

# Procesamiento Digital de Señales

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

Preparado por

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

### 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.

### I. PREDICCIÓN LINEAL Y SÍNTESIS DE VOCALES

- 1) Cree una función  $X = exciteV(N, Np)$  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 define 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  $Np - 1$  muestras en cero. Asumiendo que  $f_s = 8000$  sps, genere 1 segundo de esta señal con una frecuencia fundamental de 100 Hz.  
**(3ptos)** Muestre su **segmento de código**. Escuche la señal y **grafique su magnitud** del espectro entre  $[0 \ f_s/2]$  y su amplitud en dB.
- 2) Cargue el archivo *vowels.mat* en MATLAB. Utilizando el comando  $y = lpc(x, p)$  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  $p = 15$ .  
**(5ptos)** **Grafique** la magnitud de cada filtro en función de la frecuencia entre  $[0 \ f_s/2]$ . **Comente** sus resultados.
- 3) Utilice la señal generada en el punto I.1 para excitar a los filtros obtenidos en el punto I.2. Utilice el comando  $y = filter()$  para estos efectos (note que estos coeficientes siempre comienzan en 1). Genere sonidos sintetizados para cada vocal.  
**(5ptos)** 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?. Genere los **archivos de audio** y guárdelos en formato *.wav* como: *matlab\_vowel\_a* a *matlab\_vowel\_u*.
- 4) Cree su propia función  $y = mylpc(x, p)$  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*.  
**(8ptos)** **Repita** los puntos la síntesis de las vocales de los puntos I.2 y I.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?. Reporte esta vez en su informe solo las respuestas en frecuencia de sus nuevos filtros. **Comente y compare** sus resultados. Genere los **archivos de audio** y guárdelos en formato *.wav* *mylpc\_vowel\_a* a *mylpc\_vowel\_u*.

## II. CLASIFICACIÓN DE SEGMENTOS VUS

Cargue el archivo de *test\_training\_signals.mat* en MATLAB. Cree una función que permita obtener el número de cruces de cero por milisegundo en una señal (puede interpretarse como el número de veces en que se produce un cambio de signo entre muestras consecutivas, dividido por el largo en ms de la señal, por tanto es de tipo no entero). El objetivo es utilizar esta función junto a la función *rms()* 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).

- 1) **Entrenamiento.** Utilizando las funciones *plot*, *soundsc* y *ginput* de MATLAB obtenga manualmente lo que usted determine que son los segmentos sin señal, de señal con y sin vibración de las cuerdas vocales para la señal *training\_signal*. Intente identificar por separado cada letra.

(6ptos) **Grafique** la señal original con estas porciones superpuestas en distintos colores. **Calcule los valores RMS y de cruces por cero** para cada letra y silencios además de clasificarlas como V, U, o S y muéstrelas en una **tabla**. **Grafique** mediante marcadores y colores distintos para V, U y S un **diagrama de nube de puntos (scatter plot)** en que cada punto representa el segmento de la cada letra identificada, donde su posición en X es su valor RMS, y su posición en el eje Y es el número de sus cruces por cero. Identifique los patrones de forma visual, y establezca criterios simples para distinguir los sonidos V, U y S, definiendo umbrales por ejemplo.

- 2) **Prueba.** Divida la señal *test\_signal* en segmentos de 20 ms y calcule los valores RMS y de cruces por cero para cada segmento. La clasificación VUS se debe hacer usando los criterios elegidos previamente para los valores RMS y cruce por cero, esta vez aplicada a cada segmento (frame) de 20 ms.

(8ptos) 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 es más costo-efectivo para lograr un buen resultado?. Indique el **método y parámetros de selección** para sus mejores resultados. **Grafique** usando *subplot* la señal *test\_signal*, la variable VUS (1, -1 o 0) y el valor RMS (puede usar la función *stairs()*) para los segmentos de la señal.

## III. SÍNTESIS DE VOZ HABLADA

Los pasos anteriores permiten configurar la compresión de la voz similarmente a lo que ocurre en un terminal transmisor de la voz. De esta manera, solo la información resultante del análisis de la voz es enviada a un receptor para que realice la síntesis.

En esta sección, debe sintetizar la voz como lo haría el receptor que descomprime la información. Pero dado que no hacemos la transmisión de los datos comprimidos en este laboratorio, se hará la compresión y descompresión en actos seguidos.

Recuerde que, para analizar y comprimir la voz debe:

- Cargue el archivo de audio, en este caso, *test\_signal* del archivo *test\_training\_signals.mat* en MATLAB.
- Divida la señal de audio en segmentos de 20 ms
- Defina automáticamente estos segmentos en base a los umbrales del punto III.2 utilizando la función de la detección de porciones VUS.
- Genere y guarde en vectores de Matlab los parámetros relevantes para cada segmento
  - un vector que indique si la ventana es VUS con flags (1,-1,0)
  - el valor RMS (como medida de la potencia) cada ventana
  - los coeficientes LPC de cada ventana mediante su función *mylpc* y guárdelos además en una matriz *A*

- 1) Sintetice cada ventana de la señal utilizando las siguientes ideas:

- 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.

- Si la ventana tiene voz, pero sin vibración de las cuerdas vocales (U): Utilice ruido blanco  $X = rand(1, N)$  filtrado por  $y = filter(1, a, x)$ .
- Si la ventana no tiene voz (S): inserte ceros para toda la ventana utilizando el comando `zeros(1, N)` de MATLAB.

Obtenga el valor RMS de la señal resultante por cada ventana, y corrija su amplitud para lograr el valor RMS original del segmento de modo que cada ventana de la señal sintetizada tenga la misma potencia que la ventana correspondiente de la señal original.

(8ptos) Junte todas las ventanas para **crear una señal sintetizada** completa. Escuche esta señal sintetizada y gráfiquela junto a la señal original, y **comente** sobre su calidad de audio. ¿**Cómo** cree se podría **mejorar** esta síntesis?. Genere los **archivos de audio** y guárdelos en formato `.wav`, por ejemplo `my_test_signal.wav`

- 2) (2ptos) 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 matrix  $A$  y los flags para VUS y potencia, respecto a la tasa de transmisión cruda de un canal a 8 ksps. Utilice el comando MATLAB `whos` para este punto.

### 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.