

# Algoritmo robusto para la detección de la frecuencia fundamental de la voz basado en el espectrograma

José Antonio Díaz<sup>(1)</sup>, Christine Sapienza<sup>(2)</sup>, Howard B. Rothman<sup>(2)</sup>, Yaser Natour<sup>(2)</sup>

<sup>(1)</sup> Escuela de Ingeniería Eléctrica, Universidad de Carabobo, Valencia, Venezuela

<sup>(2)</sup> Department of Communication Sciences and Disorders, University of Florida, USA

Email: jadiaz@uc.edu.ve

## Resumen

La detección de la frecuencia fundamental de la voz ( $F_0$ ) ha sido un problema particularmente difícil de procesamiento digital de señales. Este parámetro es necesario para documentar la vibración de las cuerdas vocales y alteraciones a sus patrones vibratorios en la presencia de una patología. Existe una gran variedad de algoritmos para extraer y analizar  $F_0$ . En muchas oportunidades se ha reportado que estas técnicas no funcionan bien para diferentes tipos de interlocutores y su rendimiento disminuye a medida que se incrementa el nivel de ruido. El objetivo principal de esta investigación fue el desarrollo de un algoritmo robusto para la extracción y análisis de  $F_0$  de voces normales y patológicas. Este algoritmo se basó en el espectrograma y utiliza técnicas de sistemas expertos para extraer  $F_0$ . El algoritmo desarrollado se probó con seis muestras normales y seis patológicas, las cuales contenían una vocal sostenida. Los resultados proporcionados por el algoritmo propuesto estuvieron en concordancia con los resultados generados por dos paquetes comerciales de análisis de voz, y además, pudo detectar  $F_0$  por un periodo de tiempo mas largo.

**Palabras clave:** Procesamiento de voz, frecuencia fundamental, espectrograma.

## Robust algorithm for the detection of the voice fundamental frequency based on the spectrogram

### Abstract

The detection of fundamental frequency ( $F_0$ ) in speech has often been shown to be a particularly difficult signal processing problem. This parameter is a necessary one for documenting vocal fold vibration and alterations to these vibratory patterns in the presence of pathology. There exist a variety of algorithms for extracting and analyzing  $F_0$ . It has been reported that these techniques do not work well for different types of talkers and decrease in performance as the noise level increases. The main objective of this research was to develop a robust algorithm for the extraction and analysis of  $F_0$  from normal and pathological voices. This algorithm is based on the spectrogram and makes use of artificial intelligence techniques to extract  $F_0$ . An algorithm was developed and tested with 6 normal and 6 abnormal samples, which contain a sustained vowel. These 12 samples were also analyzed by two commercial software packages, which make use of other techniques, and the results were compared to the results provided by the proposed algorithm. The results provided by the proposed algorithm agreed with those of the other two software packages, and the algorithm was able to detect  $F_0$  for a longer period of time.

**Keywords:** Speech processing, fundamental frequency, spectrogram.

## 1. INTRODUCCIÓN

La voz es una onda compleja de la cual pueden extraerse muchos parámetros con el objetivo de modelarla y caracterizarla. Uno de los parámetros mas frecuentemente utilizados y analizados es la frecuencia

fundamental ( $F_0$ ), ya que está directamente relacionado con el "pitch" que nosotros percibimos [1]. También este parámetro es necesario para documentar la vibración de las cuerdas vocales, y alteraciones a sus patrones vibratorios en presencia de una patología [2, 3].

Existe una gran variedad de algoritmos para la extracción de  $F_0$  [4-10]. Algunos de los métodos utilizados mas frecuentemente para el cálculo de  $F_0$  son los siguientes:

1. Detección de cruces por cero.
2. Detección de picos.
3. Comparación de las formas de onda.
4. Autocorrelación.
5. Espectrografía.
6. Cepstrum.

A pesar del número de algoritmos existentes y de los esfuerzos en desarrollarlos, estos algoritmos no trabajan bien para diferentes tipos de interlocutores y su rendimiento disminuye cuando el nivel de ruido aumenta [11-14]. Por lo tanto, es necesaria una mejora de los métodos existentes.

El objetivo principal de esta investigación fue contribuir al área de análisis de voz, desarrollando un algoritmo robusto para la extracción de  $F_0$ , el cual pudiera ser utilizado en voces normales y patológicas.

## **2. SELECCIÓN DE LA MUESTRA**

Primero se establecieron las condiciones que debía cumplir el grupo de muestras con las cuales se probaría el algoritmo propuesto. Se decidió utilizar muestras que contuvieran una /a/ sostenida, lo cual simplificaría la fase de prueba del algoritmo, y se podrían utilizar muestras mas complejas, como discurso continuo, en estudios futuros.

Existía una base de datos de muestras en el departamento de Ciencias y Desordenes de la Comunicación de la Universidad de Florida, la cual contenía muestras de /a/ sostenidas de voces normales y patológicas. Se seleccionaron aleatoriamente doce muestras de esta base de datos, seis de voces normales y seis de voces patológicas.

## **3. DESARROLLO DEL ALGORITMO**

Tal como se indicó anteriormente, existe una amplia variedad de algoritmos para la extracción y análisis de  $F_0$ . Si se analiza la lista de métodos que se muestra arriba, se observa que se puede clasificar en

dos categorías: primera, algoritmos que trabajan en el dominio del tiempo [5-7]. Segunda, algoritmos que realizan una transformación sobre la señal, y extraen  $F_0$  de la señal transformada [9,10].

Los algoritmos que pertenecen al segundo grupo tienen ciertas ventajas. La señal de voz es una onda compleja compuesta de la suma de todos los diferentes armónicos. La extracción de  $F_0$  de esta onda compleja no es una tarea fácil en la mayoría de los casos, y en algunos casos, como en las voces patológicas, esta tarea es aún mas difícil. También, sabemos que el oído humano transforma la señal de voz en sus componentes de frecuencia en la coclea, antes de que los impulsos eléctricos lleguen al cerebro. Esto sugiere que la transformación de la señal de voz al dominio tiempo-frecuencia ayudaría a la detección de  $F_0$ . Además, esta transformación descompone la señal en sus componentes de frecuencia.

La señal de voz puede transformarse al dominio tiempo-frecuencia utilizando varios métodos como el espectrograma, la distribución de Wigner, Wavelets, etc. El espectrograma se seleccionó entre estos diferentes métodos debido a que es robusto frente al ruido y no presenta términos cruzados como sucede con otros métodos.

Se desarrolló un algoritmo para calcular y graficar el espectrograma de una muestra de voz. Varios parámetros debieron seleccionarse a fin de calcular el espectrograma. Estos son los siguientes:

1. Longitud de la ventana
2. Solapamiento de las ventanas
3. Tipo de ventana
4. Frecuencia mínima a ser mostrada
5. Frecuencia máxima a ser mostrada

En vez de especificar estos parámetros en el código de software, se le dio al usuario la posibilidad de cambiarlos a su gusto. Después de calcular el espectrograma, el software lo muestra en una ventana como se indica en la Figura 1. En esta Figura, la frecuencia fundamental ( $F_0$ ) y los armónicos de la muestra de voz pueden ser identificados. Las personas que han trabajado con voz, saben como identificar  $F_0$  visualmente a partir del espectrograma. Existe un conjunto de reglas que nos permite extraer la información de  $F_0$  del espectrograma [1]. Si se

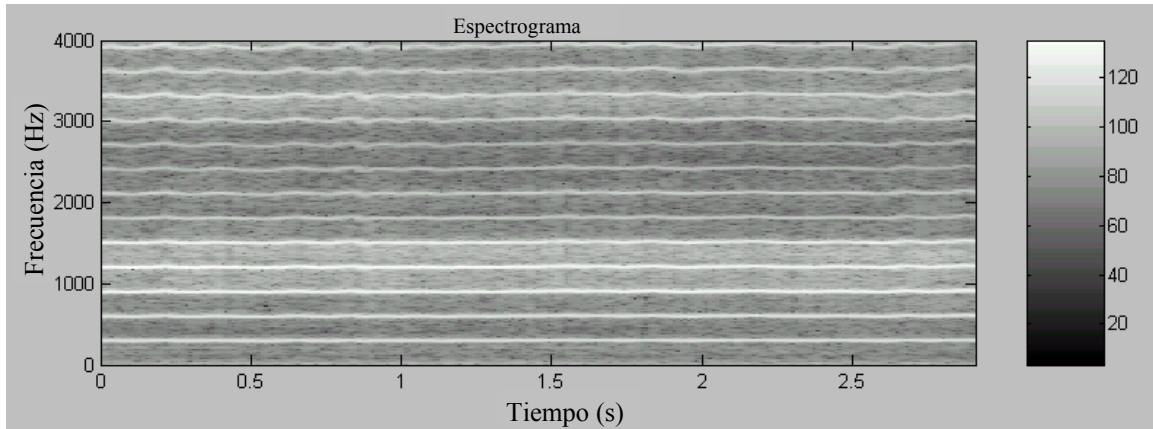


Figura 1. Ejemplo de espectrograma: Frecuencia Vs. Tiempo

implementan estas reglas en un sistema experto, y se le aplican al espectrograma, se estaría en capacidad de extraer  $F_0$  de la misma manera que lo hacemos visualmente.

El espectrograma es una superficie en un espacio tri-dimensional, donde  $F_0$  y los armónicos representan “picos” de esta superficie (Ver Figura 2). Se identifica  $F_0$  visualmente cuando se observa uno de estos “picos”.

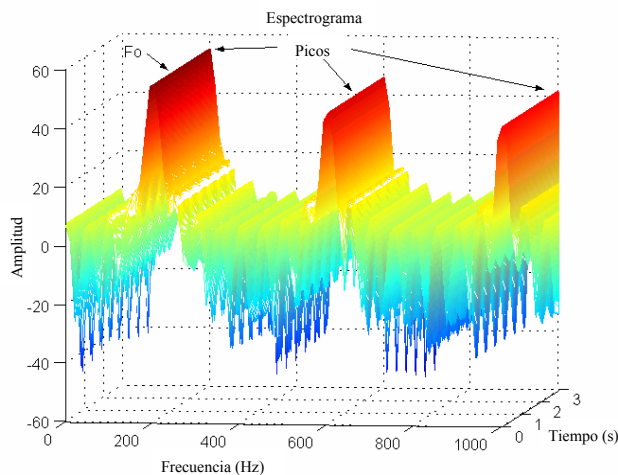


Figura 2. Espectrograma en el espacio tri-dimensional: Amplitud Vs. Frecuencia Vs. Tiempo.

Basado en las reglas que se utilizan para identificar visualmente  $F_0$ , se creó el siguiente conjunto de reglas, y se implementaron en MATLAB:

1. Comenzar en tiempo igual a cero y frecuencia igual a cero (este punto representa el origen en la Figura 1).

2. Si  $F_0$  se ha detectado previamente ir al paso 7, sino ir al paso 3.
3. Aumentar la frecuencia y evaluar cada punto en el espectrograma.
4. Si un punto es un máximo local, determinar la altura del punto.
5. Si la altura es mayor que un nivel umbral, se ha encontrado  $F_0$ , sino, no se halló  $F_0$  en ese instante de tiempo.
6. Aumentar el tiempo, hacer la frecuencia igual a cero (0) y regresar a 2.
7. Encontrar el máximo valor del espectrograma en un rango de frecuencia alrededor del valor de  $F_0$  hallado previamente.
8. Determinar la altura del pico
9. Si la altura es mayor que un nivel umbral, se ha encontrado  $F_0$ , sino, no se halló  $F_0$  en ese instante de tiempo.
10. Aumentar el tiempo, hacer la frecuencia igual a cero (0) y regresar a 2.

Este es el conjunto de reglas básico que se implementó en MATLAB para extraer la curva de  $F_0$ . La versión final del algoritmo contiene mas reglas, las cuales mejoran el rendimiento del algoritmo. Estas son las razones por las que se incremento el número de reglas:

1. No es conveniente comenzar la búsqueda de  $F_0$  en frecuencia igual a cero (0), ya que puede existir una componente DC en la frecuencia cero (0) que puede ser erróneamente detectada como  $F_0$ .
2. No hay necesidad de buscar  $F_0$  a altas frecuencias. Es conocido que la frecuencia fundamental para hombres y mujeres está alrededor de 100 y 200 Hz respectivamente.
3. La superficie del espectrograma tiende a tener amplitudes más bajas a altas frecuencias. Por lo tanto, las alturas de los picos deben compararse al valor promedio de las componentes de frecuencia alrededor de ellos, y no compararse al valor promedio de todas las componentes de frecuencia.
4. La resolución en frecuencia del espectrograma es baja, por lo tanto, es conveniente extraer un armónico mas alto primero, y calcular  $F_0$  dividiendo la curva obtenida por el número del armónico extraído. Esto mejorará la resolución.

Los siguientes parámetros son utilizados por el software para extraer el armónico del espectrograma, y pueden ser seleccionados por el usuario:

1. Frecuencia inicial de búsqueda.
2. Frecuencia final de búsqueda.
3. Ancho de la base.
4. Ancho de la ventana.
5. Altura del pico.

Las frecuencias iniciales y finales de búsqueda representan el intervalo de frecuencias

donde se encuentra el armónico deseado. El ancho de la base es el rango de frecuencias usado para calcular la altura promedio alrededor de un punto en el espectrograma. El ancho de la ventana es el intervalo de frecuencias donde se buscará el armónico deseado cuando ha sido detectado previamente. Es una ventana deslizante centrada en la frecuencia del  $F_0$  detectado previamente. La altura del pico es el nivel umbral utilizado para comparar las alturas de los picos. Alturas por encima de este valor indican que  $F_0$  ha sido encontrado, alturas por debajo de este valor indican que  $F_0$  no ha sido encontrado.

El software calcula la curva de  $F_0$  basado en las reglas descritas arriba y la dibuja sobre el espectrograma, esto le permite al usuario comparar la curva con el espectrograma y decidir si el software detectó la curva correctamente. En caso de que el software falle, el usuario puede cambiar los parámetros indicados arriba, y realizar un nuevo análisis. La Figura 3 muestra el resultado de extraer el armónico 11 de la muestra 1. Se puede apreciar que la curva extraída sigue las oscilaciones del espectrograma, por lo tanto se puede proceder a extraer la curva de  $F_0$ . El usuario debe introducir el número del armónico (11 en este caso) y el software determina la curva de  $F_0$ . La Figura 4 muestra los resultados.

El software también calcula las siguientes estadísticas de la curva de  $F_0$  y las muestra en la ventana de comandos de MATLAB:

1. Frecuencia promedio.
2. Frecuencia máxima.
3. Frecuencia mínima.
4. Desviación estándar.

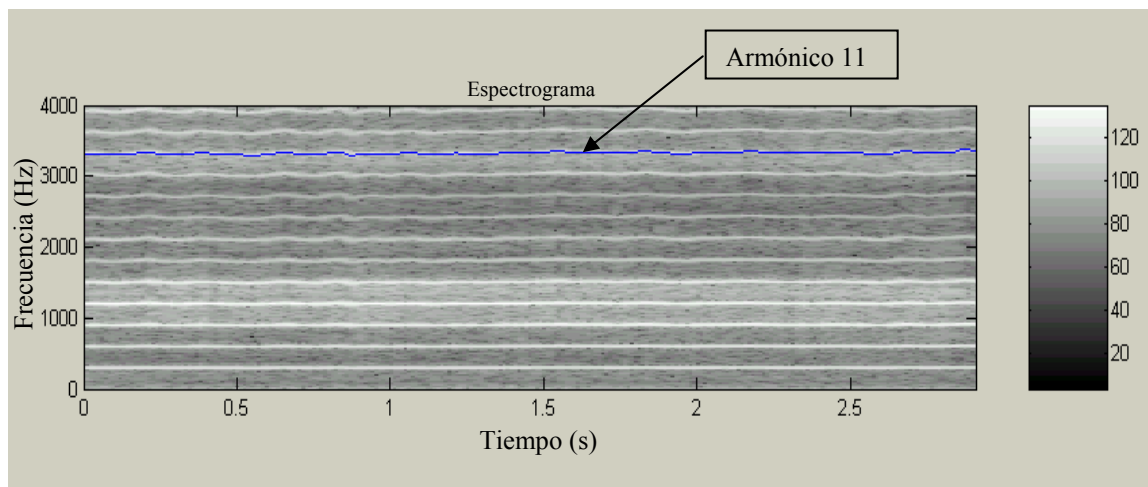
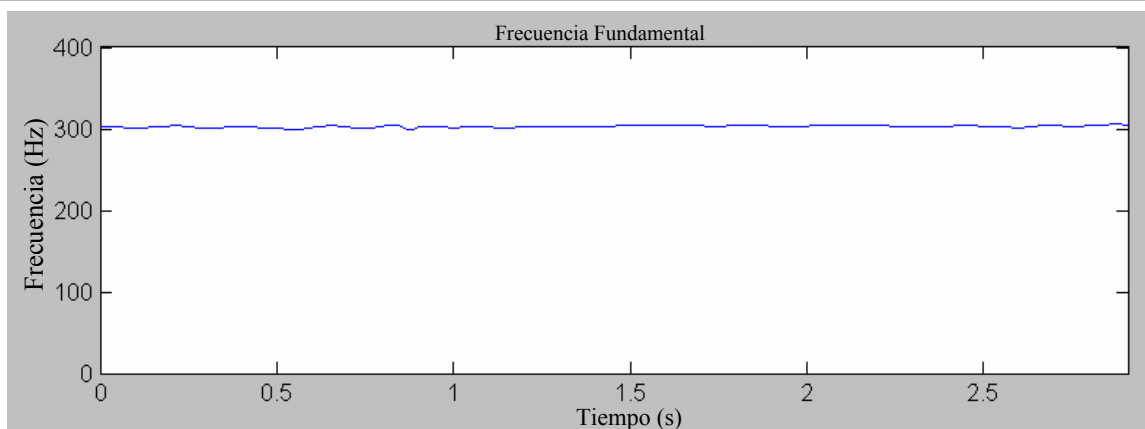


Figura 3. Armónico 11 de la muestra 1.

Figura 4. Curva de  $F_0$  para la muestra 1.

#### 4. RESULTADOS

A fin de determinar la exactitud de los resultados generados por el algoritmo propuesto, las 12 muestras seleccionadas fueron analizadas por dos paquetes de software comerciales y a través del algoritmo desarrollado e implementado en MATLAB. Estos dos paquetes de software comerciales fueron llamados software 1 y 2 en este artículo, y el software desarrollado se llamó algoritmo propuesto.

La Tabla 1 muestra los resultados obtenidos a partir de las 12 muestras bajo análisis. La columna 1 indica el número de la muestra. Las muestras 1 a la 6 corresponden a voces normales, y las muestras 7 a la 12 corresponden a voces patológicas. Las columnas 2 y 3 muestran el valor medio y la desviación estándar obtenidas con el software 1, mientras que las columnas 4 y 5 muestran el valor medio y la desviación estándar obtenidas con el software 2. Las columnas 6 y

7 contienen el valor medio y la desviación estándar obtenidas con el algoritmo propuesto. Las columnas 8 y 9 muestran la diferencia porcentual entre el valor medio de  $F_0$  obtenido con el algoritmo propuesto y los obtenidos con el software 1 y 2. Finalmente, las columnas 10 y 11 muestran la diferencia en Hz entre la desviación estándar obtenida con el software desarrollado y los valores obtenidos con el software 1 y 2.

Si se observan las columnas 8 y 9 de la Tabla 1 vemos que la diferencia porcentual en los valores de  $F_0$  son pequeños para las muestras 1 a la 8 y mayores para las muestras 9 a la 12. La diferencia porcentual para las muestras 1 a la 8 varía entre 0 y 0,23 %. Esto quiere decir que el valor medio de  $F_0$  obtenido con los tres paquetes de software bajo estudio son similares para las muestras 1 a la 8, lo cual no sucede con las muestras 9 a la 12.

Tabla 1.  $F_0$  y su desviación estándar.

| Muestra    | Software 1 |         | Software 2 |         | Algoritmo propuesto |         | Diferencia en $F_0$ (%) |            | Diferencia en Dev Est (Hz) |            |
|------------|------------|---------|------------|---------|---------------------|---------|-------------------------|------------|----------------------------|------------|
|            | $F_0$      | Dev Est | $F_0$      | Dev Est | $F_0$               | Dev Est | Software 1              | Software 2 | Software 1                 | Software 2 |
| Normal     |            |         |            |         |                     |         |                         |            |                            |            |
| Muestra 1  | 302,8      | 1,5     | 302,78     | 1,621   | 302,73              | 1,473   | 0,02                    | 0,02       | 0,03                       | 0,15       |
| Muestra 2  | 143,7      | 0,8     | 143,738    | 1,921   | 143,739             | 0,832   | -0,03                   | 0          | -0,03                      | 1,09       |
| Muestra 3  | 210,9      | 13,2    | 211,764    | 13,607  | 211,284             | 12,017  | -0,18                   | 0,23       | 1,18                       | 1,59       |
| Muestra 4  | 132,4      | 1,7     | 132,452    | 1,94    | 132,438             | 1,608   | -0,03                   | 0,01       | 0,09                       | 0,33       |
| Muestra 5  | 122,3      | 1,2     | 122,293    | 1,249   | 122,327             | 1,081   | -0,02                   | -0,03      | 0,12                       | 0,17       |
| Muestra 6  | 146,2      | 0,5     | 146,242    | 0,669   | 146,235             | 0,497   | -0,02                   | 0          | 0                          | 0,17       |
| Patologica |            |         |            |         |                     |         |                         |            |                            |            |
| Muestra 7  | 109        | 1,2     | 109,037    | 1,476   | 109,011             | 1,258   | -0,01                   | 0,02       | -0,06                      | 0,22       |
| Muestra 8  | 178,5      | 2,3     | 178,561    | 3,824   | 178,547             | 2,22    | -0,03                   | 0,01       | 0,08                       | 1,6        |
| Muestra 9  | 102,1      | 38,5    | 131,392    | 44,913  | 155,405             | 10,161  | -34,3                   | -15,45     | 28,34                      | 34,75      |
| Muestra 10 | 124,4      | 32,7    | 231,025    | 7,625   | 231,174             | 5,196   | -46,19                  | -0,06      | 27,5                       | 2,43       |
| Muestra 11 | 37,7       | 11,9    | 177,533    | 37,625  | 179,62              | 7,367   | -79,01                  | -1,16      | 4,53                       | 30,26      |
| Muestra 12 | 68,1       | 19,4    | 176,701    | 68,987  | 205,409             | 53,459  | -66,85                  | -13,98     | -34,06                     | 15,53      |

Las muestras 9 a la 12 corresponden a voces patológicas. Estas señales son más complejas y difíciles de analizar. Las Figuras 5 a la 7 muestran las curvas de  $F_0$  generadas por los tres paquetes de software para la muestra 3 la cual es representativa de las muestras normales. La forma de las tres curvas es similar. Las Figuras 8 a la 10 muestran las curvas de  $F_0$  generadas por los tres paquetes de software para la muestra 9. Observamos que las tres curvas son diferentes. Es por esta razón que los valores medios de  $F_0$  para las muestras 9 a la 12 son diferentes (Ver Tabla 1).

La Figura 11 muestra el espectrograma y la curva de frecuencia fundamental extraída con el algoritmo propuesto para la muestra 9. Se puede apreciar en esta figura que la curva extraída por el algoritmo propuesto corresponde con la frecuencia fundamental que se muestra en el espectrograma, indicando que el software extrajo la frecuencia fundamental correctamente. También se analizaron los resultados generados por el algoritmo propuesto para las muestras 10 a la 12 y se observó que las curvas de  $F_0$  coinciden con las frecuencias fundamentales mostradas en el espectrograma, por lo tanto, se puede decir que el valor obtenido con el algoritmo propuesto es un buen estimado del valor medio de  $F_0$  para las muestras 9 a la 12.

Si se observan las columnas 10 y 11 de la Tabla 1 vemos que las diferencias entre las desviaciones estándar de  $F_0$  son pequeñas para las muestras 1 a la 8 y son mayores para las muestras 9 a la 12. La diferencia en la desviación estándar para las muestras 1 a la 8 varía de 0 a 1,6 Hz, ésto es, la desviación estándar generada por los paquetes bajo estudio son

similares para las muestras 1 a la 8, lo cuál no sucede para las muestras 9 a la 12. Estas diferencias provienen del hecho de que las curvas de  $F_0$  producidas por los tres paquetes de software para las muestras 9 a la 12 son diferentes, tal como se mencionó en párrafos anteriores.

Si se observa de nuevo la Figura 4, podemos ver la curva de  $F_0$  de la muestra 1. Esta Figura muestra pequeñas variaciones en la frecuencia fundamental de esta muestra. Los parámetros escogidos para el cálculo de la curva de  $F_0$  de esta muestra fueron los siguientes:

Longitud de la FFT = 4.096  
Solapamiento = 3.500  
Frecuencia inicial = 3.100  
Frecuencia final = 3.500  
Ancho de la ventana = 400  
Altura del pico = 10  
Número del armónico = 11  
Los parámetros obtenidos fueron:  
Frecuencia media = 302,730  
Frecuencia máxima = 306,286  
Frecuencia mínima = 299,627  
Desviación estándar = 1,473

Esta muestra fue discretizada a una velocidad de 50,000 Hz. Por lo tanto, la resolución del espectrograma es:

$$f_{res} = \frac{f_s}{N} = \frac{50.000}{4.096} = 12,20Hz \quad (1)$$

Cuando el armónico se dividió por 11 para obtener la curva de  $F_0$ , se obtuvo la siguiente resolución:

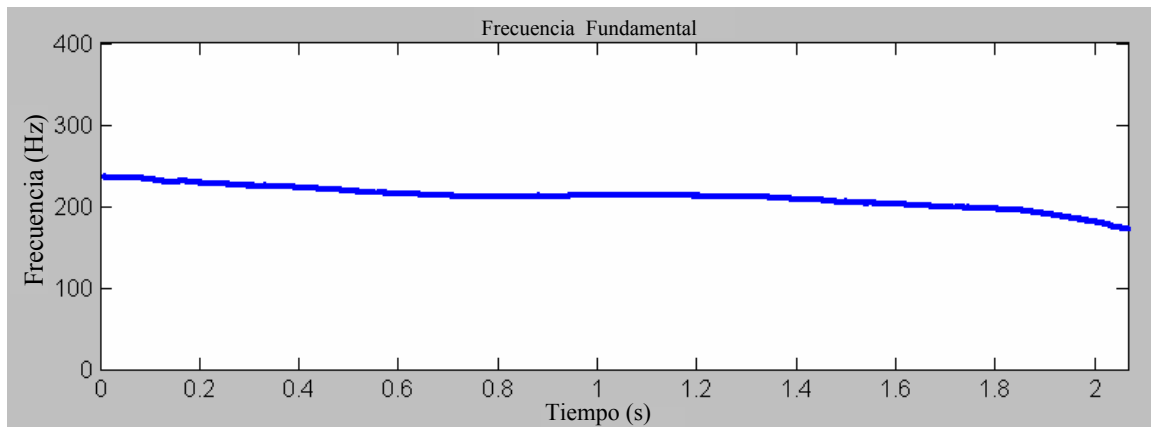
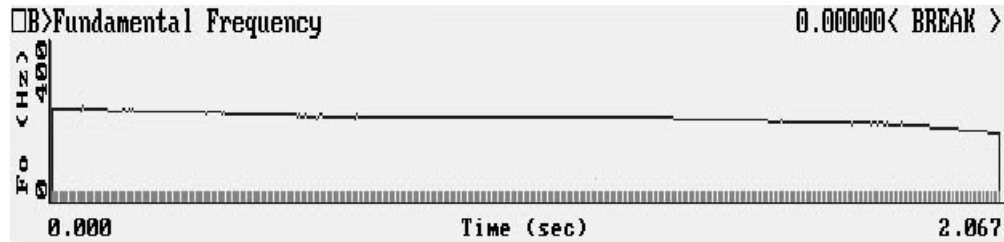


Figura 5. Curva de  $F_0$  generada por el software 1 para la muestra 3.

Figura 6. Curva de  $F_0$  generada por el software 2 para la muestra 3.

$$f_{res} = \frac{12,20}{11} = 1,109 \text{ Hz} \quad (2)$$

Si se hubiera extraído la frecuencia fundamental a partir del primer armónico en vez del armónico 11, la resolución de la curva de  $F_0$  hubiera sido 12,20 Hz.

Con una longitud de la FFT de 4096 muestras y un solapamiento de 3500 muestras, existen 596 muestras entre ventanas sucesivas. Por lo tanto, el tiempo entre ventanas sucesivas (que también es la resolución en el tiempo del espectrograma) es:

$$t_{res} = \frac{596}{50.000} = 11,92 \times 10^{-3} \text{ s} \quad (3)$$

Las Figuras 5 a la 7 muestran las curvas de  $F_0$  dibujadas por los tres diferentes paquetes de software para la muestra 3. Observamos que las tres curvas son continuas. Estas curvas son representativas de todas las muestras normales. Las Figuras 8 a la 10 muestran las curvas de  $F_0$  dibujadas por los tres paquetes de software para la muestra 9, la cual corresponde a una voz patológica. La Figura 8 muestra algunas discontinuidades

y ruido tal como indican las flechas. En la Figura 9 se observa que la curva de  $F_0$  no es continua, mientras que la Figura 10 muestra una curva continua.

Se diseñó un parámetro para medir y comparar la longitud de la curva de  $F_0$ . Este parámetro mide la longitud de la curva de  $F_0$  en porcentaje respecto a la longitud total de la señal. Este parámetro se calculó de la siguiente manera:

$$f_{ol} = \frac{\sum L_i}{L_T} \times 100 \quad (4)$$

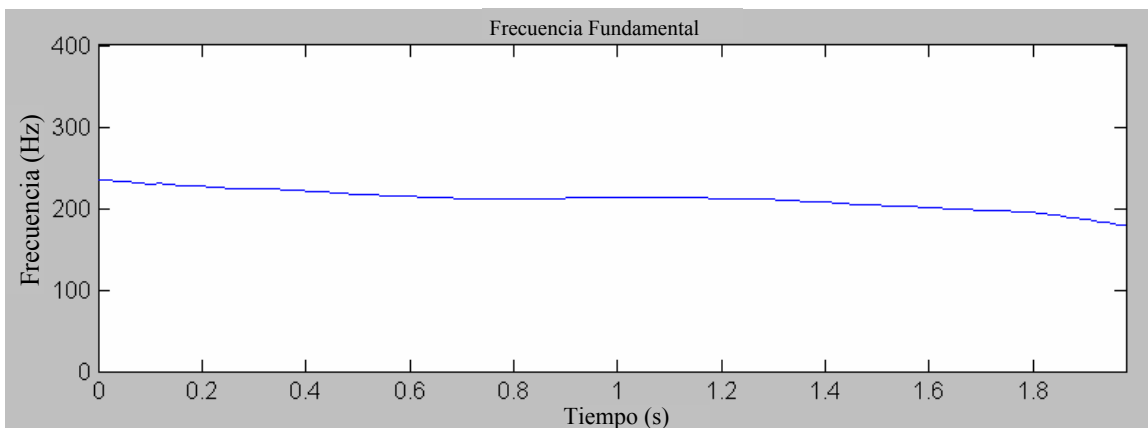
donde:

$f_{ol}$  = Longitud de la curva de  $F_0$  en porcentaje.

$L_i$  = Longitud del  $i_{mo}$  segmento de curva de  $F_0$  en segundos.

$L_T$  = Longitud total de la curva de  $F_0$  en segundos.

Un valor de 100% de  $f_{ol}$  indica que la curva de  $F_0$  es una línea continua como la que se muestra en la Figura 10, un valor menor al 100% indica que la curva de  $F_0$  tiene algunas discontinuidades como las que se ven en la Figura 9.

Figura 7. Curva de  $F_0$  generada por el algoritmo propuesto para la muestra 3.

La Tabla 2 muestra los valores de  $f_{ol}$  para los tres diferentes paquetes de software. Los valores para el software 1 varían de 81,13% a 98,96% para el software 2 varían de 73,61% a 100%, y para el algoritmo propuesto varían de 95,22% al 100%. Si comparamos estos valores observamos que el algoritmo propuesto produce los valores mayores de  $f_{ol}$  para cualquiera de las muestras, lo que indica que pudo detectar la curva de  $F_0$  por un periodo de tiempo más largo.

Tabla 2. Longitud de la curva de  $F_0$  en porcentaje.

| Muestra    | Longitud de la curva $F_0$ (%) |            |          |
|------------|--------------------------------|------------|----------|
|            | Software 1                     | Software 2 | Alg Prop |
| Normal     |                                |            |          |
| Muestra 1  | 98,96                          | 100        | 100      |
| Muestra 2  | 92,03                          | 100        | 100      |
| Muestra 3  | 91,67                          | 100        | 100      |
| Muestra 4  | 92,72                          | 100        | 100      |
| Muestra 5  | 85,65                          | 99,57      | 100      |
| Muestra 6  | 89,95                          | 100        | 100      |
| Patologica |                                |            |          |
| Muestra 7  | 81,29                          | 99,15      | 100      |
| Muestra 8  | 98                             | 99,15      | 100      |
| Muestra 9  | 93,25                          | 80,08      | 100      |
| Muestra 10 | 95,5                           | 87,6       | 97,66    |
| Muestra 11 | 81,13                          | 92,37      | 99,06    |
| Muestra 12 | 92,95                          | 73,61      | 95,22    |

## 5. CONCLUSIONES

Se desarrolló un algoritmo para la extracción y análisis de la curva de  $F_0$ . Este algoritmo se basó en el espectrograma y emplea un conjunto de reglas para extraer un armónico del espectrograma. La curva de  $F_0$

se calculó a partir del armónico extraído al dividirlo por el número del armónico.

La curva de  $F_0$  calculada tiene buena resolución en el tiempo y resolución variable en la frecuencia. El usuario puede cambiar fácilmente la resolución en el tiempo ajustando el solapamiento de la ventana, la cual puede hacerse tan pequeña como un periodo de muestreo. La resolución en frecuencia depende de dos factores: la longitud de la ventana y el armónico seleccionado. Durante el análisis de las muestras se observó que la ventana no debe ser mayor de 80 ms., ya que longitudes mayores producen distorsión o modificación de la curva de  $F_0$ . Mientras más alto sea el armónico seleccionado, mejor será la resolución en frecuencia, sin embargo, algunas veces las muestras son tan ruidosas e irregulares que el algoritmo sólo puede extraer el primer armónico. En algunos casos de muestras poco ruidosas fue posible extraer el armónico 20. La resolución en frecuencia típica del algoritmo varía entre 1 y 12 Hz.

Debido a que el algoritmo propuesto está basado en el espectrograma, éste requiere varios ciclos de la onda de voz para estimar un valor de  $F_0$ , lo cual hace que la longitud de la curva de  $F_0$  comience un corto tiempo después que lo hacen los paquetes de software 1 y 2, y también se tiene un corto tiempo antes. Los valores de  $F_0$  calculados con el algoritmo propuesto son buenos estimados de los valores reales. En 8 de las 12 muestras, los valores calculados concordaron con los de los otros dos paquetes de software. En los casos donde se observaron diferencias, se verificó que los valores obtenidos eran buenos estimados de los valores reales.

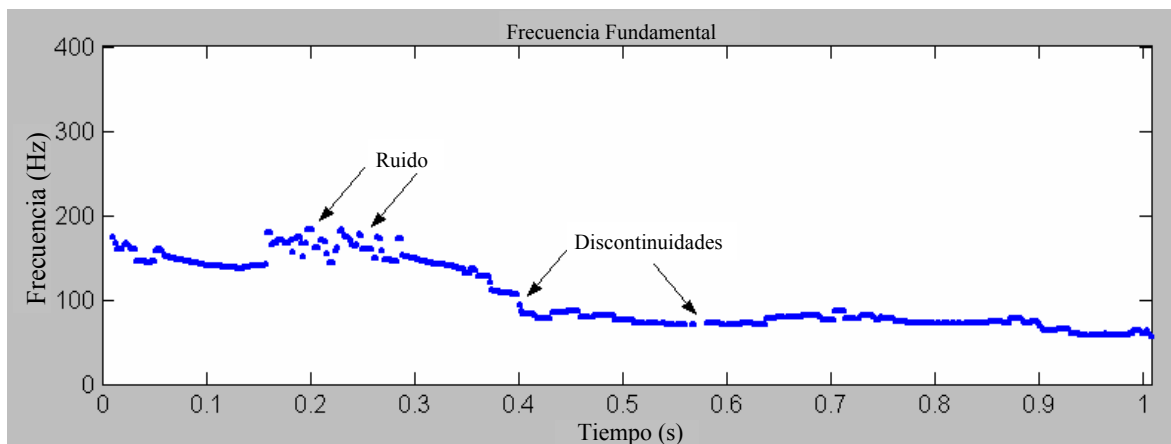


Figura 8. Curva de  $F_0$  generada por el Software 1 para la muestra 9.

El algoritmo propuesto es capaz de extraer la curva de  $F_0$  de muestras difíciles. Una comparación entre los tres paquetes de software bajo estudio, mostró que el algoritmo propuesto extrajo la curva de  $F_0$  por un periodo de tiempo más largo que los otros dos paquetes de software para todas las muestras bajo análisis (Ver Tabla 2).

El algoritmo desarrollado permite que el usuario verifique la exactitud de los resultados obtenidos. Al dibujar el armónico extraído sobre el espectrograma,

el usuario puede determinar si el software extrajo correctamente el armónico o falló. Una vez que el armónico ha sido extraído exitosamente, el usuario puede proceder a calcular la curva de  $F_0$ .

El algoritmo propuesto funciona tanto con voces normales como patológicas. Si se observa la tabla 2, se aprecia que el porcentaje de detección de  $F_0$  para los software 1 y 2 disminuye a un 81 % y 73 % para las muestras patológicas, mientras que el rendimiento del software desarrollado disminuyó a un mínimo de 95 % para la muestra 10.

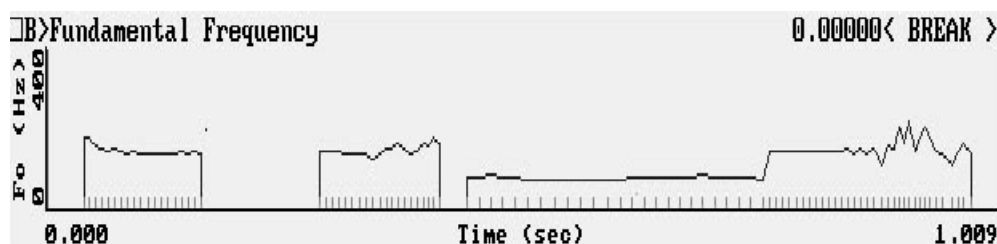


Figura 9. Curva de  $F_0$  generada por el software 2 para la muestra 9.

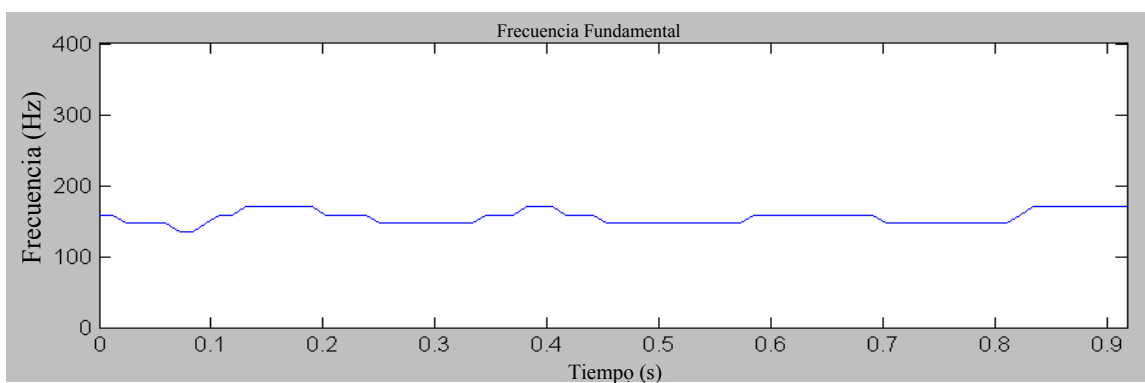


Figura 10. Curva de  $F_0$  generada por el algoritmo propuesto para la muestra 9.

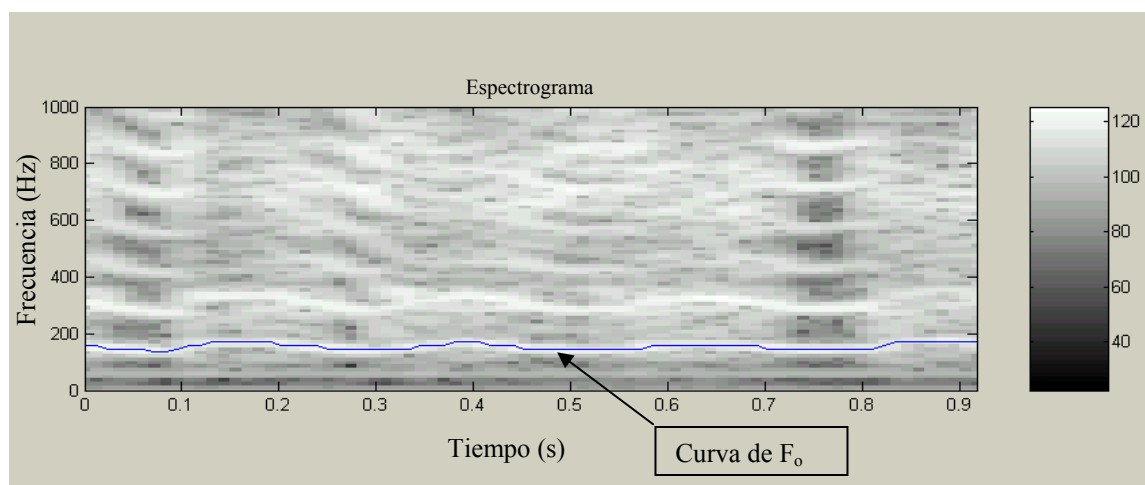


Figura 11. Espectrograma y curva de  $F_0$  para la muestra 9.

## REFERENCIAS

- [1] Deller, J. R., Proakis, J. G., and Hansen, J. H. L., Discrete-Time Processing of Speech Signals, Prentice Hall, Upper Saddle River, New Jersey, 1987.
- [2] Diaz, J. A., A Mathematical Model of Singers' Vibrato Based on Waveform Analysis, University of Florida, Gainesville, Florida, Agosto 1998.
- [3] Mellody, M., and Wakefield, G., The Time-Frequency Characteristics of Violin Vibrato: Modal Distribution Analysis and Synthesis, Journal of the Acoustical Society of America, p. 598-611, Enero 2000.
- [4] Brigham, E. O., The Fast Fourier Transform, Prentice Hall, Englewoods Cliffs, New Jersey, 1974.
- [5] Cappellini, V., and Constantinides, A. G., Digital Signal Processing, Elsevier Science Publishers B. V., North Holland, p. 325-329, 1984.
- [6] Oppenheim, V. A., and Schaffer, R. W., Discrete-time Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey, p. 149-190, 1989.
- [7] Proakis, J., and Manolakis, D., Digital Signal Processing: Principles, Algorithms and Applications, Prentice-Hall, Madrid, p. 153-234, 1998.
- [8] William, T. C., Cooley, J. W., Favon, D. L., Helms, H. D., Kaenel, R. A., Lang, W. W., Maling, G. C., Nelson, D. E., Rader, C. M., and Welch, P. D., What is the Fourier Transform?, IEEE Trans. Audio Electroacoustics, Vol. AU-15, p. 45-55, Junio 1967.
- [9] Childers, D. G., Modern Spectrum Analysis, IEEE Press, New York, New York, 1978.
- [10] Kay, S. M., Modern Spectral Estimation: Theory and Application, Prentice Hall, Englewood Cliffs, New Jersey, p. 153-270, 1988.
- [11] Chan, Y. T., Lavoie, J. M. M., and Plan, J. B., A Parameter Estimation Approach to Estimation of Frequencies of Sinusoids, IEEE Transactions on Acoustics, Speech and Signal processing, Vol. ASSP-29 No. 2, p. 214-219, Abril, 1981.
- [12] Diaz, J. A., Frequency Characterization of Singers' Vibrato, University of Florida, Gainesville, Florida, Diciembre 1995.
- [13] Loughlin, P. J., Spectrographic Measurement of Instantaneous Frequency and the Time-Dependent Weighted Average Instantaneous Frequency, Journal of the Acoustical Society of America, p. 264-274, Enero 1999.
- [14] Yannis, S., Decomposition of Speech Signals into a Periodic and Non-periodic Part Based on Sinusoidal Models, Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Vol. 1, p. 514-517, 1996.