# ANÁLISIS Y DESARROLLO DE UN PHASE VOCODER

GARCIA ALEJO DANIEL
Diciembre, 2018

Universidad Nacional de Tres de Febrero, Procesamiento digital de señales
alejocsc@gmail.com

Abstract – *En el siguiente informe se desarrolla el funcionamiento de un phase vocoder programado en MatLab. Se analizan variaciones en el algoritmo y su influencia en la señal procesada en aspecto temporal y frecuencial. El algoritmo funciona sobre la transformada de Fourier de tiempo corto (STFT) y permite realizar variaciones de tempo en un archivo de audio sin modificar su contenido espectral y variaciones de frecuencia, sin modificar su duración. Basado en el código escrito por William A. Sethares [1].*

## 1. INTRODUCCION

El phase vocoder, cuya primera presentación data de 1966, es un sistema que procesa el tiempo y la frecuencia de señales de audio. Puede ser visto como una técnica en la que señales de voz son representadas por su fase y espectro en tiempos cortos. Fue diseñado para economizar ancho de banda en transmisiones y para ser un medio para la compresión y expansión de tiempo de las señales [2]. Es también una solución de alta calidad para modificación de escala de tiempo y de tono. Se puede implementar como combinación de escalado de tiempo y conversión de frecuencia de muestreo o como un rastreo de picos de señal y modificación de su posición para luego ser sumado al espectro original [3]. Muchas veces también se utiliza este procesamiento para efectos sonoros buscados en el procesamiento de señales de audio en forma artística.

## 2. MARCO TEORICO

### 2.1 TRANSFORMADA DE FOURIER DE TIEMPO REDUCIDO (STFT)

La transformada de Fourier de tiempo corto (STFT) es un método de análisis y síntesis para el procesamiento de señales, que comienza aplicando una serie de ventanas sobre una señal dividiéndola en pequeños segmentos. Una transformada de Fourier rápida (FFT) es aplicada sobre cada segmento por separado para obtener su contenido espectral. Luego de realizar las modificaciones deseadas se sintetiza la señal a través de una FFT inversa, para recuperar cada segmento en el dominio temporal (Fig. 1). La serie de segmentos se suman y se obtiene, en caso de no haber realizado modificaciones, una señal de salida idéntica a la señal de entrada.

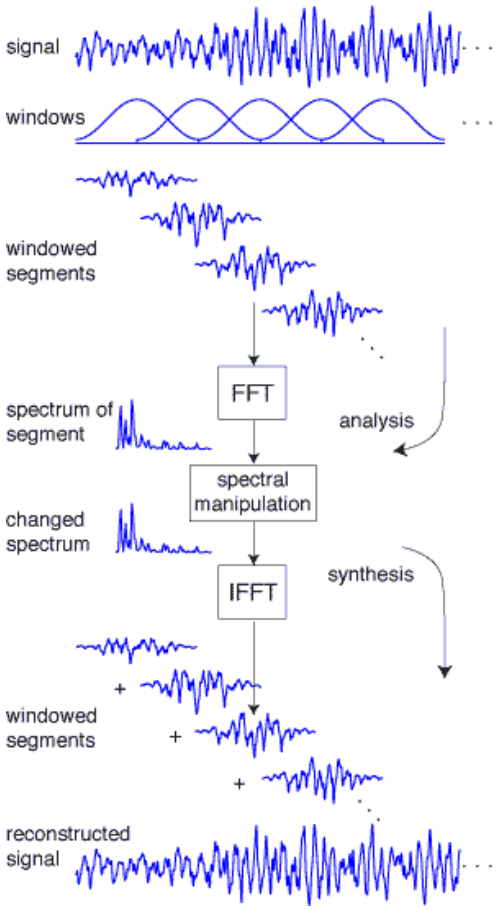


Fig. 1. – STFT y ISTFT sobre una señal

La frecuencia de resolución de la transformada de Fourier está definida por la frecuencia de muestreo de la señal procesada y la longitud de muestras de la ventana (Ec. 1).

$$\boldsymbol{Fresolución} = \frac{\boldsymbol{Frecuencia\ de\ muestreo}}{\boldsymbol{Longitud\ de\ ventana}}$$

Ecuación 1 – Resolución de frecuencia FFT

### 2.2 VENTANAS

Las ventanas son funciones temporales matemáticas que se multiplican con las funciones a analizar en procesamiento de señales. Limitan en tiempo el comportamiento de la señal y dependiendo de su tipo, traerán diversas consecuencias en la transformación de la señal. Por propiedades de transformada de Fourier, al multiplicar en tiempo, se producirá una convolución en frecuencia, con lo que la respuesta en

frecuencia de la ventana puede afectar a la señal. Como también puede tener efecto a la hora de reconstruir la señal. Se muestra en la Tabla 1 la comparación de características relevantes de las mismas en distintos tipos.

| Ventana | Amplitud relativa de lóbulo lateral [dB] | Ancho aproximado de lóbulo principal |
|---|---|---|
| Rectangular | -13 | 4π/(M+1) |
| Bartlett | -25 | 8π/M |
| Hanning | -31 | 8π/M |
| Hamming | -41 | 8π/M |
| Blackman | -57 | 12π/M |

Tabla 1: Varias ventanas y características.

### 2.3 FRECUENCIA DE MUESTREO

La frecuencia de muestreo de una señal implica la cantidad de muestras de nivel por segundo tomadas a la hora en que fue discretizada. De acuerdo al teorema de Nyquist, la frecuencia de muestreo debe ser mayor o igual al doble de la máxima frecuencia de la señal a discretizar, es decir que en señales de audio no debe ser inferior a 40KHz.

### 2.4 SALTO (HOP)

Durante el proceso de STFT las distintas ventanas temporales poseen una separación temporal definida por el salto o HOP. Es el tiempo de retardo entre cada ventana. Es decir que en una señal se obtienen una cantidad de ventanas dependientes del salto como se observa a continuación (Ec. 2)

$$\boldsymbol{Hop = \frac{Longitud\ de\ ventana}{parametro\ de\ solapamiento}}$$

Ecuación 2 – cálculo de longitud de salto, donde el parámetro de solapamiento es un entero.

### 2.5 AJUSTE DE FASE

El ajuste de fase es la característica fundamental del Phase Vocoder. Utilizando la información de fase de otorgada por la FFT, se pueden realizar mejores estimaciones de frecuencia entre distintas ventanas de integración. Conociendo el valor de fase para dos ventanas en una frecuencia, o resolución de frecuencia, y conociendo la diferencia de tiempo entre ambas (HOP), se puede conocer la variación de frecuencia entre una ventana y otra, que no se podía observar dada la resolución de la transformada (Fig.2). Esa variación de frecuencia está dada por la Ec. 3.

$$Fn = \frac{(\theta_2 - \theta_1 + 2\pi n)}{(2\pi(t_2 - t_1))}$$

Ecuación 3 – Calculo de diferencial de frecuencia

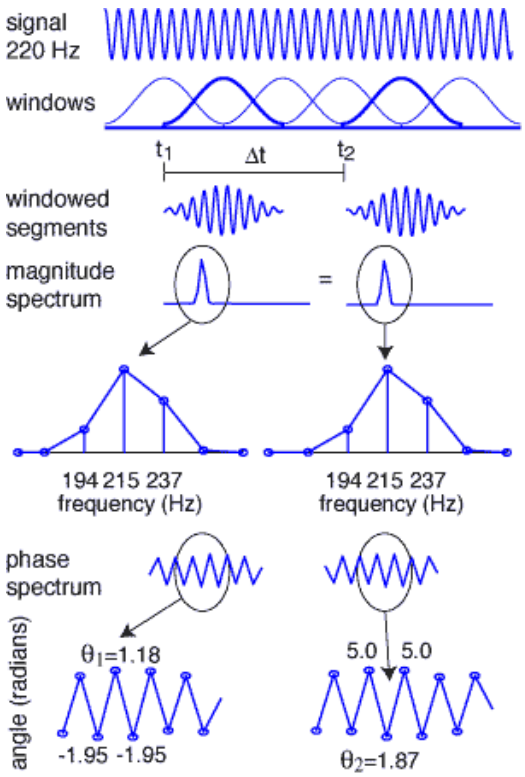


Figura 2. Análisis de fase sobre dos ventanas

Se debe tener en cuenta que la fase obtenida debe ser desenrollada (phase unwrapping) para obtener la fase real de la frecuencia sobre la que se está trabajando, de otro modo se obtienen infinitos diferenciales de frecuencia.

## 3. DESARROLLO DE CODIGO

El desarrollo del código parte del algoritmo realizado por William A. Sethares. Este consta de un único ciclo, donde se trabaja sobre cada ventana temporal. El

ciclo parte a la señal original en muestras del tamaño de la ventana elegida, una vez terminado el primer ciclo sobre la primera ventana, suma la porción de salto (HOP) para analizar la siguiente muestra. Esto nos da una cantidad de ciclos como se observa en la figura a continuación (Ec. 4).

$$Col = \frac{Longitud\ de\ señal - Longitud\ de\ ventana}{Hop}$$

Ec. 4 Cantidad de ciclos for (debe ser redondeado)

Se puede dividir al ciclo en tres etapas. En primer lugar, se procesa la señal realizando una transformada de Fourier de tiempo corto (STFT) la cual nos devuelve los vectores de magnitud y fase de la ventana procesada. Estos vectores tendrán una definición de frecuencia directamente relacionada a la frecuencia de muestreo y la longitud de la ventana. El siguiente paso es el ajuste de frecuencia a partir de fase. Este se realiza sobre las frecuencias fundamentales de la señal analizada. Mientras más frecuencias se tomen más preciso será el análisis y más lento el procesamiento. Estas frecuencias se obtienen a partir de la búsqueda de picos sobre el vector de magnitud con la función FindPeaks4 [5]. Se calcula la fase y magnitud de cada uno de estos picos y se la relaciona con la fase en la siguiente ventana. A partir de esta relación de fase, se realiza un ajuste de fase en la segunda ventana. El último paso, es reescribir a los vectores fase y magnitud de forma compleja para realizar la transformada de Fourier inversa. Se genera el vector tiempo sobre el cual se suman las ventanas procesadas.

### 3.1 MODIFICACION DE TIEMPO

Para modificar el tiempo de la señal procesada sin realizar modificaciones sobre el contenido espectral se debe modificar la relación de salto de ventana (HOP) entre el análisis (STFT) y la síntesis (ISTFT). Si aumento el salto entre ventanas al momento de ubicar las ventanas procesadas sobre el vector tiempo, estoy aumentando la diferencia de tiempo entre ventana y ventana, es decir, el tiempo total de la señal procesada (Ec. 5).

$$\Delta \boldsymbol{Tiempo} = \frac{\boldsymbol{Hop}}{\boldsymbol{Frecuencia\ de\ muestreo}}$$

Ecuación 5. Diferencia de tiempo entre ventanas

En la medida que la relación entre saltos aumente, se deben modificar los parámetros de largo de ventana, tipo de ventana y el parámetro de solapamiento de salto, para mejorar el funcionamiento del código.

### 3.2 MODIFICACION DE FRECUENCIA

Para realizar modificaciones de frecuencia se utiliza el mismo concepto que para la modificación de tempo, agregando un parámetro de desplazamiento (SHIFT). Este parámetro determina la variación sobre la frecuencia de muestreo a la hora de escribir el archivo con la señal de salida. Al modificar la frecuencia de muestreo, se realizan modificaciones de tiempo y frecuencia sobre la señal. Entonces para modificar el contenido de la señal en frecuencia, se realiza una modificación de tiempo sobre la señal de salida y se modifica la frecuencia de muestreo por el mismo factor, de tal modo la duración de la señal no se verá afectada, y si su contenido espectral.

## 4. RESULTADOS

Se procesaron distintos tipos de señales: 3 señales de voz humana, una señal rítmica (batería electrónica) y dos señales de instrumentos acústicos, obteniendo diversos resultados desde un punto de vista psicoacustico, pero a nivel análisis grafico alcanza con mostrar los resultados obtenidos con una señal. A continuación se observan los gráficos de amplitud en función de tiempo para una señal de voz humana sin procesamiento, y tras haberle realizado modificaciones de espectro y duración (Fig. 3).

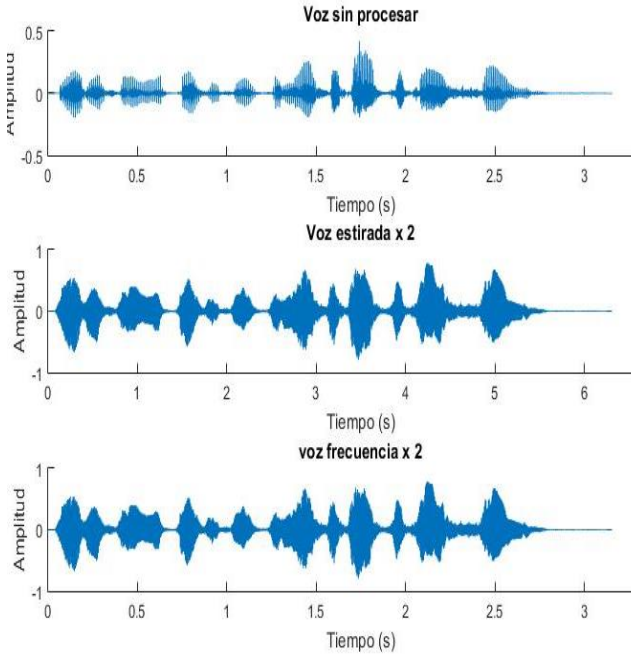


Figura 3 – Tiempo vs amplitud con y sin procesamiento

A continuación se puede observar el espectrograma de la misma señal de voz sin procesamiento y duplicando o dividiendo su longitud (Fig. 4 y 5).

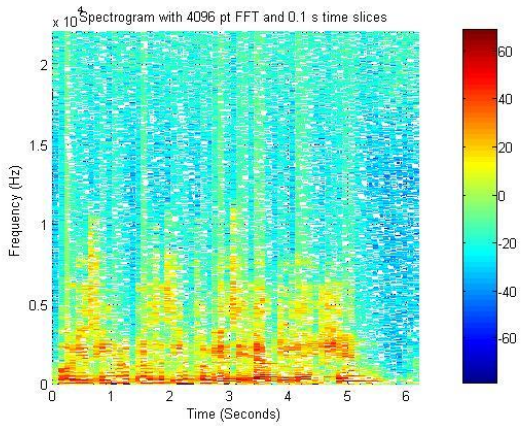


Figura 4 – Espectrograma de voz duplicado el tiempo.

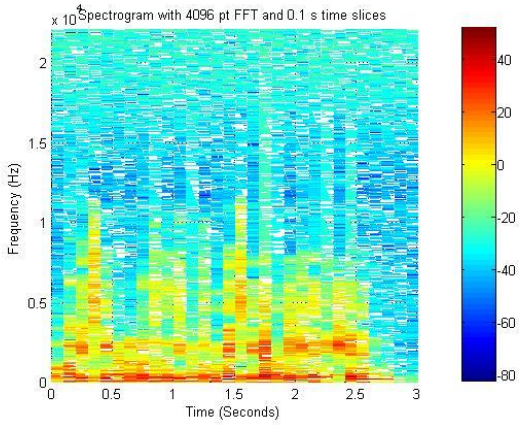


Figura 5 – Espectrograma de voz sin procesamiento

En el siguiente grafico temporal podemos observar la respuesta en frecuencia instantánea de la misma señal de voz con y sin modificaciones de espectro (Fig. 6).

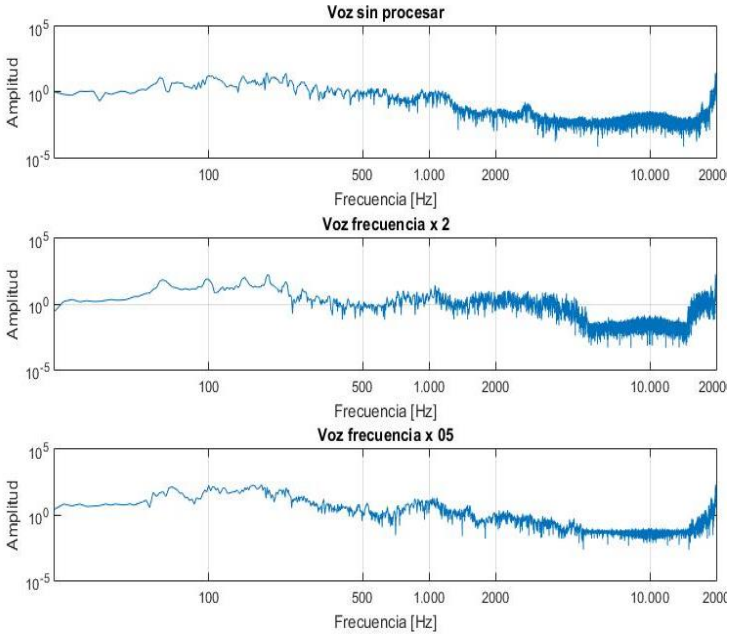


Figura 6 – Espectrograma instantáneo sobre una señal modificada en frecuencia

En la siguiente figura se observa que sucede con la reconstrucción temporal de la señal llevando al extremo la relación ventana y salto (Fig. 7).

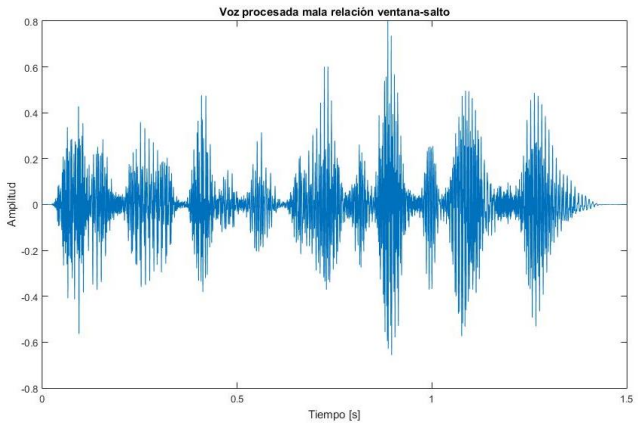


Figura 7 – audio procesado experimentando con relación de ventana y salto

## 5. CONCLUSIONES

El código fue probado sobre distintos tipos de señales y siempre devolvió un comportamiento satisfactorio. Sobre señales de voz, se puede apreciar que el estiramiento y compresión de la señal es útil manteniendo la inteligibilidad hasta altas modificaciones, no así con el desplazamiento en frecuencia, donde duplicar la frecuencia ya alcanza los límites de inteligibilidad. Para sonidos más simples como un violín el código responde de mejor manera. Para sonidos impulsivos, se pierde información al realizar desplazamientos en frecuencia o modificaciones de tiempo dadas sus características en bajas frecuencias.

Es importante ser consciente de las dimensiones de la ventana, de la longitud del salto y del parámetro de relación entre ambos, para optimizar el funcionamiento del código. No todos las señales responden de la misma forma, sonidos impulsivos o de contenido frecuencial complejo será mejor procesado con ventanas grandes y saltos bajos, para tener una mejor definición de frecuencia.

A futuro se debería implementar código para interpolar la señal de salida cuando se realizan modificaciones de frecuencia, para poder editar la frecuencia de muestreo de salida, y asegurarse que sea 44100 Hz, u otro valor normalizado.

El uso de distintas ventanas no demostró variaciones apreciables en la reconstrucción de la señal. Probablemente llevando las condiciones al extremo se pueda apreciar mejor, utilizando mayor hop y menor cantidad de ventanas.

## REFERENCES

[1] William A. Sethares – Phase Vocoder in Matlab (http://sethares.engr.wisc.edu/vocoders/phasevocoder.html)

[2] Flanagan J. L, Golden R. M. "*Phase Vocoder*". The Bell System Technical Journal, vol. 45, pp. 1493-1509. Estados Unidos, Noviembre 1966.

[3] Laroche J, Dolson M. "New phase-vocoder techniques for pitch.shifting, harmonizing and other exotic effects"1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York, Oct. 17-20, 1999

[4] Transforms - William A. Sethares

[5] FindPeaks4 version modified from findPeaks.m by P. Moller-Nielson 28-3-03