Proyecto

DTMF

NATALIA MARTÍN YUSTE

JAVIER RUEDA PÉREZ

ALBERTO SÁNCHEZ LÓPEZ

JORGE TORRES DE GABRIEL

ÍNDICE

Introducción

En telefonía, la marcación por tonos también llamada marcación por tonos duales de multifrecuencia, consiste en que cuando un usuario pulsa una tecla de su teléfono correspondiente al dígito que quiere marcar, se envían dos tonos de distinta frecuencia, los tonos como se puede apreciar en la tabla un tono es de baja frecuencia inferior a un 1kHz y el otro tono esta en rango de altas frecuencias entre 1kHz y 2 Khz. Las frecuencias han sido seleccionas con mucha precesión, para que la detección del multitono sea precisa.

HZ	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

En un teléfono convencional no tiene la última columna de la tabla, esta fue usada por los militares para tener prioridad en la línea.

1. DESCRIPCIÓN DETALLADA DE LA DETECCIÓN DE TONOS DUALES DE MULTIFRECUENCIA

Requerimientos DTMF

Las especificaciones ITU Q.24 para la detección DTMF son las siguientes:

- **Tolerancia de la frecuencia:** para que un tono DTMF sea válido debe tener una desviación en frecuencia dentro del 1.5% de tolerancia. Los tonos con una desviación en frecuencia mayor al 3.5% deben ser rechazados. Este requerimiento ayuda a prevenir la detección de falsas señales como tonos DTMF válidos.
- **Duración de la señal:** Un tono DTMF con una duración mínima de 40ms debe ser considerado válido. La duración de la señal no debe ser menor de 23ms (considerado sin funcionamiento).
- Un **espaciado mínimo entre tonos** de 50 ms
- Los dígitos requieren que sean transmitidos en un ratio de menos de 10 por segundo
- **Atenuación de la señal:** El detector debe trabajar con una relación señal-ruido (SNR) de 15db en el peor de los casos y una atenuación de 26dB.
- **Interrupción de la señal:** Una señal DTMF válida interrumpida por un máximo de 10ms, no debe ser detectada como dos tonos distintos.
- **Pausa en la señal:** Una señal DTMF válida separada por una pausa de tiempo de al menos 40ms debe ser detectada como dos tonos distintos.
- **Fase:** El detector debe operar con un máximo de 8dB en fase normal y 4dB en fase invertida.
- **Rechazo al habla:** El detector debe operar en la presencia del habla rechazando la voz como un símbolo DTMF válido.

2. EXPLICACIÓN DETALLADA DEL ALGORITMO DE GOERTZEL

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 Goertzel1 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 emplean actualmente en algunas modernas centrales telefónicas digitales.

2.1 Definiciones

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. 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

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

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

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 :

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

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

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

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:

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

Y desarrollando mediante la identidad de Euler:

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

Se concluye en la ecuación 3

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

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 la 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:

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

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:

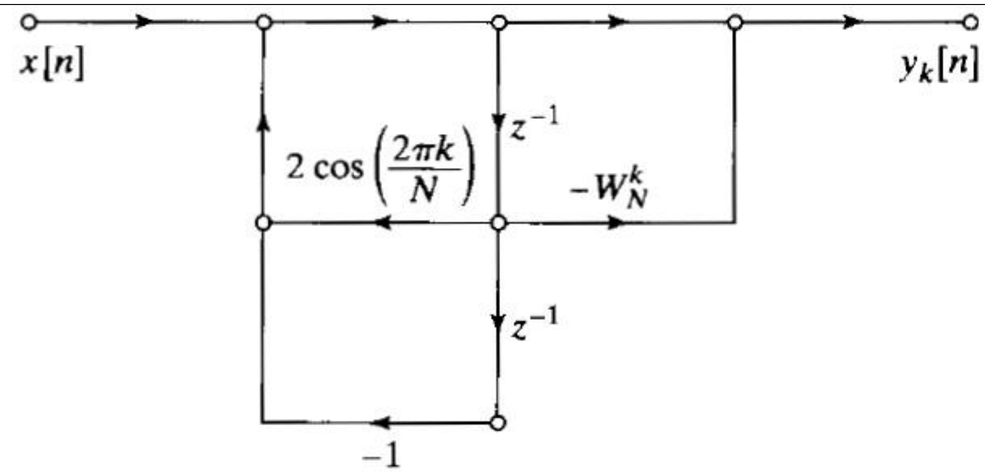


Figura 1. Diagrama del filtro IIR del algoritmo Goertzel de la ecuación 8

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

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

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

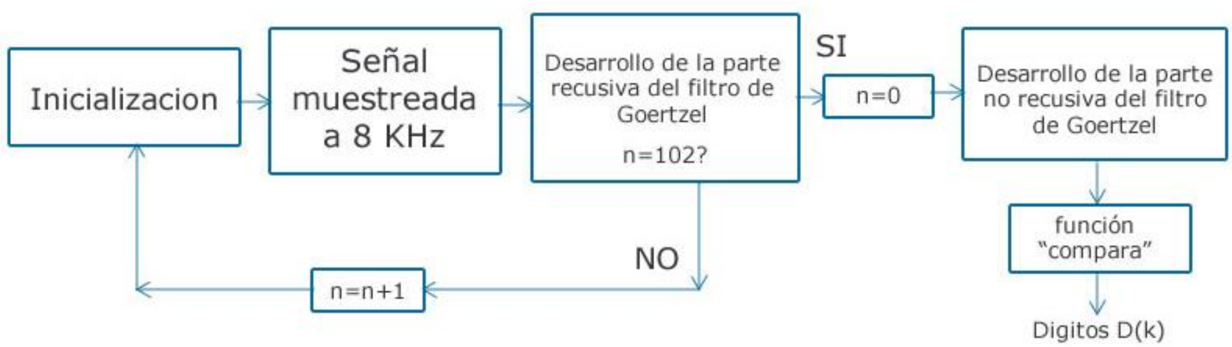
Ecuación 9. Ecuación en diferencias del algoritmo de Goertzel.

3. Desarrollo en Matlab

Para el desarrollo del código, hemos dividido el trabajo en dos funciones y en un programa principal. Las dos funciones las hemos llamado "goertzel" y "compara", y el programa principal le denominamos "busca". A continuación entramos a describir en detalle lo que realiza cada elemento del programa

3.1 "GOERTZEL"

Se trata de la función que desarrolla en Matlab el algoritmo de Goertzel.



Dado que el resultado tendrá una matriz con dimensiones *length(señal)/102* usamos un redondeo de tipo floor, guardándolo en la variable yk.

```
n1 = length(senal); %longitud de la señal original

ntoma102 = floor(n1/102); %numero muestras señal / 102
%enteros más próximos a menores o iguales a A.

yk = (1:ntoma102); %dimension matriz
```

El flujograma lo hemos programado mediante el uso de condiciones iniciales de reposo, y dividiéndolo en dos partes, una recursiva (parte izquierda del flujograma) y una no recursiva (parte derecha del grafo).

El siguiente código calculamos los k-ésimos coeficientes según la frecuencia evaluada, el orden de la DFT y la frecuencia de muestreo. Se introducirán posteriormente como valores de la función.

Para la parte recursiva del flujograma realizamos un bucle de 102 iteraciones de la señal original *senal(n+m)*, a lo cual le sumamos una variable m hasta completar los 102 segmentos en los que ha sido dividida las muestras de la señal.

En cuanto a la parte no recursiva, cada vez que concluye las iteraciones de la parte recursiva, actualizamos el valor de m (*m=m+102*) restamos a A el producto de la memoria M1 por la ventana WkN, guardando cada resultado en la matriz definida anteriormente.

```
%datos para cálculos

k = round(fevaluada*N/fmuestreo);
%el késimo término de la DTF
theta = 2*pi*k/N;
WkN = exp(-theta*sqrt(-1));


for j=1:ntoma102 ;

%iniciamos las condiciones iniciales
       M1=0;
       M2=0;

    %parte recursiva
    for n=1:102 ;

%cuando n=102+m, tomamos A para la parte no recursiva

A=senal(n+m)+2*cos(theta)*M1-M2;
M2=M1;
M1=A;
    end

    %parte no recursiva
    m=m+102;
    yk(j)= (A-M1*WkN);

end


   x = yk;
```

3.2 "COMPARA"

Función que se encarga de deducir los dígitos pulsados a partir de los resultados obtenidos en el algoritmo de Goertzel.

```
sf1=abs(goertzel(senal, 697, puntos));
sf2=abs(goertzel(senal, 770, puntos));
sf3=abs(goertzel(senal, 852, puntos));
sf4=abs(goertzel(senal, 941, puntos));


fa1=abs(goertzel(senal, 1209, puntos));
fa2=abs(goertzel(senal, 1336, puntos));
fa3=abs(goertzel(senal, 1477, puntos));
```

Guardamos en variables el módulo de los resultados obtenidos tras ejecutar el algoritmo de Goertzel a dichas frecuencias. Comparamos las bajas frecuencias con cada una de las posibilidades de alta frecuencia. Si ambos superan el umbral establecido (155000), guardamos en la matriz "D" el valor del digito correspondiente. La matriz D tendrá las mismas dimensiones que yk.

3.3“BUSCA”

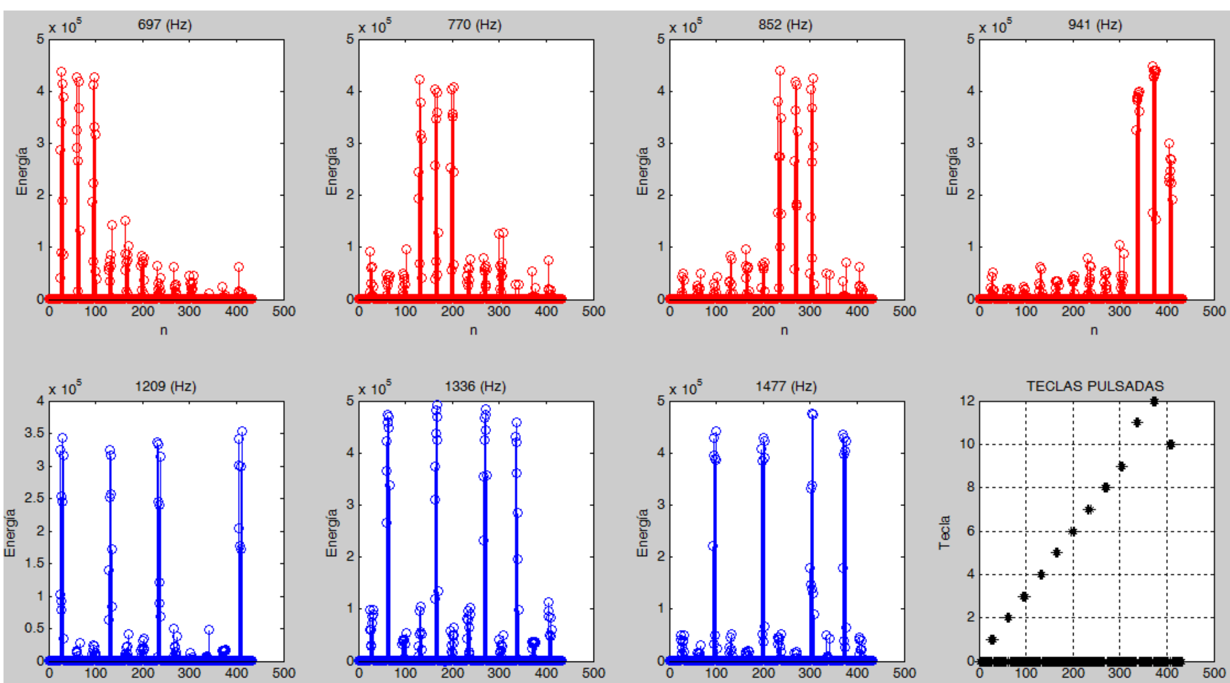
Se encarga de imprimir por pantalla el resultado tras pasar por el algoritmo de Goertzel cada una de las frecuencias y los dígitos de la función “compara”.

En este programa principal, establecemos el orden de la DFT a 102 (como nos indican las especificaciones), para obtener un valor correcto de los k-ésimos coeficientes. La fórmula para el cálculo de k es:

$$k = \frac{\text{fevaluada}}{\text{fmuestreo}} * \text{N}$$

Frecuencia (Hz)	K-ésimos coeficientes
697	8.88 ~ 9
770	9.81 ~ 10
852	10.86 ~ 11
941	11.99 ~ 12
1209	15.41 ~ 15
1336	17.03 ~ 17
1477	18.83 ~ 19

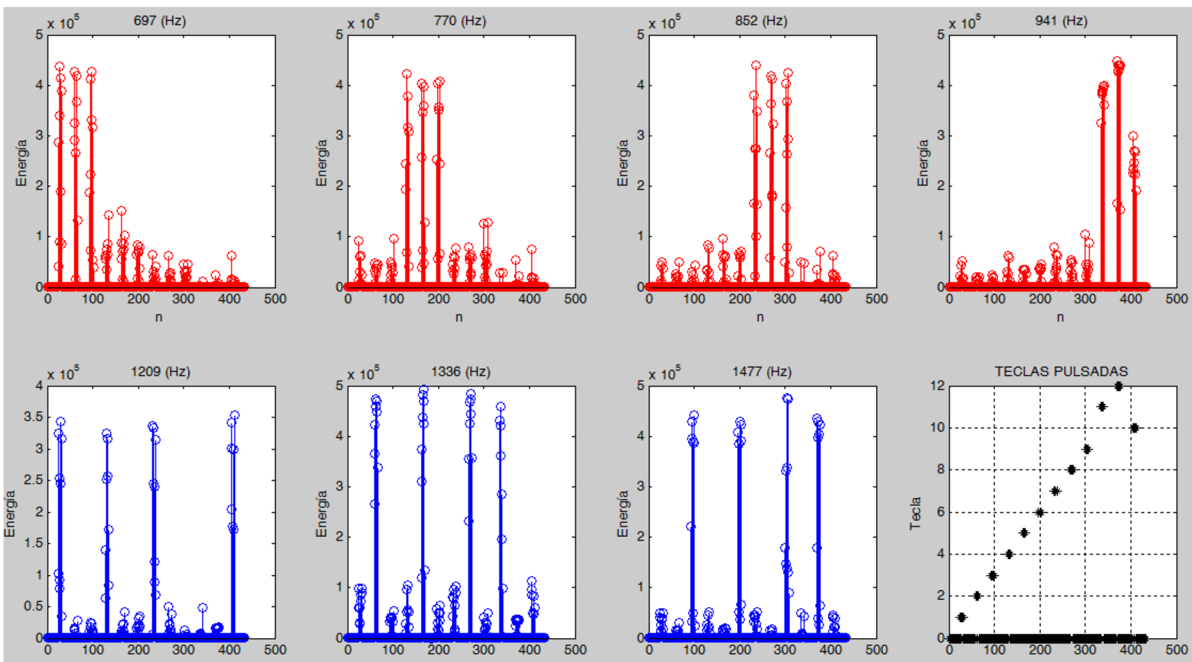
El resultado de ejecutar el programa para TONOS_01 es el siguiente:



4. PRUEBAS Y RESULTADOS OBTENIDOS

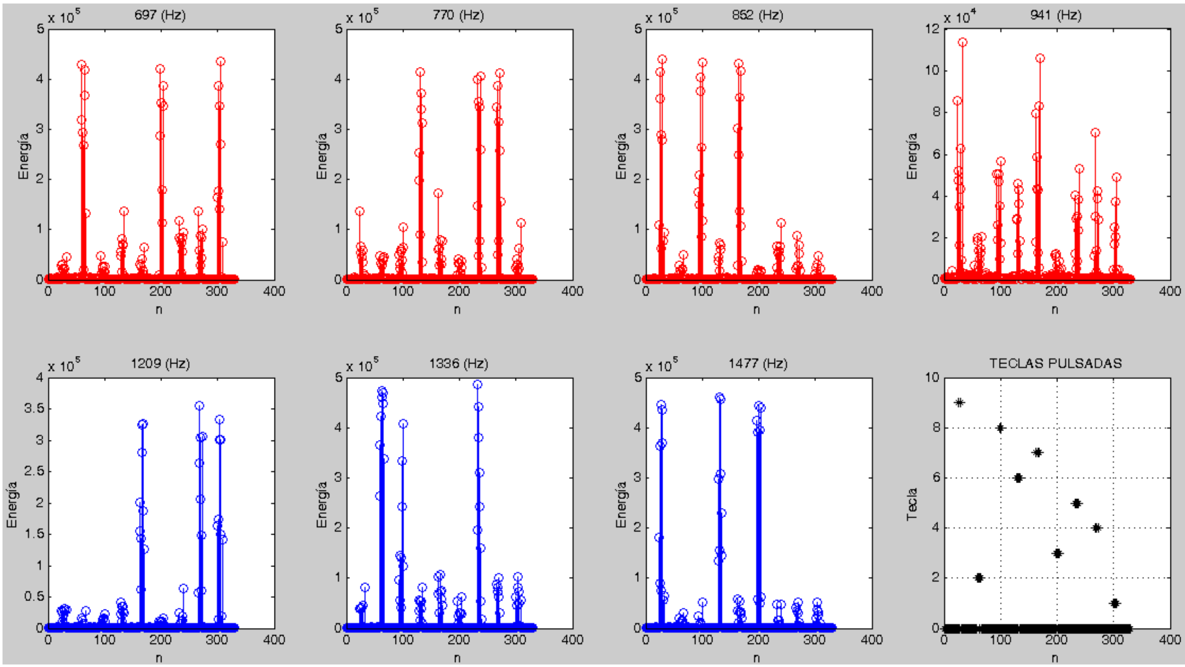
Haciendo uso de todo el código detallado con anterioridad, obtenemos los resultados de cada archivo proporcionado, deduciendo los dígitos pulsados en cada uno. En la gráfica TECLAS PULSADAS, cada valor hace referencia a su tecla correspondiente, menos 0, # y * que toman los valores 11, 12 y 10 respectivamente. Adjuntamos para cada archivo una captura de la ejecución.

4.1 TONOS_01



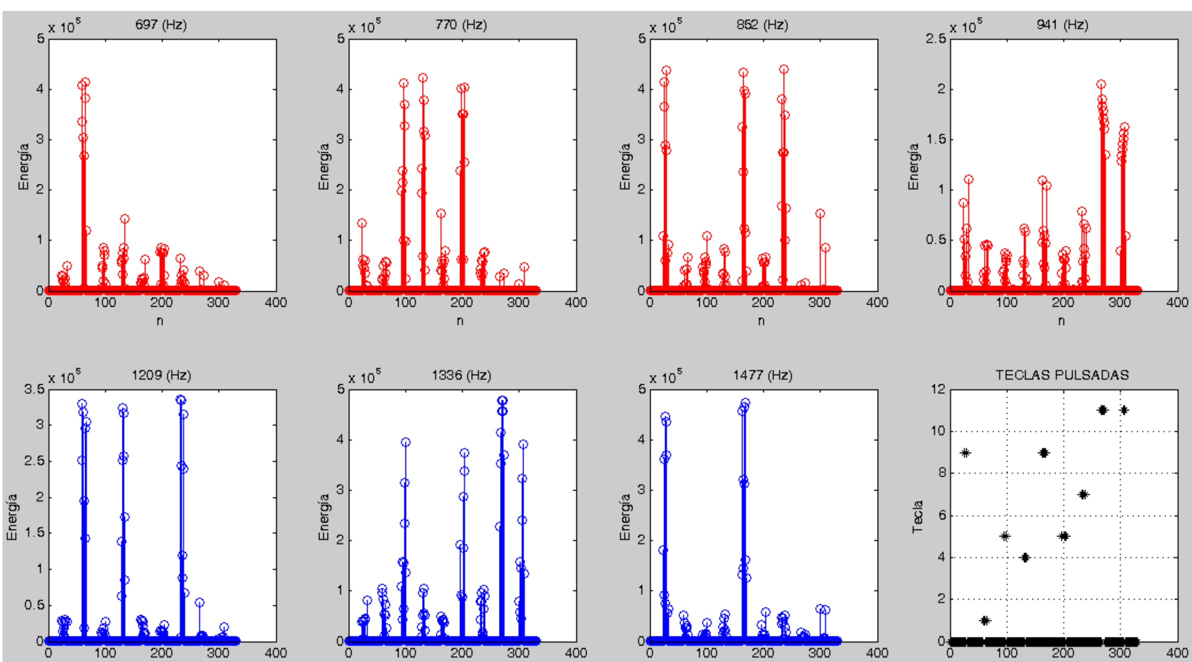
Las TECLAS pulsadas son: 1 2 3 4 5 6 7 8 9 0 # *

4.2 TONOS_02



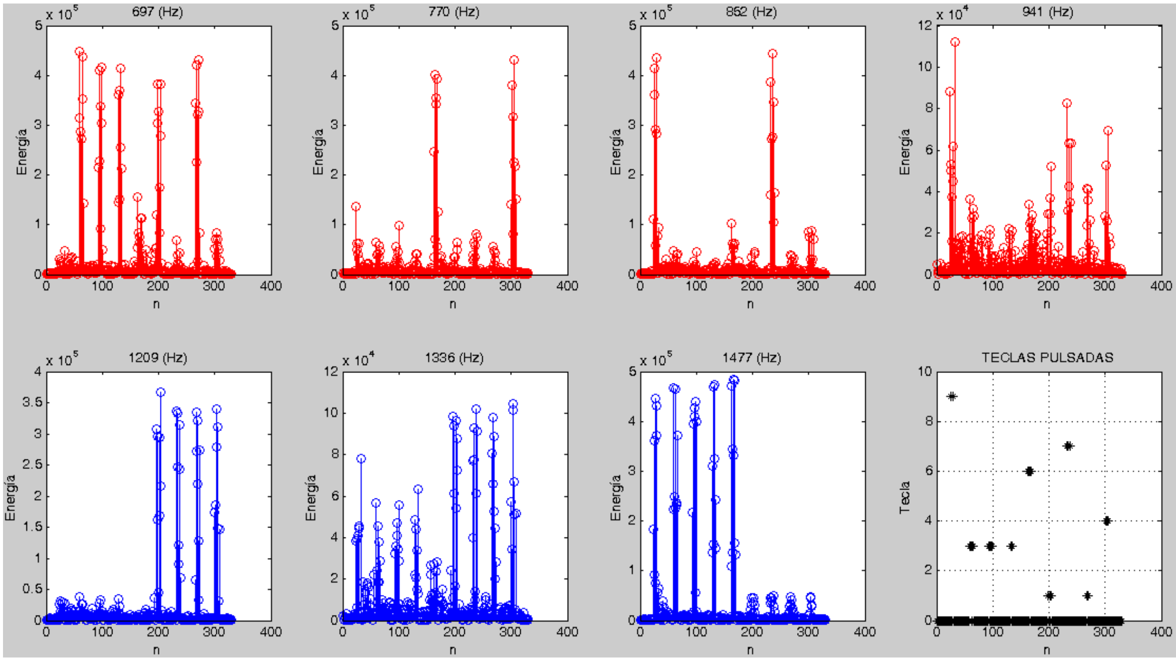
Las TECLAS pulsadas son: 9 2 8 6 7 3 5 4 1

4.3 TONOS_X1



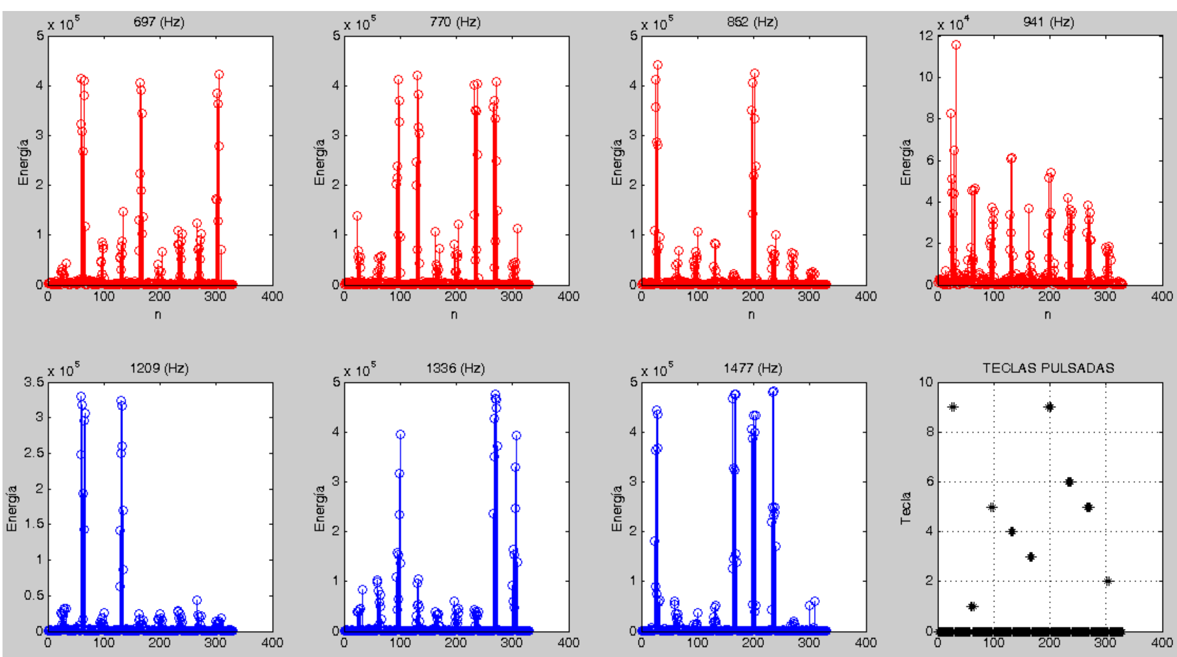
Las TECLAS pulsadas son: 9 1 5 4 9 5 7 0 0

4.4 TONOS_X2



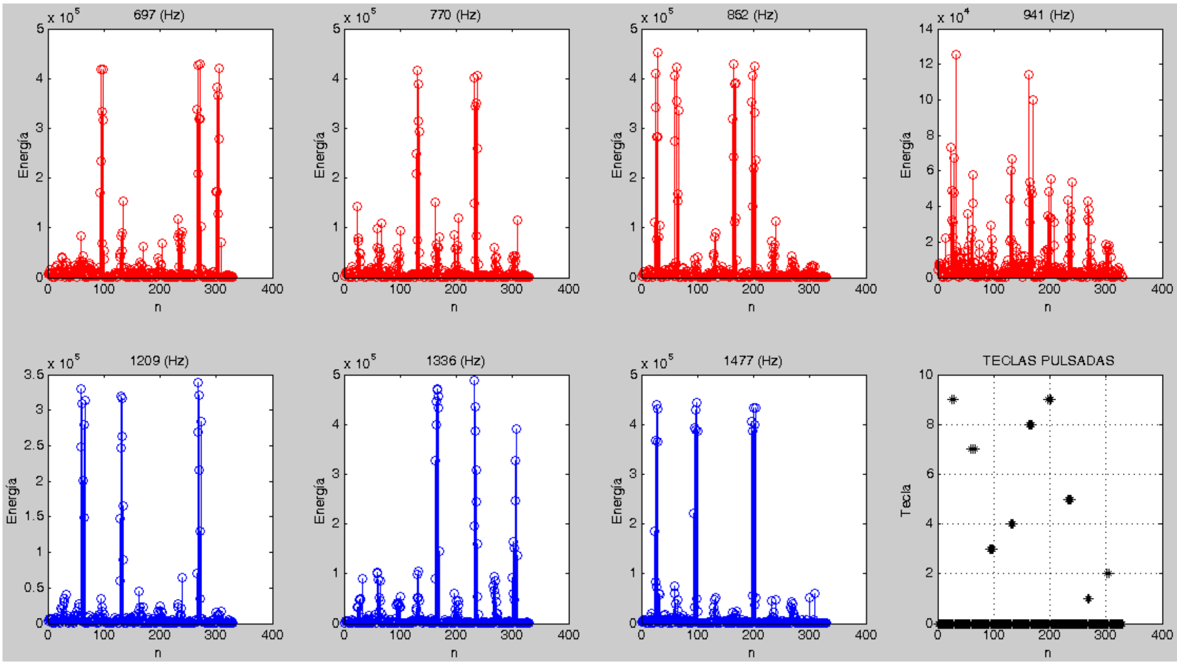
Las TECLAS pulsadas son: 9 3 3 3 6 1 7 1 4

4.5 TONOS_X3



Las TECLAS pulsadas son: 9 1 5 4 3 9 6 5 2

4.6 TONOS_X4



Las TECLAS pulsadas son: 9 7 3 4 8 9 5 1 2

ANEXO

Filtro Paso Banda

Con el objetivo de eliminar el ruido presente en los ficheros proporcionados utilizamos un filtro que deje pasar solo el rango de frecuencias bajas y altas en donde puedan existir posibles tonos DTMF.

Al ejecutar el programa se reproduce la señal original, a continuación se imprime por pantalla el grafico y por último se reproduce la señal filtrada.

Para nuestro objetivo, este filtro no es necesario, ya que sin él, el resultado obtenido ha sido correcto. No obstante, con su uso, podremos eliminar ruido.

```
TONOS_01
fs=8000;
y=dtmf;
sound(y,fs) %señal original

%Filtro de frecuencias bajas
F1=[587 687 951 1051];
delta1=0.000576;
%rizado 0.01 db en banda de paso;
delta2=0.00001;
%atenuacion de 100db
dev=[delta2 delta1 delta2];
fs=8000; A=[0 1 0];
[n,Fo,Ao,w]=firpmord(F1,A,dev,fs);
h1=firpm(n,Fo,Ao,w);

%Filtro de frecuencias altas
F2=[1099 1199 1487 1587];
delta1=0.000576;
%rizado 0.01 db en banda de paso;
delta2=0.00001;
%atenuacion de 100db
dev=[delta2 delta1 delta2];
fs=8000; A=[0 1 0];
[n,Fo,Ao,w]=firpmord(F2,A,dev,fs);
h2=firpm(n,Fo,Ao,w);

h=h1+h2; %suma de ambos filtros
figure
freqz(h,1,1001,8000)
yf=filter(h,1,y);
sound(yf,fs)  %señal filtrada
```