Vocoder de Aplicación Musical en Tiempo Real

Rodriguez Turco, Martín Sebastián y Diaz, Ian Cruz

***Abstract*—Proponemos el diseño de un sistema Vocoder que permite cambiar la frecuencia fundamental de la voz y además es capaz de generar acordes mediante una combinación adecuada de frecuencias previamente seleccionadas en tiempo real. Esto tiene una amplia utilización en sistemas musicales para producir voces sintéticas.**

I. Introducción

Para el siguiente proyecto se busco implementar un **Vocoder** de aplicación musical. Un vocoder es un sistema que analiza el sonido de una persona hablando, y mediante diferentes algoritmos realiza cambios en los pulsos glotales sin alterar el filtro articulatorio, y de esta manera logra sintetizar un sonido artificial de la señal de voz de la persona que este hablando. Junto con esto, y eligiendo frecuencias fundamentales de los pulsos glotales acordemente, se pueden generar melodías y acordes para utilizar en ambientes musicales.

II. Enfoque

Para poder realizar este sistema se decidió utilizar el enfoque *Linear Prediction Coefficients* (LPC), que se detallará brevemente a continuación.

A. Modelo de producción de voz

Si recordamos del modelo de producción de voz, podemos simplificar el modelo en los bloques de la Figura 1.

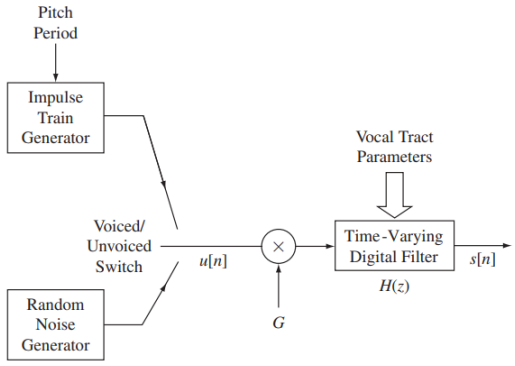


Fig. 1: Diagrama en bloques del modelo de producción de voz

Realizando algunos cálculos [1] [2], se llega a la conclusión que el filtro error de predicción estima la inversa del filtro articulatorio, por lo tanto, simplemente estimando el filtro error de predicción podemos utilizar el filtro articulatorio obtenido de una porción de audio para sintetizar unos pulsos glotales artificiales.

B. Implementación

Para la implementación de este sistema, se realizó el método de la autocorrelación para estimar los coeficientes del filtro de error de producción y una ventana de *hann* para realizar la sintetización[1].

Para poder realizar esto, el código implementado siguió los siguientes pasos[2]:

1) Tomar un bloque de muestras de tamaño N y aplicar ventana de *hahn*.
2) Determinar si la voz es sonora o no sonora.
3) Aplicar **LPC** para poder obtener los parámetros del tracto articulatorio.
4) Generar uno o varios trenes de impulsos a las frecuencias deseadas. Esto cambiara el *pitch* de la voz o generara armónicos agradables a la voz.
5) Convolucionar este nuevo tren de impulsos con el pulso glotal artificial elegido.
6) Aplicar el filtro articulatorio extraído en (3)
7) Guardar el bloque procesado y aplicar OLA con el bloque anterior.

C. Determinación de voz sonora

Para la determinación de voz sonora se realizó un análisis *naive* sobre la señal de entrada. Esto es, realizando pocos cálculos a la señal de entrada y sin demasiado procesamiento, para esto, dado que ya necesitábamos calcular la autocorrelación de la señal de entrada, se la utilizo a nuestro favor, estableciendo un cierto *threshold* de decisión para vos sonora o no sonora.

Empiricamente se decidió que sí:

$$max(R_{XX}(\tau)) \geqslant 0.2 \Rightarrow Sonora \tag{1}$$

Dandonos resultados lo suficientemente aceptables en nuestras pruebas.

D. Pulsos glotales

Para la implementación de pulsos glotales sintéticos, se utilizaron 4 formas de pulsos distintas, ellas son:

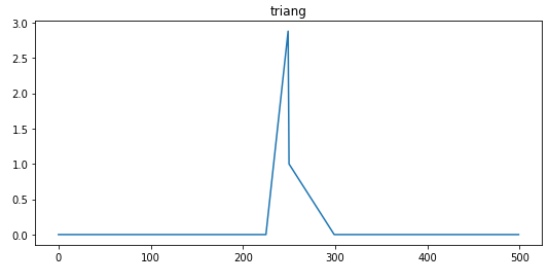
- **Triangular**: Cuenta con una amplia personalización donde se es capaz de ajustar la pendiente de crecimiento, la de decrecimiento, la altura, el ancho, el final de la primera rampa, y el comienzo de la segunda rampa.

[1]Recordemos que para sintetizar una señal correctamente debemos cumplir con el requisito de que la suma de las ventanas en el tiempo debe ser constante, y por lo tanto, con las ventanas de hann, el overlap debe ser del 50%.

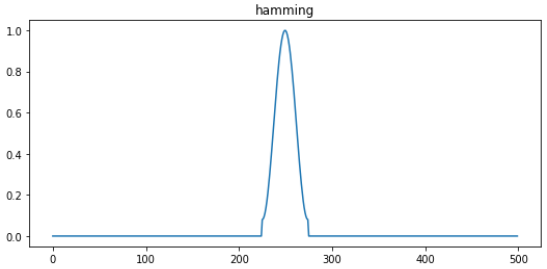
[2]Este código fue basado en [3], donde ese código fue implementado para poder sintetizar la voz con pulsos artificiales, pero intentando modificar lo menos posible la voz de salida respecto de la entrada, por ejemplo para utilizarla en codificación de voz.

- **Hamming**: Es una ventana de hamming centrada, con ancho ajustable y amplitud ajustable.
- **Cuadrada**: Es un pulso cuadrado con personalización del ancho y la altura del mismo
- **Exponencial**: Un pulso centrado en 0 donde la parte negativa es una exponencial creciente y la parte positiva una exponencial decreciente.

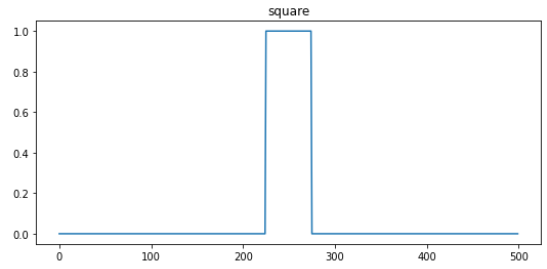
Ejemplos de estos pulsos se pueden ver en la figura 2. Por otro lado, si sintetizamos una señal de voz de ejemplo con estos pulsos glotales, con los siguientes parámetros:



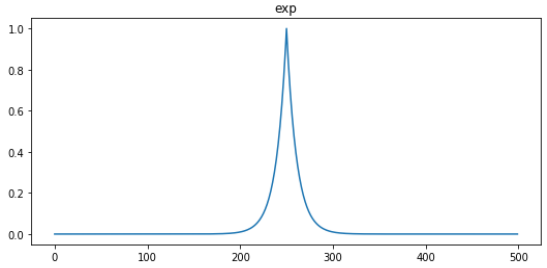
(a) Pulso glotal Triangular



(b) Pulso glotal Hamming



(c) Pulso glotal Cuadrada



(d) Pulso glotal Exponencial

Fig. 2: Pulsos glotales

- Frecuencia = 500 (Hz)
- Overlap = 50%
- Bloque = 32 (ms)

III. Generación de acordes

Musicalmente, resulta útil poseer en nuestro programa la capacidad de que el sistema funcione con acordes y no con tonos puros, para poder hacer la voz más armoniosa musicalmente. Para esto, se debió determinar que acordes se iban a implementar, y determinar sus respectivas frecuencias.

Para esto se decidió implementar solo acordes mayores y menores que son los mas frecuentemente usados, sin embargo, el agregado de nuevos acordes resulta trivial.

Recordemos que un acorde mayor se compone de su frecuencia tonal, su tercera justa y su quinta justa, es decir, su frecuencia tonal, una frecuencia 4 semitonos superior, y otra frecuencia 7 semitonos superior. Para esto por ejemplo si queremos generar un acorde mayor de C_4 (Do), debemos buscar su frecuencia fundamental y con esas determinar las siguientes frecuencias para la generación del acorde, en la Figura 3 se puede ver en el piano como queda la distribución de teclas del acorde de C_4 (Do)[3].

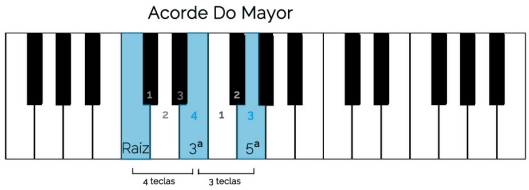


Fig. 3: Acorde de Do Mayor

Por otro lado, para la generación de acordes menores se sigue la misma lógica que para los acordes mayores, pero con la diferencia de que la tercera es disminuida, es decir, se encuentra a 3 semitonos en lugar de 4 como los acordes mayores. Así, si se quiere un acorde de Cm_4 (Do menor) se llega a lo que se ve en la Figura 4.

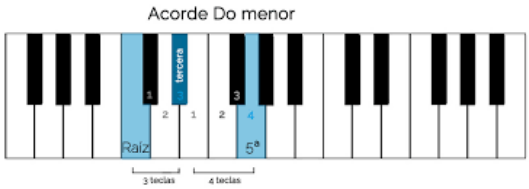
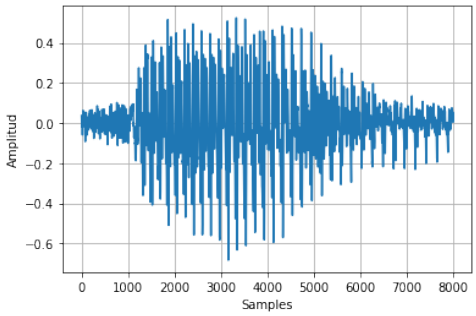


Fig. 4: Acorde de Do menor

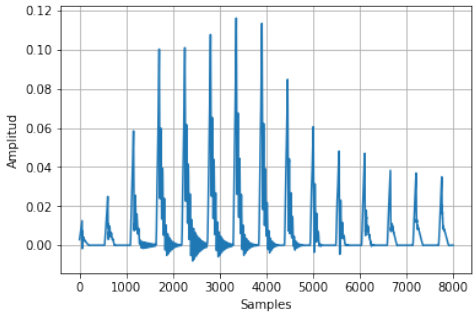
IV. Resultados

Si resintetizamos una señal de voz de una mujer con una frecuencia fundamental de aproximadamente $f_0 \approx 250(Hz)$ se obtienen los resultados que se muestran en la Figura 5.

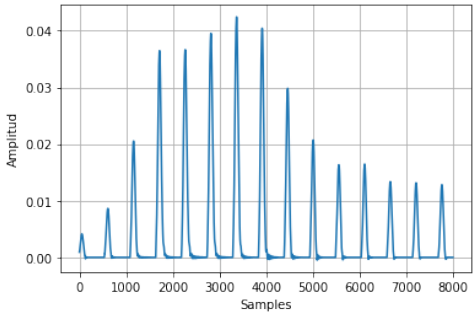
[3]Resulta sencillo ver la distribucion de teclas en el piano, ya que los semitonos son las teclas entre sí en el eje horizontal, es decir, si se quiere subir un semitono, se debe tocar una tecla a la derecha y viceversa.



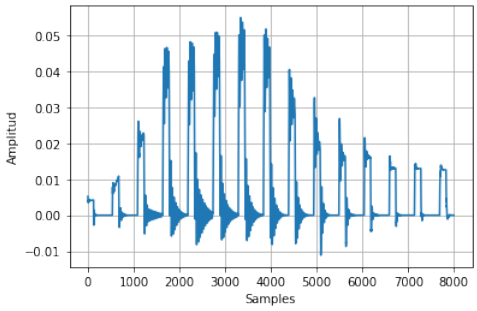
(a) Señal original



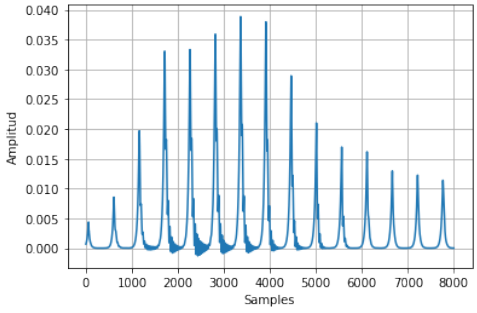
(b) Síntesis con triangular



(c) Síntesis con Hamming



(d) Síntesis con Cuadrada



(e) Síntesis con Exponencial

Fig. 5: Síntesis de señales con diferentes pulsos glotales

REFERENCES

1 Quatieri, T. F., *Discrete Time Speech Signal Proccesing: Principles and practice*, 1st ed. Prentice-Hall, Inc, 2002.

2 Rabiner, L. R. and Schafer, R. W., *Introduction to Digital Speech Processing*, 1st ed. now Publishers Inc., 2007.

3 Bechtold, B., "Pocoder." [Online]. Available: https://github.com/bastibe/pocoder