ALTERNATIVA AL ANÁLISIS EN FRECUENCIA DE LA FFT MEDIANTE EL ALGORITMO GOERTZEL

Alternative Frequency Analysis of the FFT Using the Goertzel Algorithm

JIMMY ALEXANDER CORTÉS OSORIO
Ingeniero Electricista, M.Sc
Profesor Asociado
Universidad Tecnológica de Pereira
jacoper@utp.edu.co

JAIRO ALBERTO MENDOZA VARGAS
Ingeniero Electricista, M.Sc
Profesor Asociado
Universidad Tecnológica de Pereira
jam@utp.edu.co

JOSÉ A. MURIEL ESCOBAR
Ingeniero Mecánico, M.Sc
Profesor SENA Dosquebradas
jamuriel@sena.edu.co

RESUMEN

El análisis en frecuencia permite extraer información que no es evidente mediante la simple observación de una señal en el tiempo. La transformada discreta de Fourier (DFT) es la respuesta natural e inmediata a la transformada continua de Fourier en el mundo digital (CFT), pero con limitaciones de tiempo de ejecución debido a la casi total ausencia de optimización de dicho algoritmo. La transformada rápida de Fourier (FFT) resuelve esta limitación en tiempo mejorando de manera importante el tiempo de cálculo consumido por la transformada discreta (DFT), pero deja la puerta abierta para la implementación de variaciones a la transformada discreta de Fourier (DFT) para propósitos específicos.

El algoritmo Goertzel es un filtro digital derivado de la transformada discreta de Fourier (DFT) que puede detectar las componentes de frecuencia específica en una señal, sin analizar todo el espectro, resultando en un menor tiempo de ejecución. Este algoritmo es de gran utilidad en el manejo de los tonos DMFT (Dual-Tone Multi-Frequency) cada vez más usados en los sistemas de reconocimiento de tonos usados por las compañías para prestar o vender servicios a través de teléfonos fijos o celulares.

PALABRAS CLAVE: algoritmo, análisis, DFT, DMFT, FFT, frecuencia, Goertzel, programa, transformada.

ABSTRACT

The frequency analysis extracts information that is not evident by simple observation of a signal in time. The discrete Fourier transform (DFT) is the natural and immediate response to the continuous Fourier transform in the digital world (CFT), but with limited runtime due to the almost total absence of the optimization algorithm. The fast Fourier transform (FFT) solves the limitation in time improving significantly the computing time consumed by the discrete transform (DFT), but leaves the door open for the implementation of changes to the discrete Fourier transform (DFT) to specific purposes.

Goertzel algorithm is a digital filter derived from the discrete Fourier transform (DFT) which can detect specific frequency components in a signal without analyzing the whole spectrum, resulting in a minimum execution time. This algorithm is useful in managing DMFT tones (Dual-Tone Multi-Frequency) increasingly used in the tone recognition systems used by companies to offer services via landline or cellular.

KEYWORDS: algorithm, analysis, DFT, DMFT, FFT, frequency, Goertzel, program, transform.

1. INTRODUCCIÓN

El algoritmo de Goertzel es una técnica para el procesamiento digital de señales (DSP) el cual identifica las componentes de frecuencia de una señal. Este fue publicado por el Dr. Gerald Goertzel[1] en 1958.

Goertzel es un filtro digital derivado de la transformada discreta de Fourier (DFT) que puede detectar componentes de frecuencia específicas en una señal, como por ejemplo para permitir que los circuitos de conmutación telefónica digital con tecnología DSP puedan identificar los tonos característicos generados cuando un número se marca en el sistema. Estas técnicas de procesamiento digital de señales se emplea actualmente en algunas modernas centrales telefónicas digitales.

2. DEFINICIONES

2.1 La Transformada Discreta de Fourier

La transformada de Fourier descompone la señal como la suma de en senos y cosenos de diferentes frecuencias y amplitudes desfasadas en el tiempo.

[1] Gerald Goertzel (1920-2002)

Fecha de Recepción: Enero 26 de 2010
Fecha de Aceptación: Marzo 25 de 2010

En las aplicaciones de ingeniería y tratamiento de señales, se considera el proceso de manera discreta y no continua, puesto que los sistemas de adquisición de datos operan de manera digital[1].

La ecuación 1 presenta la definición de la transformada discreta de Fourier.

$$x(k) = \sum_{n=0}^{N-1} x(n) e^{(\frac{-2\pi kn}{N})j}$$

Ecuación 1 . Definición de la DFT.

N = Corresponde al número total de muestras
n = Es la enésima muestra original
k = Es el késimo término de la DTF

2.2 Deducción del algoritmo de Goertzel [2] [3] [4] [5]
El algoritmo de Goertzel parte de la definición de la DFT, por lo que se procede a realizar, inicialmente, una reescritura de la ecuación 1 para la DFT :
Si se define a:

$$W_N^k = e^{(\frac{-2\pi k}{N})j}$$

$$\left[W_N^k\right]^n = \left[e^{(\frac{-2\pi k}{N})j}\right]^n$$

$$W_N^{kn} = e^{(\frac{-2\pi kn}{N})j}$$

Por lo que la ecuación 1 resulta nuevamente como la ecuación 2, una vez reescrita:

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

Ecuación 2. Reescritura de la definición de la DFT

Como:

$$W_N^{-kN} = \left[W_N^k\right]^{-N} = \left[e^{(\frac{-2\pi k}{N})j}\right]^{-N} = e^{(\frac{-2\pi kN}{N})j} = e^{(-2\pi k)j}$$

Y desarrollando mediante la identidad de Euler:

$$e^{(-2\pi k)j} = \cos(-2\pi k) + \sin(-2\pi k)j = 1$$

Se puede notar, entonces que:

$$W_N^{-kN} = 1$$

Ecuación 3. Identidad de Euler

Haciendo uso de la identidad de Euler, presentada en la ecuación 3, sobre la ecuación 2 se obtiene:

$$x(k) = \sum_{n=0}^{N-1} x(n)\,[1]\, W_N^{kn}$$

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{-kN} W_N^{Kn}$$

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{-kN+kn}$$

Factorizando se llega a la ecuación 4.

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{-k(N-n)}$$

Ecuación 4. Reescritura de la ecuación correspondiente a la DFT haciendo uso de la identidad de Euler.

Si se desarrolla la sumatoria resultante en 4, se obtiene la expresión:

$$x(k) = x(0)W_N^{-k(N-0)} + x(1)W_N^{-k(N-1)} + x(2)W_N^{-k(N-2)} + \ldots + x(N-1)W_N^{-k(N-(N-1))}$$

$$x(k) = x(0)W_N^{-kN} + x(1)W_N^{-kN}W_N^{k} + x(2)W_N^{-kN}W_N^{2k} + \ldots + x(N-1)W_N^{-k}$$

$$x(k) = x(0) + x(1)W_N^{k} + x(2)W_N^{2k} + \ldots + x(N-1)W_N^{-k}$$

Factorizando, se llega a ecuación 5:

$$x(k) = \left\{\left[(x(0)W_N^{-k} + x(1)\right] W_N^{-k} + x(2)W_N^{-k} + \ldots + x(N-1)\right\} W_N^{-k}$$

Ecuación 5. Expresión de la DFT como sumas que llevan a la expresión general el ecuación en diferencias.

Expresando como una ecuación en diferencias la ecuación 5, se tiene la ecuación 6:

$$y(n) = W_N^{-k} y(n-1) + x(n)$$

Ecuación 6. Expresión en ecuación en diferencias de la DFT.

Donde y(n) representa la salida y y(n-1) la salida anterior y x(n) la entrada. Aplicando la transformada Z a la ecuación 6 en diferencias, se obtiene:

$$y(z) = W_N^{-k} y(z) Z^{-1} + x(z)$$

$$y(z) - W_N^{-k} y(z) Z^{-1} = x(z)$$

$$y(z)\left[1 - W_N^{-k} Z^{-1}\right] = x(z)$$

De la donde la función de transferencia resultante es la ecuación 7:

$$H(z) = \frac{y(z)}{x(z)} = \frac{1}{1 - W_N^{-k} Z^{-1}}$$

Ecuación 7. Función de transferencia de la DFT mediante su desarrollo en la transformada Z.

Multiplicando el numerador y el denominador de la función de trasferencia de la ecuación 7 por:

$$1 - W_N^{k} Z^{-1}$$

Se llega a:

$$H(z) = \frac{y(z)}{x(z)} = \frac{1 - W_N^k Z^{-1}}{(1 - W_N^{-k} Z^{-1})(1 - W_N^k Z^{-1})}$$

$$H(z) = \frac{1 - W_N^k Z^{-1}}{1 - \left[(W_N^{-k} + W_N^k) Z^{-1} - Z^{-2}\right]}$$

$$H(z) = \frac{1 - W_N^k Z^{-1}}{1 - \left[2(\cos(\frac{2\pi k}{N})) Z^{-1} - Z^{-2}\right]}$$

Así, se llega a la ecuación 8 que corresponde al filtro derivado del algoritmo Goertzel:

$$H(z) = \frac{1 - W_N^k Z^{-1}}{1 - 2\cos(\frac{2\pi k}{N}) Z^{-1} + Z^{-2}}$$

Ecuación 8. Función de transferencia de la DFT desarrollada.

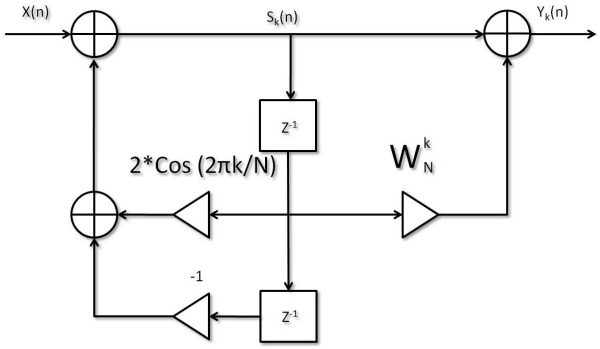

Figura 1. Diagrama del filtro IIR del algoritmo Goertzel de la ecuación 8.

Llevando la ecuación 8, nuevamente, a su expresión en diferencias, se tiene la ecuación 9, que resulta ser una expresión en diferencias del algoritmo:

$$\left\{1 - 2\cos(\frac{2\pi k}{N}) Z^{-1} + Z^{-2}\right\} H(z) = 1 - W_N^k Z^{-1}$$

$$y(n) - 2\cos(\frac{2\pi k}{N}) y(n-1) + y(n-2) = x(n) - W_N^k x(n-1)$$

Ecuación 9. Ecuación en diferencias del algoritmo de Goertzel.

Con el objetivo de implementar el algoritmo, la ecuación 9, se descompone como las ecuaciones 10 y 11.

$$S(n) = x(n) + 2\cos(\frac{2\pi k}{N}) y(n-1) - y(n-2)$$

Ecuación 10. Elemento S(n) del algoritmo.

$$Y(n) = S(n) - W_N^k x(n-1)$$

Ecuación 11. Elemento Y(n) del algoritmo.

3. Implementación del Algorimo en Matlab 7.6

En este apartado, se implementa el algoritmo que permite utilizar el algoritmo de Goertzel. A diferencia del la DFT, la cual busca todas las frecuencias existentes, el algoritmo de Goertzel pretende verificar la existencia de ciertas frecuencias previamente conocidas sin realizar todos los cálculos. Esta aplicación es altamente útil en la detección de tonos DMFT usados en muchos sistemas telefónicos de servicio automático de respuesta para recargas de telefonía móvil, transacciones bancarias y servicio al cliente.

Para efectos del algoritmo, las ecuaciones 10 y 11 se deben reescribir como las ecuaciones 12 y 13 apoyados en el diagrama del filtro IIR de la figura 1.

$$S_k(n) = x(n) + 2\cos(\frac{2\pi k}{N}) S_k(n-1) - S_k(n-2)$$

Ecuación 12.

$$Y_k(n) = S_k(n) - W_N^k S_k(n-1)$$

Ecuación 13.

La ecuación 14 calcula la DFT del k-ésimo término para la frecuencia buscada $F_{buscada}$ con una frecuencia $F_{Muestreo}$ de una señal de N muestras:

$$k = N\frac{F_{buscada}}{F_{Muestreo}}$$

Ecuación 14. K-ésimo término de la frecuencia buscada

Para la implementación se deben considerar que los valores de Sk(-2) y Sk(-1) son cero. El proceso se debe repetir hasta recorrer todos elementos N de la señal digital a explorar.

```
function x=goertzelmio(senal,fbuscadas,fmuestreo, puntos)
% Algortimo Goerzel para la detección
%de frecuencias especificas
%mediante filtrado
for fevaluada=fbuscadas
N=puntos;
k=round(fevaluada*N/fmuestreo);
theta=2*pi*k/N;
realW = cos(theta);
WkN=-exp(-theta*j);

sk1 = 0;
sk2 = 0;
n=1;
    while (n<N)
    sk  = senal(n) +2*realW*sk1 - sk2;
    yk=sk-WkN*sk1;
    sk2 = sk1;
    sk1 = sk;
    n=n+1;
    end
magnitud(fevaluada)=abs(yk);
end
  x=magnitud;
end
```

Figura 2. Código en Matlab del algoritmo Goerzelmio.m

La figura 2 deja ver el código del algoritmo del filtro IIR Goertzel en Matalab 7.6

4. Ejemplo de Implementación en la detección de tonos DMFT

4.1 Generador de tonos DMFT

Los tonos DMFT son la suma de dos funciones cosenoidales de diferentes frecuencias. En la figura 3 se ilustra la combinación de frecuencias sobre un teclado digital. Algunas de las teclas no son de uso común, pero se dejan como referencia ya que pueden usarse con otros propósitos de codificación en aplicaciones de propósito específico. La ecuación 15 muestra como la suma de las funciones cosenoidales generan los tonos DMFT.

$$x(t) = Cos(2\pi f_{Columna}) + Cos(2\pi f_{Fila})$$

Ecuación 15. Función generadora de tonos DMFT

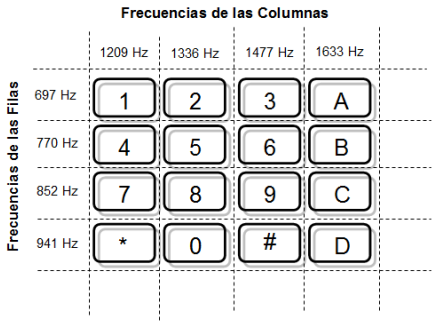

Figura 3. Teclado con combinación de frecuencias para la generación de tonos.

La figura 4 ilustra el detalle de los primeros milisegundos del tono generado por el teclado de la figura 3 para cuando la tecla 1 está presionada. Este ha sido producido haciendo uso de la ecuación 15 con una frecuencia de muestreo fm=8000Hz. De igual forma en la figura 5, se entrega el código en Matlab que permite la simulación de un archivo de audio que se genera, se guarda en formato wav y posteriormente reproduce el archivo del tono.

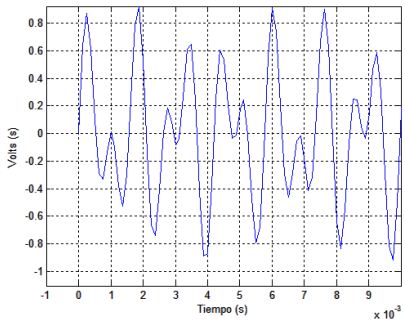

Figura 4. Zoom al tono DMFT generado para la tecla 1.

```
% Generador DTMF. Este produce un tono
% para la tecla múmero 1
clc
fs1=697; %Fila
fs2=1209;%Columnas
% Periodo de alta frecuencia
ts=1/fs2;
% Frecuencia muestreo
fm=8000;
tm=1/fm;
t=[0:tm:ts*1000];
armonico1=sin(2*pi*fs1*t);
armonico2=sin(2*pi*fs2*t);
% Se forza a que la amplitud
% de la señal no pase de -1,+1
tono=armonico1+armonico2;
% Se normaliza el tono a la unidad
tononorm=tono/max(tono+0.1);
% Se guarda el tono en un Wav
wavwrite(tononorm,'miuno.wav');
% Se lee el tono de un Wav
tonor=wavread('miuno.wav');
% Reproduce la muestras sin leer wav
% soundsc(tono);

[sonido_uno, fm, numbits]= wavread('miuno.wav');
muestras=length(sonido_uno);
% Reproduce la senal guardada a
% la frecuencia deseada
wavplay(sonido_uno,fm)
duracion=muestras/fm;
tiempo=linspace(0,duracion,muestras);
% Grafica la Señal del tono
plot(tiempo, tononorm)
xlabel('Tiempo (s)')
ylabel('Volts (s)')
grid
```

Figura 5.

4.2 Ejemplo comparativo del algoritmo propietario de Matlab 7.6 y el desarrollado. [6] [7]

En la primera parte, se implementa el código haciendo uso de la función llamada *goertzelmio.m*. Este ejemplo, busca encontrar las frecuencias de 696 Hz, 770 Hz y 1209 Hz, dos de los cuales corresponden a la tecla 1, dentro de una señal. Si estas son encontradas, no se recibe información sobre magnitud ni su fase, solamente sobre su existencia al presentar un pico ante su presencia. Este hecho hace dramáticamente importante la diferencia entre la FFT (Fast Fourier Transform) y el algoritmo Goertzel.

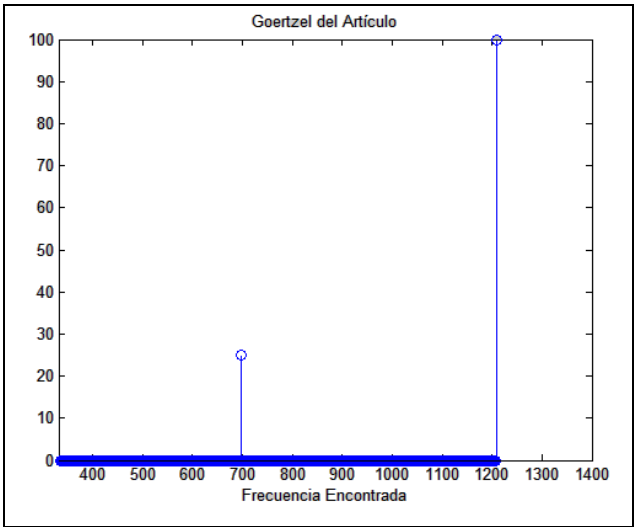

Figura 6. Tonos detectados por el algoritmo Goertzelmio implementado.

La figura 6 permite apreciar los tonos detectados por el algoritmo *goertzelmio.*. Esta gráfica presenta picos solo alrededor de los 700 Hz y 1200 Hz. Si se realiza un zoom de detalle sobres estos puntos, se pueden ver con mejor exactitud la lectura de las frecuencias buscadas.

En este ejemplo se ejecutan los algoritmos *goertzelmio.m* y *goertzel.m.* Como ya se ha mencionado, el primero corresponde a la implementación de los autores de este artículo y el segundo es propietario del *signal processing toolbox* de Matlab 7.6. Las figuras 7a y 7b entregan el código en Matlab.

```
clear all
clc
% Frecuencia de algunos
% de los tonos generados por el teclado (Hz)
%ftonos = [697 770 852 941 1209 1336 1477];
fmuestreo = 8000;
N = 2000;
% Se genera un tono que posee las frecuencias
% 697 Hz y 1209 Hz (Tecla 1).
tonos = sum(cos(2*pi*[697;1209]*(0:N-1)/fmuestreo));
% Se determina la hora del reloj interno del PC
% antes de ejecutar el algoritmo
antes=clock;
% Se ejecuta el algorimo Goertzel
% implementado para este artículo.
% Se implementa el Algorimo para que busque las
% frecuencias de 697 Hz, 770 Hz y 1209 Hz.
fbuscadas=[697,1209 770];
magnitud=goertzelmio(tonos,fbuscadas,fmuestreo, N);
% Se determina la hora del reloj interno del PC
% despues de ejecutar el algoritmo
despues=clock;
% Se mide el tiempo transcurrido durante
% la ejecución del algoritmo
pasomio=despues-antes
% Se normaliza la Salida
magnitud=magnitud/max(magnitud)*100;
figure(1)
stem(magnitud)
xlabel('Frecuencia Encontrada')
title('Goertzel del Artículo')
```

Figura 7a. Ejemplo usando Goerzelmio.

En la segunda parte, del mismo ejemplo, se ejecuta el algoritmo pero usando la función propietaria de Matab.

```
% Ejemplo Goertzel Matlab
% Indices of la DFT para cada Frecuencia DMFT
k = round(fbuscadas/fmuestreo*N);
% Se determina la hora del reloj interno del PC
% antes de ejecutar el algoritmo
antes=clock;
% Se ejecuta el algoritmo Goertzel
% PROPIO de Matlab para que busque las
% frecuencias de 697 Hz y 1209 Hz.
ydft = goertzel(tonos,k+1);
% Se determina la hora del reloj interno del PC
% antes de ejecutar el algoritmo
despues=clock;
% Se mide el tiempo transcurrido durante
% la ejecución del algoritmo
pasomatlab=despues-antes
estim_f = round(k*fmuestreo/N);
figure(2)
stem(estim_f,abs(ydft))
xlabel('Frecuencia Encontrada')
title('Goertzel de Matlab')
```

Figura 7b. Ejemplo usando Goerzel Matlab.

La figura 8 se puede comparar con la figura 6, notándose su similitud al rededor de las frecuencias encontradas.

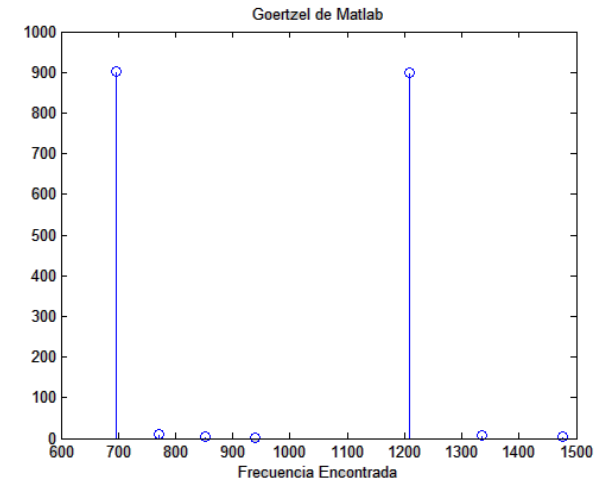

Figura 8. Tonos detectados por el algoritmo Goertzel de Matlab.

Como estudio de caso, se realizó un comparativo del tiempo de ejecución entre el algoritmo Goertzelmio y Goertzel de Matlab 7.6, el cual retornó los valores encontrados en la tabla 1. Los algoritmos se ejecutaron en una computadora portátil con procesador Pentium Dual Core, 2 Gigas en memoria RAM bajo Windows Vista 7.

Algoritmo	Tiempo (s)
Goertzelmio de los Autores	0.0190
Goertzel de Matlab 7.6	0.0470

Tabla 1. Comparativo en tiempo de los algoritmos goerzel.

5. CONCLUSIONES

El algoritmo Goertzel es una forma eficiente de desarrollar aplicaciones que requieran la búsqueda de la respuesta de un sistema ante cierta frecuencia específica. Este no reemplaza, en todos los casos, a la transformada rápida de Fourier (FFT), pero si facilita, de manera notable, su implementación en dispositivos tales como microcontroladores o FPGA´s para la detección de frecuencias de tonos DMFT.

Los autores implementaron una versión del algoritmo, que bajo condiciones que no son de laboratorio, resulta más rápido que el entregado de manera propietaria por Matlab 7.6.

REFERENCIAS

[1] Cortes O, J.A., DEL ANÁLISIS DE FOURIER A LAS WAVELETS ANÁLISIS DE FOURIER, Scientia et Technica Año XIII, No 34, Mayo de 2007. [En línea] disponible en: http://www.utp.edu.co/php/ revistas/ ScientiaEtTechnica /docsFTP /102536151-156.pdf (consultado el 8 de Marzo de 2010)

[2] Goertzel's Algorithm [En línea] disponible en: http://cnx.org/content/m12024/latest/ (consultado el 8 de Marzo de 2010)

[3] Fotis E. Andritsopoulo, AN ACCURATE DTMF DECODER BASED ON THE LOW-COMPLEXITY GOERTZEL ALGORITHM, APRIL, 2001 [En línea] disponible en: http://www.members.tripod.com /dsp_peru /ensayo1.pdf (consultado el 8 de Marzo de 2010)

[4] Matthew D. Felder,,Efficient Dual-Tone Multifrequency Detection Using the Nonuniform Discrete Fourier Transform, IEEE SIGNAL PROCESSING LETTERS, VOL. 5, NO. 7, JULY 1998 [En línea] disponible en: http://www.members.tripod.com /dsp_peru /ensayo2.pdf (consultado el 15 de Octubre de 2009)

[5] Microstar Laboratories, Detecting a single frequency efficiently [En línea] disponible en: http://www.mstarlabs.com /dsp /goertzel /goertzel.html (consultado el 8 de Marzo de 2010)

[6] Texas Instrumenst, Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80 [En línea] disponible en: http://focus.ti.com /lit/an/spra066 /spra066.pdf (consultado el 8 de Marzo de 2010)

[7] The MatlabWorks, Discrete Fourier transform with second-order Goertzel algorithm [En línea] disponible en: http://www.mathworks.com /access /helpdesk /help /toolbox /signal /goertzel.html (consultado el 8 de Abril de 2010)