatura con la regla empírica de considerar como mínimo un formante por cada kHz de ancho de banda.

1) Programe una función llamada `[amp, w] = positiveSpectrum(x)` que recibe una señal digital x de un largo cualquiera N (será útil para N grandes). Usando la función `fft()` de Matlab la función `positiveSpectrum(x)` debe entregar el vector `amp` de amplitudes de la DFT solo para el rango de frecuencias positivas correctamente formateado para que en el índice `amp(1)` se encuentre el bin de frecuencia cero y que el último elemento de `amp(N/2)` [1] se encuentre el bin de la frecuencia de Nyquist (o el bin más cercano a ella). `w` es un vector de igual largo que `amp` con los índices de la frecuencia normalizada en [rad/muestra]. Esta función le permitirá después comparar la respuesta en frecuencia de una señal con la de un filtro.
(3pto) Muestre el segmento de código de su función `positiveSpectrum(x)`.
(3pto) Use el resultado de su función `positiveSpectrum(x)` junto con la función `plot()` de Matlab para **mostrar el espectro de la señal** `vowel_a`, entre el rango $[0,\ \pi]$ rad/muestra en dB. **¿Cuantos formantes reconoce?**. º

2) Experimente con varios ordenes de filtro arbitrariamente bajos para la señal `vowel_a` y `vowel_u` usando el comando `lpc()` de Matlab.
(3pto) Grafique de forma superpuesta la respuesta del filtro AR entregado por `lpc()` para orden 15, orden 2 y espectro de la `vowel_a` con **eje vertical en dB**.
Se recomienda usar los vectores retornados por el comando `[h,w] = freqz()` para los filtros y de la función `positiveSpectrum(x)` para usar un mismo gráfico, además de la conversión `mag2db(abs(h))` y `mag2db(amp)`.
(1pto) Usando el comando `zplane(b,a)`, **grafique también la ubicacion de los polos** y ceros del filtro en el plano $\mathcal{Z}$.
(4pto) Grafique de forma superpuesta la respuesta del filtro AR entregado por `lpc()` para orden 15, orden 2 y espectro de la `vowel_u`. **Grafique también la ubicacion de los polos** y ceros del filtro en el plano $\mathcal{Z}$.

(5pto) Comente al respecto de las respuestas en frecuencia que el algoritmo de reducción del error de predicción entrega para orden 2 en ambos casos. **¿Resultan polos siempre reales?, ¿complejos conjugados?, ¿por qué?**.

3) Para poder comparar cómo se oye una vocal sintetizada con filtros AR de orden distinto, se recomienda que cree un script de Matlab donde realice las siguientes acciones:
   - cargar las señales `vowels.mat`
   - inicializar una variable con la señal de la vocal elegida para analizar, por ejemplo: `chosen_vowel = vowel_a`
   - inicializar dos órdenes de filtro AR a comparar, por ejemplo uno de referencia `p1` y prueba `p2`: `p1=15, p2=9`
   - usar su función `exciteV(N, Np)` para crear un estímulo de tren de impulos de 100Hz de 0.5 s.
   - obtener los dos conjunto de coeficientes a_k para ambos filtros (`lpc()`)
   - sintetizar y escuchar la vocal elegida con ambos filtros AR (`filter()`). Puede usar `pause()` entre los sonidos.
   - graficar en dB en un mismo gráfico la respuesta en frecuencia de la vocal elegida, y ambos filtros
   - Finalmente, graficar los polos en el plano $\mathcal{Z}$ para cada filtro.

   Usando su script, experimente con las distintas vocales sintetizadas probando para distintos ordenes del filtro AR, observando las respuestas en frecuencia, ubicación de polos en el plano $\mathcal{Z}$ y cómo suenan.

   (4pto) Para la señal `vowel_a` **muestre la comparación de espectros y polos para orden 9** y **comente** si aprecia diferencias significativas entre el sonido sintetizado respecto al orden 15. Recuerde lo expuesto en la cápsula de *Respuesta en frecuencia de un sistema de solo ceros o solo polos*.
   (4pto) Para la señal `vowel_u` **muestre la comparación de espectros y polos para el mínimo orden en que usted no aprecia diferencias** significativas entre el sonido sintetizado respecto al orden 15. Para esta prueba trate de notar las diferencias pensando en que debe sonar a *la "u" de una misma persona*.
   (4pto) Explique brevemente cual sería su criterio de selección de orden para un filtro AR para LPC.

4) Como se mencionó en experiencias pasadas, es posible y recomendable implementar los filtros IIR de alto orden como una cascada de filtros de segundo orden. *Para el caso particular del filtro AR usado en la síntesis LPC* no es realmente necesario implementarlo como cascadas de 2do orden, dado que es un filtro de solo polos, y los problemas numéricos en los que se puede incurrir al implementarlo en solo una ecuación de diferencias no serán apreciables para precision de punto flotante (los problemas son más evidentes operando con aritmética de punto fijo donde puede haber *overflow* y/o ruido por redondeo de valores pequeños, y cuando existen ceros cerca de los polos).

   Matlab cuenta con el comando `[sos, g] = tf2sos(b,a)` para convertir una función de transferencia (filtro digital) con coeficientes b y a de ordenes mayores a 2 en una serie de filtros biquad correspondientes al equivalente cascada de 2do orden. Además de conocer esta herramienta de Matlab, se usará esta vez para analizar el aporte individual de las

[1] Si N es impar elija hasta $(N-1)/2$

respuestas en frecuencia de cada par de polos (o polo real) del filtro AR a la respuesta en frecuencia total.

(4pto) Revise la documentación de Matlab para el comando `[sos, g] = tf2sos(b,a)` y **explique brevemente** cómo interpretar los elementos de la matriz `sos` y del escalar `g` que entrega a partir de los coeficientes b y a. ¿**a que corresponden las filas de** `sos`?, ¿siempre retorna una matriz `sos` de igual cantidad de filas y columnas?, ¿qué es `g`?.

(4pto) Considere el filtro AR de orden 15 obtenido usando LPC para `vowel_a`. **Obtenga la respuesta en frecuencia para cada biquad** que resulte de la transformación `tf2sos(b,a)` y **grafíquelas de forma superpuesta** en un misma gráfica (en dB) junto con la respuesta en frecuencia total del filtro AR. **Comente al respecto** a la relación de la ubicación de los polos, la respuesta en frecuencia de cada biquad y la respuesta total.

V. Observando el residuo de predicción

Una vez que se ha obtenido un filtro AR para un frame de un sonido vocal, se puede interpretar que la señal $x[n]$ del frame es el resultado de filtrar una señal de entrada $e[n]$ por el filtro AR, $H(z)$:

$$X(z) = E(z) \cdot H(z) \tag{5}$$

donde la única señal conocida es $x[n]$. Sin embargo, es posible invertir el filtro de solo polos fácilmente para obtener el llamdo reciduo de predicción $e[n]$:

$$E(z) = H^{-1}(z) \cdot X(z) \tag{6}$$

$$E(z) = \left(\frac{1}{A(z)}\right)^{-1} \cdot X(z) \tag{7}$$

$$E(z) = A(z) \cdot X(z) \tag{8}$$

donde $A(z)$ es el polinomio de denominador del filtro AR, pero que usado en el numerador de un filtro de solo ceros (filtro FIR) se comporta como el *filtro inverso*.

Si se filtra la señal $x[n]$ por el filtro inverso se obtiene $e[n]$, una señal que si se pasa por el filtro AR obviamente producirá la señal $x[n]$ exactamente. Para el modelo simplificado e idealizado de nuestro tracto vocal con estructura de filtro AR, $e[n]$ es la entrada al filtro y representaría (de modo burdo) lo que las cuerdas vocales emitirían. Pero es una aproximación muy sencilla pues el modelo mismo también lo es. No obstante, $e[n]$ entrega información que permitiría mejorar la síntesis (descompresión) del segmento vocal.

1) Obtenga la respuesta en frecuencia del filtro inverso para la señal `vowel_a` y compárela con el filtro AR de orden 15.
   (2pto) Grafique de forma superpuesta la respuesta del filtro AR y el filtro inverso.
   Si `[h, w] = freqz(1,a,numPoint)` (con `a` el polinomio de denominador) entrega la respuesta del filtro AR, entonces `[h, w] = freqz(a,1,numPoint)` es la respuesta del filtro inverso.
2) Obtenga el residuo usando el filtro inverso y la señal original `vowel_a`. Grafique la señal y observe su *forma de onda en el momento de vocalización*.
   (3pto) Grafíque el residuo para la señal `vowel_a` y filtro AR de orden 15. **Comente respecto** a la pertinencia de modelar este estímulo usando un tren de impulsos en el receptor que sintetiza la señal.

3) Considerando la respuesta en frecuencia del filtro AR analizado junto al espectro de la señal `vowel_a`, es de esperar que la diferencia en el espectro de la señal analizada y la respuesta del filtro quede en el residuo $e[n]$. Se espera que el espectro en frecuencias del residuo tenga una respuesta aproximadamente plana con todo el contenido armónico del sonido vocal. Esto será consecuente con la respuesta temporal de $e[n]$ observada en el punto anterior.
   (3pto) Usando su función `positiveSpectrum(x)` **grafique el espectro en frecuencias del residuo** para la señal `vowel_a` y filtro AR de orden 15. **Comente sobre el contenido armónico y ruido del residuo**, comparado con el espectro de la señal `vowel_a`.

4) En Matlab existe la función `pitch()` que permite obtener una aproximación a la frecuencia fundamental de una señal de audio cuando hay vibración de las cuerdas vocales (cuando hay cierta periodicidad). Si bien internamente usa uno de varios algoritmos no triviales, se basan en la información que queda en el residuo de predicción de la segmentación en frames.
   (3pto) Grafique la autocorrelación del residuo para la señal `vowel_a` y filtro de orden 15. Limite la autocorrelación al rango de ordenes menores a 1000 para poder analizarla mejor. Use subplot para poder **compararla con la autocorrelación de la señal** `vowel_a` en el mismo rango de órdenes. **Comente de donde puede ser más fácil extrar la información de F0** (pitch o frecuencia fundamental) de un segmento vocal.