

Procesamiento Digital de Señales

Lab. 5 - Parte I: Análisis y Compresión de Voz en MatLab

Preparado por

Dr. Matías Zanartu, e-mail: Matias.Zanartu@usm.cl

I. INTRODUCCIÓN

En este laboratorio se estudiará una aplicación particular del procesamiento digital de señales en la cual se manipulan señales de audio asociadas a la voz humana, lo que permite extender conceptos de análisis espectral, introducir modelos auto-regresivos e ideas de compresión de señales en función a dichos modelos. Para una introducción al tema del análisis y compresión de voz humana ver la documentación adicional a esta guía de laboratorio. En esta primera sesión se estudiarán señales de voz típicas, herramientas para su análisis y compresión de señales de voz mediante síntesis vía predicción lineal en Matlab.

II. PREDICCIÓN LINEAL Y SÍNTESIS DE VOCALES

- 1) Cree una función $X = \text{exciteV}(N, Np)$ que represente el sonido de las cuerdas vocales mediante un tren de impulsos, donde N es el largo de la señal en muestras y Np el período (por ende la frecuencia fundamental) también en muestras. El tren de impulsos corresponde a $[1 \ 0 \ 0 \ 0 \ \dots \ 0 \ 1 \ 0 \ 0 \ 0 \ \dots]$, donde cada impulso se separa de otro por Np muestras. Asumiendo que $f_s = 8000\text{Hz}$, genere 1 segundo de esta señal con una frecuencia fundamental de 100 Hz. Escuche la señal y grafique su magnitud del espectro entre $[0 \ f_s/2]$ y en su amplitud en dB. (3 Ptos.)
- 2) Cargue el archivo *vowels.mat* en MATLAB. Utilizando el comando $y = \text{lpc}(x, p)$ de MATLAB (para $x = a, e, i, o, u$), obtenga el filtro AR que simula el efecto modulador del tracto vocal en cada vocal utilizando un orden del filtro $p = 15$. Grafique la magnitud de cada filtro en función de la frecuencia entre $[0 \ f_s/2]$. Comente sus resultados. (6 Ptos.)
- 3) Utilice la señal generada en el punto II.1 para excitar a los filtros obtenidos en el punto II.2. Utilice el comando $y = \text{filter}(1, a, x)$ para estos efectos (note que estos coeficientes siempre comienzan en 1). Genere sonidos sintetizados para cada vocal. Para cada caso, escuche y grafique la magnitud del espectro entre $[0 \ f_s/2]$ y en su amplitud en dB. ¿Puede distinguir auditivamente las vocales en cada caso?. Comente sobre la calidad de audio de la señal sintetizada. ¿Cómo se podría mejorar esta síntesis?. Comente sus resultados. Genere los archivos de audio y guárdelos en formato *.wav*. (6 Ptos.)
- 4) Cree su propia función $y = \text{mylpc}(x, p)$ en MATLAB (para $x = a, e, i, o, u$), utilizando las pautas detalladas en los apuntes de LPC que acompañan este laboratorio. Recuerde que debe resolver $\mathbf{R}_x \hat{a} = \mathbf{r}_x$ para obtener los coeficientes \hat{a} , donde \mathbf{R}_x y \mathbf{r}_x se obtienen en MATLAB utilizando los comandos *xcorr* y *toeplitz*. Repita los puntos II.2 y II.3 utilizando su función *mylpc*, donde deberá considerar que $a = [1; -\hat{a}]$ para mantener la estructura estándar de los coeficientes. ¿Son sus resultados iguales a los anteriores?. Comente y compare sus resultados. Genere los archivos de audio y guárdelos en formato *.wav*. (10 Ptos.)

III. CLASIFICACIÓN DE SEGMENTOS VUS

Cargue el archivo de *test_training_signals.mat* en MATLAB. Cree una función que permita calcular el valor RMS de una señal. Cree también otra función que permita obtener el número de cruces de cero en una señal (puede interpretarse como el número de veces en que se produce un cambio de signo entre muestras consecutivas). El objetivo es utilizar estas funciones para la detección de porciones de la señal con vibración de las cuerdas vocales (V), porciones con voz pero sin vibración de las cuerdas vocales (U), y silencio (S). (2 Ptos.)

- 1) Utilizando las funciones *plot*, *soundsc* y *ginput* de MATLAB obtenga manualmente lo que usted determine que son los segmentos sin señal, con y sin vibración de las cuerdas vocales para la señal *training_signal*.

Grafique la señal original con estas porciones superpuestas en distintos colores. Calcule los valores RMS y de cruces de cero para las porciones VUS y muéstrelas en una tabla. (4 Ptos.)

- 2) Divida la señal *test_signal* en segmentos de 25 ms y calcule los valores RMS y de cruces por cero para cada segmento. Defina automáticamente los segmentos VUS. Para ello defina umbrales para los valores RMS y cruce por cero de cada segmento. Calcule de forma separada para el método RMS, el cruce de ceros, y una combinación de ambos métodos que usted puede definir. Compare estos resultados con una inspección manual. ¿Qué método funciona mejor?. Grafique la señal *test_signal* y las porciones VUS de la señal en cada caso. (10 Ptos.)

IV. SÍNTESIS DE VOZ HABLADA

Cargue el archivo de *test_training_signals.mat* en MATLAB. Divida la señal *test_signal* en segmentos de 25 ms, utilice la función de la detección de porciones VUS, y defina automáticamente estos segmentos en base a los umbrales del punto III.2. Genere además un vector que indique si la ventana es VUS con flags (1,-1,0), respectivamente. Calcule la energía (RMS) de cada ventana y guardar los valores en un vector. Calcule los coeficientes LPC de cada ventana mediante su función *mylpc* y guárdelos además en una matriz *A*. (4 Ptos.)

Sintetice cada ventana de la señal utilizando las siguientes ideas:

- 1) Si la ventana tiene vibración de las cuerdas vocales (V): Utilice su función $X = exciteV(N, Np)$ junto con $y = filter(1, a, x)$ de igual forma que en el punto II. Utilice una frecuencia de 100 Hz en este caso. (4 Ptos.)
- 2) Si la ventana tiene voz, pero sin vibración de las cuerdas vocales (U): Utilice ruido blanco $X = rand(1, N)$ filtrado por $y = filter(1, a, x)$. (4 Ptos.)
- 3) Si la ventana no tiene voz (S): inserte ceros para toda la ventana utilizando el comando *zeros(1, N)* de MATLAB. (4 Ptos.) Normalice la señal resultante, de modo que cada ventana de la señal sintetizada tenga la misma energía que la ventana correspondiente de la señal original. Junte todas las ventanas para crear una señal sintetizada completa. Escuche esta señal sintetizada y comente sobre su calidad de audio. ¿Cómo se podría mejorar esta síntesis?. Comente y compare sus resultados. Genere los archivos de audio y guárdelos en formato *.wav*. (4 Ptos.)
- 4) Cree una función que busque la frecuencia fundamental de una señal x en el rango de 80 Hz a 150 Hz. Utilice la DFT de la señal y la función $[C, I] = max(x)$ para realizar la búsqueda (asuma que f_0 corresponde al valor máximo en la magnitud del espectro entre 80 Hz y 150 Hz). Utilice esta función para mejorar la síntesis en el punto II mediante una mejor estimación de Np . (4 Ptos.)
- 5) Estime la razón de compresión entre la señal original y la sintetizada, comparando el número de bytes utilizados por MATLAB para almacenar la señal original con la suma del número de bytes de la matriz *A* y los flags para VUS y energía. Utilice el comando MATLAB *whos* para este punto. (3 Ptos.)

Informe de Laboratorio:

Para todos los puntos de este laboratorio: Presente las funciones de MATLAB requeridas, junto con sus cálculos, ecuaciones y gráficos. Comente sus observaciones y etiquete sus gráficos adecuadamente. Los códigos generados y el informe deberán ser enviados vía email en un archivo comprimido.