

Procesamiento Digital de Señales

Lab. 5 - Parte II: Análisis y Compresión de Voz en la LCDK

Preparado por

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

I. PREDICCIÓN LINEAL Y SÍNTESIS DE VOCALES

En este laboratorio se trabajará con los conceptos de Predicción Lineal y Síntesis de Voz vistos en el laboratorio pasado. Como diseño conceptual se podrá utilizar el código programado en MatLab, pero la implementación se realizará en la LCDK utilizando como estructura los archivos fuente entregados en la sección Archivos. Las actividades de este laboratorio para lograr la síntesis de voz mediante predicción lineal, están segmentadas de forma progresiva de la siguiente manera:

- 1) Implementar el método de predicción lineal de orden 15 ($p = 15$) y *frame* de 10 (ms). Considere una frecuencia de muestreo de 8 (kHz). Para ello se deben estimar los coeficientes *lpc* utilizando el algoritmo Levinson-Durbin, el cual se encuentra programado en archivo `levinson.c`. Este algoritmo requiere solo del vector de autocorrelaciones del *frame*. Para el cálculo del vector \mathbf{r}_x se debe implementar la función de correlación cruzada `xcorr()` de la señal de entrada en el archivo `functions.c`. Considere que la señal de entrada primero pasa a través de una ventana *hamming* (definida en `const_arrays.h`). Luego, con el vector \mathbf{r}_x se deben encontrar los coeficientes *lpc* con el algoritmo dado (recuerde que este algoritmo resuelve la ecuación $\mathbf{R}_x \hat{\mathbf{a}} = \mathbf{r}_x$, donde el vector $\hat{\mathbf{a}}$ corresponden a los coeficientes *lpc*). Incluya un ejemplo de coeficientes encontrados verificando sus valores. (8 Ptos.)
- 2) Finalmente, para la síntesis de voz se debe implementar la función `filter()` considerando que la función de excitación corresponde inicialmente solo a una señal delta (definida en `const_arrays.h`). (4 Ptos.)
- 3) Modifique el algoritmo del punto I.1 de modo que pueda identificar la presencia de voz en la entrada. Para ello defina alguna estrategia para la detección de voz (por ejemplo un umbral RMS) para cambiar la excitación al filtro entre *frames* de voz y silencio. Para el caso de silencio utilice como excitación ruido blanco implementado desde `excitation_noise` en `const_arrays.h`) con datos obtenidos con el comando `rand` de Matlab, o bien generados en tiempo real por la función `prand()` de la LCDK. Compare los resultados de ambos casos. Registre un archivo de audio con la mejor síntesis lograda con el mensaje: *la prueba de los tres tristes tigres*. (6 Ptos.)
- 4) Estudie la posibilidad de modificar su algoritmo *lpc* considerando una señal de excitación dada por una señal de pulsos desde un generador de funciones de frecuencia variable. Comente sobre el efecto de estimular con esta señal al sintetizador de voz. (3 Ptos.)

Informe de Laboratorio:

Presente sus diseños, ecuaciones y los gráficos requeridos. Comente sus observaciones al implementar cada función en las plantillas e archivos fuente entregados. Los códigos generados y el informe deberán ser enviados vía *email* en un archivo comprimido.