

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



RECONOCIMIENTO DE LOCUTORES

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

RICARDO ANTONIO CASTAÑEDA DE LA ROSA

PROMOCIÓN
1977 - II

LIMA – PERU
2005

RECONOCIMIENTO DE LOCUTORES

DEDICATORIA

Este trabajo esta dedicado a mi familia, madre, esposa e hijos sin cuyo apoyo no hubiese podido realizarlo, ese constante apoyo material y espiritual es el motor de este estudio.

SUMARIO

Este informe enfoca de una manera práctica el tratamiento de una señal de voz, esta señal se toma en forma analógica y se hace un muestreo para posteriormente digitalizarla. Cuando la tenemos en forma vectorial podemos obtener ciertas características en el dominio del tiempo y de la frecuencia a un conjunto de estas características se le denomina patrón de voz, este patrón de voz se puede almacenar en un archivo y lo más importante es que se puede comparar con otros patrones de voz, al realizar esta comparación podemos reconocer al locutor cuyo patrón de voz coincida con otro de los patrones de voz que están almacenados en una base de datos sirviendo así como una forma de reconocimiento individual.

INDICE

INTRODUCCIÓN	01
CAPÍTULO I	
MARCO CONCEPTUAL PROCESAMIENTO DE SEÑALES	03
1.1 Antecedentes del Estudio	03
1.2 La transformada de Fourier	04
1.3 Muestreo de Señales	08
1.4 La Transformada Discreta de Fourier	10
CAPÍTULO II	
TRATAMIENTO DE LA VOZ	16
2.1 Captura de sonido y Formato WAV	16
2.2 Cálculos sobre archivos de voz	18
2.3 Cálculo de patrones de voz.	21
2.4 Programas básicos para señales de audio	23
CAPÍTULO III	
PROGRAMAS PARA RECONOCIMIENTO DE LOCUTORES	26
3.1 Análisis en el dominio del tiempo	26
3.2 Análisis en el dominio de la frecuencia	42
3.3 Comparación de patrones en el tiempo y en la frecuencia	52
3.4 Reconocimiento de locutor	72

CAPÍTULO IV

RESULTADOS	89
4.1 Resultados del estudio comparativo de los patrones de tres archivos nombre.wav del mismo locutor y otro de un locutor diferente.	89
4.2 Resultados del programa que halla el patrón de 5 locutores y del programa que reconoce finalmente al locutor.	97
CONCLUSIONES	99
BIBLIOGRAFIA	100

INTRODUCCIÓN

El principal objetivo de este estudio es reconocer un locutor utilizando su patrón de voz esto lo podemos realizar si consideramos los siguientes pasos:

- Grabar una señal de voz y después digitalizarla.
- Hallar un patrón de la señal de voz.
- Realizar una búsqueda de un patrón de voz comparándolo con otros patrones almacenados, para identificar el locutor que ha emitido la voz.

La idea es lograr identificar un patrón de voz, pero este patrón se puede obtener para prácticamente todas las señales analógicas generadas en el mundo físico. Podemos considerar que los resultados obtenidos se pueden extrapolar para cualquier señal de audio.

La señal de voz tiene un patrón diferente para cada persona (algo así como una huella digital). La tecnología se desarrolla para que los programas interactivos de todo tipo reciban órdenes de voz de algún titular. Un campo importante de los patrones de voz es usarlo como clave para la realización de operaciones bancarias, este hecho parte de que las tarjetas de crédito o de ahorros son muy peligrosas, pues los portadores están evidenciando con su posesión que tienen una cantidad de dinero en algún cajero, de lo cual aprovecha la delincuencia para asaltar al propietario poniendo en peligro su vida. Pero no solamente para manejo de cajeros nos sirve los patrones de voz, sino para ejecutar los distintos programas que en la actualidad modernizan nuestra vida.

Con esta pequeña colaboración esperamos avanzar un mínimo trecho en el desarrollo de esta tecnología.

CAPÍTULO I

MARCO CONCEPTUAL PROCESAMIENTO DE SEÑALES

1.1 Antecedentes del Estudio

El tratamiento digital de las señales de voz es un problema que se ha abordado en los últimos 30 años, siendo una parte importante del procesamiento de señales. Uno de los problemas que nos merecen más atención es el reconocimiento de un locutor. Las señales de voz se estudian en el dominio del tiempo que es donde inicialmente se encuentran almacenadas en nuestros dispositivos informáticos. Para profundizar aun más pasamos estas señales del dominio del tiempo al dominio de la frecuencia donde obtenemos parámetros que nos indican la importancia de cada frecuencia (perteneciente al ancho de banda de cada señal) en un intervalo determinado (ventana). Esta tecnología se basa en la transformada de Fourier.

El tratamiento de las señales de voz es una área donde las técnicas de tratamiento digital de la señal ha dado buenos resultados y todavía se siguen investigando. Las propiedades de las señales de voz son todavía estudiadas extensamente, los algoritmos son diversos y se encuentra mucha literatura al respecto.

Los problemas básicos del tratamiento de voz son:

- El realzado de la voz
- La síntesis
- La codificación digital de la voz
- Reconocimiento de locutor

El reconocimiento de locutor tiene como una base la transformada de Fourier, cuando estudiamos la voz en el dominio de la frecuencia.

1.2 La transformada de Fourier

Para entender la transformada de Fourier, se recorrerá un camino que nos llevará de la serie de Fourier a la transformada de Fourier.

La serie de Fourier nos dice que si $f(t)$ es una función periódica de periodo T , se puede representar mediante la serie trigonométrica.

$$f(t) = (a_0 / 2) + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \text{sen}(n\omega_0 t)) \quad \dots\dots\dots(1.1)$$

Considerando $\omega_0 = 2\pi/T$. Basándonos en que las funciones $\cos n\omega_0 t$ y $\text{sen } n\omega_0 t$ son ortogonales, entonces los coeficientes de Fourier estarán expresados por:

$$a_0 = (2/T) \int_{-T/2}^{T/2} f(t) dt \quad \dots\dots\dots(1.2)$$

$$a_n = (2/T) \int_{-T/2}^{T/2} f(t) \cos(n\omega_0 t) dt \quad \dots\dots\dots(1.3)$$

$$b_n = (2/T) \int_{-T/2}^{T/2} f(t) \text{sen}(n\omega_0 t) dt \quad \dots\dots\dots(1.4)$$

Por ser una función importante mencionamos la función impulso unitario $\delta(t)$,

sabiendo que es expresada como $\delta(t) = \begin{cases} 0, t \neq 0 \\ \infty, t = 0 \end{cases} \quad \dots\dots\dots(1.5)$

también indicamos que $\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad \dots\dots\dots(1.6)$

y $\int_{-\infty}^{\infty} \delta(t) \phi(t) dt = \phi(0) \quad \dots\dots\dots(1.7)$

para llegar a la forma compleja de la serie de Fourier nos basamos en que \cos

$nw_0t = (1/2)(e^{jnw_0t} + e^{-jnw_0t})$ y también que $\text{sen } nw_0t = (1/2j)(e^{jnw_0t} - e^{-jnw_0t})$. De lo

anterior llegamos a $f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jnw_0t}$ (1.8)

Los coeficientes serán expresados por: $c_0 = (1/T) \int_0^T f(t) dt$ (1.9)

$c_n = (1/T) \int_{-T/2}^{T/2} f(t) e^{-jnw_0t} dt$ (1.10)

Utilizando el concepto de Serie compleja de Fourier, hallamos el espectro de frecuencias (Valores de c_n en cada nw_0), de un tren de pulsos rectangulares de alto “A” y ancho “d” y con periodo T, este será:

$c_n = (Ad/T) \text{sen}(nw_0d/2) / (nw_0d/2) = (Ad/T) \text{Sa}(nw_0d/2)$ (1.11)

Esta función es llamada función de muestreo $\text{Sa}(x) = \text{sen}(x)/x$ finalmente vemos la serie compleja de Fourier para un tren de impulsos unitarios

$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) = (1/T) \sum_{n=-\infty}^{\infty} e^{jnw_0t}$ (1.12)

Después de haber entendido la serie de Fourier ahora vamos a llegar a la transformada de Fourier considerando que el periodo de la serie se vuelve muy grande ($T \rightarrow \infty$), a sea se hace $f(t)$ no periódica. De (1.8) y (1.10) tenemos

$f(t) = \sum_{n=-\infty}^{\infty} \left\{ (1/T) \int_{-T/2}^{T/2} f(t) e^{-jnw_0t} dt \right\} e^{jnw_0t}$ (1.13)

expresamos (1.13) de la forma $f(t) = \sum_{n=-\infty}^{\infty} \left\{ (1/2\pi) \int_{-T/2}^{T/2} f(x) e^{-jnw_0x} dx \right\} w_0 e^{jnw_0t}$ como

$T \rightarrow \infty$ y $w_0 = \Delta w$ también $nw_0 = n\Delta w \rightarrow w$, entonces (1.13) se convierte en

$f(t) = \sum_{n=-\infty}^{\infty} \left\{ (1/2\pi) \int_{-T/2}^{T/2} f(x) e^{-jwx} dx \right\} e^{jw t} \Delta w$ (1.14)

Llevando esta expresión al límite $T \rightarrow \infty$, $\Delta w \rightarrow dw$ y $\sum \rightarrow \int$ llegamos a

$$f(t) = (1/2\pi) \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \right) e^{j\omega t} d\omega \quad \dots\dots\dots(1.15)$$

De (1.15) tomamos la parte entre los paréntesis y la llamamos la transformada de

$$\text{Fourier (F(w)). } F(w) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad \dots\dots\dots(1.16)$$

y la transformada inversa de Fourier será entonces de la misma $\dots\dots\dots(1.15)$

$$f(t) = (1/2\pi) \int_{-\infty}^{\infty} F(w) e^{j\omega t} d\omega \quad \dots\dots\dots(1.17)$$

Como un primer ejemplo de aplicación hallamos la transformada de Fourier de un pulso rectangular $p_d(t)$, $p_d(t) = 1$, si $|t| < 1/2$ y $p_d(t) = 0$, si $|t| > 1/2$, entonces por

$$(1.16) F(w) = \int_{-\infty}^{\infty} p_d(t) e^{-j\omega t} dt = \int_{-d/2}^{d/2} 1 \cdot e^{-j\omega t} dt \text{ de donde llegamos a:}$$

$$F(w) = d \text{ sen}(w d/2) / (w d/2) \quad \dots\dots\dots(1.18)$$

Ahora veremos algunas **Propiedades de la Transformada de Fourier.**

- $\mathfrak{F} [a_1 f_1(t) + a_2 f_2(t)] = a_1 F_1(w) + a_2 F_2(w)$
- $\mathfrak{F} [f(a t)] = (1/|a|) F(w/a)$
- $\mathfrak{F} [f(-t)] = F(-w)$
- $\mathfrak{F} [f(t-t_0)] = F(w) e^{-j\omega t_0}$ y $\mathfrak{F} [f(t) e^{-j\omega t_0}] = F(w-w_0)$
- $\mathfrak{F} [f(t) \cos \omega_0 t] = (1/2) F(w-\omega_0) + (1/2) F(w+\omega_0)$
- Si $\mathfrak{F} [f(t)] = F(w)$ entonces $\mathfrak{F} [F(t)] = 2\pi f(-w)$
- $\mathfrak{F} [f(t)'] = jw F(w)$ y $\mathfrak{F} \left[\int_{-\infty}^t f(x) dx \right] = (1/jw) F(w)$

Para el análisis de señales digitales necesitamos conocer el concepto

de *Convolución*, está definida por $f(t) = \int_{-\infty}^{\infty} f_1(x) f_2(t-x) dx = f_1(t) * f_2(t)$.

La *Convolución* de una función $f(t)$ con la función impulso unitario

$$f(t) * \delta(t) = f(t) \quad \dots\dots\dots(1.19)$$

Relacionando la T.F. con el concepto de *Convolución* tenemos:

$$\text{Convolución en el tiempo } \mathfrak{F}[f_1(t) * f_2(t)] = F_1(w) F_2(w) \quad \dots\dots\dots(1.20)$$

$$\text{Convolución en la frecuencia } \mathfrak{F}[f_1(t)f_2(t)] = (1/2\pi) F_1(w)* F_2(w) \quad \dots\dots\dots(1.21)$$

También vamos a considerar las transformadas de algunas funciones que podríamos utilizar mas adelante.

$$\text{T.F. de la función impulso } \mathfrak{F}[\delta(t)] = 1 \quad \dots\dots\dots(1.22)$$

$$\text{T.F. de la función } e^{jw_0t} \quad \mathfrak{F}[e^{jw_0t}] = 2\pi \delta(w - w_0) \quad \dots\dots\dots(1.23)$$

$$\text{T.F. de la función escalón unitario } \mathfrak{F}[u(t)] = \pi\delta(w) + 1/jw \quad \dots\dots\dots(1.24)$$

Ahora podemos desarrollar la transformada de Fourier de un tren de impulsos

unitarios. Por definición $\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ y usando (1.12) llegamos

$\delta_T(t) = (1/T) \sum_{n=-\infty}^{\infty} e^{jnw_0t}$ Entonces por (1.12) y (1.25), tenemos la T.F. de $\delta_T(t)$ será

$$\mathfrak{F}[\delta_T(t)] = (1/T) \sum_{n=-\infty}^{\infty} \mathfrak{F}[e^{jnw_0t}] = (2\pi/T) \sum_{n=-\infty}^{\infty} \delta(w - nw_0) \quad \dots\dots\dots(1.25)$$

$$\text{como } w_0 = 2\pi/T, \text{ entonces } \mathfrak{F}[\delta_T(t)] = w_0 \delta_{w_0}(w) \quad \dots\dots\dots(1.26)$$

Entonces la T.F. de un tren de pulsos rectangulares de ancho “d” y periodo “T”

$$F(w) = (2\pi/T) \sum_{n=-\infty}^{\infty} Sa(n\pi d/T) \delta(w - nw_0) \quad \dots\dots\dots(1.27)$$

1.3 Muestreo de Señales

Para nuestro caso tomaremos una señal de voz, esta señal es inmediatamente muestreada y convertida a tiempo discreto. Aquí vamos a considerar el teorema del muestreo sobre una señal $f(t)$ de banda limitada (de $-w_m$ a w_m) . Al muestrear una señal esta es multiplicada por un tren de impulsos, o sea $f(t)$ es multiplicada por $\delta_T(t)$.

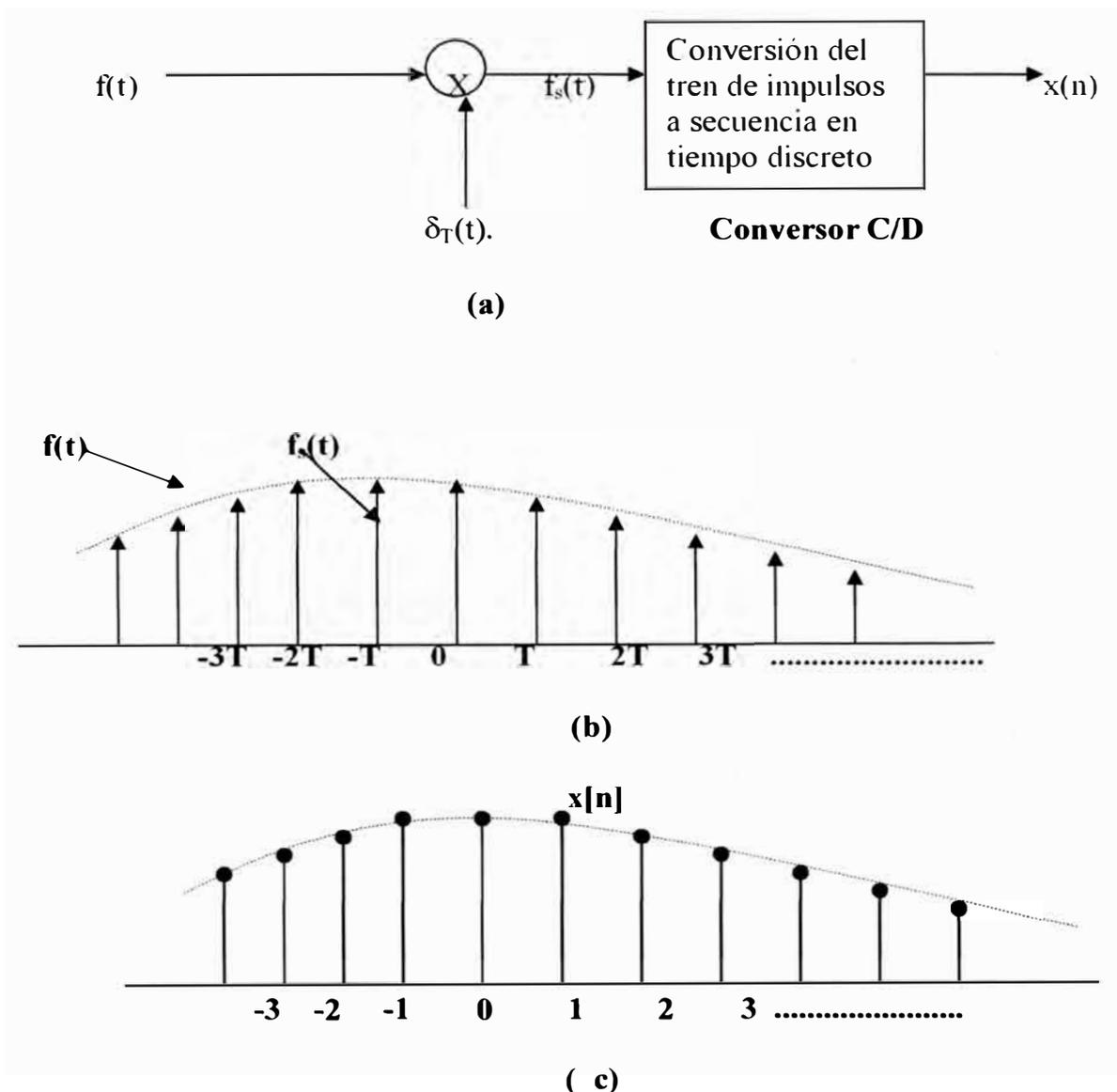


Figura 1.1.(a) Esquema del proceso de muestreo (b) Efecto de multiplicarla por $\delta_T(t)$. (c) Señal discretizada

Este proceso en la frecuencia lo mostramos en la figura 1.2

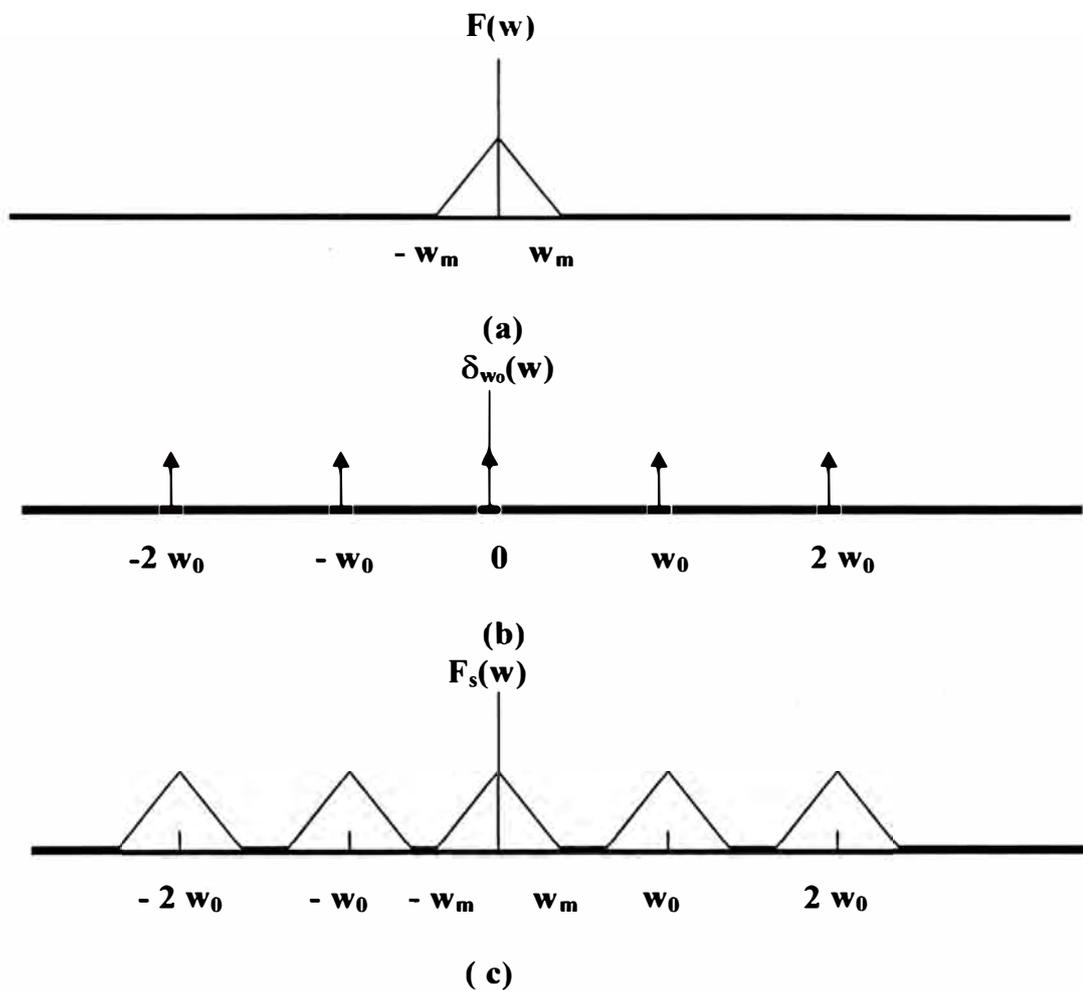


Figura 1.2(a) Señal de banda limitada **(b)** Tren de impulsos
(c) Señal Muestreada

$$\text{Por (1.26) tenemos que } \mathfrak{F}[\delta_T(t)] = w_0 \delta_{w_0}(w) = w_0 \sum_{n=-\infty}^{\infty} \delta(w - nw_0) \quad \text{.....(1.28)}$$

Aplicamos el concepto de Convolución en frecuencia tenemos:

$$\mathfrak{F}[fs(t)] = F_s(w) = (1/2\pi)[F(w) * w_0 \delta_{w_0}(w)] \quad \text{.....(1.29)}$$

A $F(w)$ lo vemos en la fig. 1.2.(a) y a $w_0 \delta_{w_0}(w)$ lo vemos en la fig. 1.2.(b).

$$\text{Operando tenemos } F_s(w) = (1/T)[F(w) * w_0 \delta_{w_0}(w)]$$

$$\begin{aligned}
&= (1/T) \sum_{n=-\infty}^{\infty} F(\omega) \delta(\omega - n\omega_0) \\
&= (1/T) \sum_{n=-\infty}^{\infty} F(\omega - n\omega_0) \quad \dots\dots\dots(1.30)
\end{aligned}$$

La figura 1.2.(c) nos muestra lo obtenido en (1.30), como hemos dicho anteriormente $f(t)$ es de banda limitada, entonces de acuerdo a esto para tener una señal sin solapamientos necesariamente $\omega_0 \geq 2\omega_M$ (criterio de Nyquist).

1.4 La Transformada Discreta de Fourier

Como vamos a aplicar todos nuestros algoritmos sobre una muestra de voz, en nuestro caso 1 dígito o más, que son en realidad secuencias de duración finita. Utilizaremos una nueva nomenclatura $x[n]$, para secuencia de voz en el tiempo y $X[k]$ para secuencia en la frecuencia.

Iniciamos la transformada discreta de Fourier (DFT) considerando la serie discreta de Fourier. Tenemos una secuencia $x_p[n]$ que es periódica de periodo N o sea $x_p[n] = x_p[n + rN]$. Entonces la serie de Fourier discreta (DSF) se representara como:

$$x_p[n] = \sum_{k=0}^{N-1} X_p[k] e^{j(2\pi/N)kn} \quad \dots\dots\dots(1.31)$$

Donde “N” es el numero de exponenciales requeridos y $X_p[k]$ es el coeficiente del termino k 'esimo. Para hallar $X_p[k]$ como en el caso continuo utilizamos el

concepto de ortogonalidad y llegamos a $X_p[k] = \sum_{n=0}^{N-1} x_p[n] e^{-j(2\pi/N)kn} \quad \dots\dots\dots(1.32)$

Para facilitar la escritura hacemos $W_N = e^{-j(2\pi/N)}$. Entonces tendremos (1.31) y (1.32) escrito así:

$$X_p[k] = \sum_{n=0}^{N-1} x_p[n] W_N^{kn} \quad \dots\dots\dots(1.33)$$

$$x_p[n] = \sum_{k=0}^{N-1} X_p[k] W_N^{-kn} \quad \dots\dots\dots(1.34)$$

Tenemos entonces la pareja de la Serie discreta de Fourier (DSF), aplicando estos conceptos hallamos la DSF para un tren de impulsos periódicos.

$$\sum_{r=-\infty}^{\infty} \delta[n - rN] = (1/N) \sum_{k=0}^{N-1} e^{j(2\pi/N)kn} \quad \dots\dots\dots(1.35)$$

La Convolución se define como $x_{3p} = \sum_{m=0}^{N-1} x_{1p}[m]x_{2p}[n - m] \quad \dots\dots\dots(1.36)$

A continuación indicamos algunas propiedades de la DSF.

➤ $ax_{1p}[n] + bx_{2p} \leftrightarrow aX_{1p}[k] + bX_{2p}[k]$

➤ $X_p[n] \leftrightarrow Nx_p[-k]$

➤ $x_p[n - m] \leftrightarrow W_N^{kn} X_p[k]$

➤ $W_N^{-ln} x_p[n] \leftrightarrow X_p[k - l]$

➤ $\sum_{m=0}^{N-1} x_{1p}[m]x_{2p}[n - m] \leftrightarrow X_{1p}[k]X_{2p}[k]$

➤ $x_{1p}[n]x_{2p}[n] \leftrightarrow (1/N) \sum_{l=0}^{N-1} X_{1p}[l]X_{2p}[k - l]$

Ahora calculamos la transformada discreta de Fourier (TDF), considerando una secuencia $x[n]$ de longitud finita de N muestras, de forma que $x[n] = 0$ para $0 \leq n \leq N-1$ (Recordar que $x_p[n]$ es periódica. Escogemos entonces los coeficientes de un periodo asociados a la secuencia de un periodo, estos coeficientes de un periodo $X[k]$ que se relacionan con $X_p[k]$ por:

$X[k] = X_p[k]$ para $0 \leq k \leq N-1$ y $X[k]$ será 0 en el resto $\dots\dots\dots(1.37)$

Entonces las expresiones siguientes serán la DFT.

$$X[k] = \sum_{n=0}^{N-1} x[n] W^{kn} \quad \dots\dots\dots(1.38)$$

$$x[n] = (1/N) \sum_{k=0}^{N-1} X[k] W_N^{-kn} \quad \dots\dots\dots(1.39)$$

Con la restricción que $X[k] = 0$ fuera del intervalo $0 \leq k \leq N-1$ y que $x[n] = 0$ fuera del intervalo $0 \leq n \leq N-1$.

Para los cálculos computacionales se utiliza la transformada rápida de Fourier (FFT). Teniendo en cuenta que $W_N^{-kN} = e^{-j(2\pi/N)Nk} = e^{j2\pi k} = 1$ por ser producto de la periodicidad.

Basado en esto se estructuran las secuencias de cálculo (algoritmo) sobre las entradas como subsecuencias más pequeñas y estas en subsecuencias más pequeñas entonces tenemos un DIEZMADO en el tiempo y si estructuramos las secuencias de cálculo de salida en subsecuencias más pequeñas entonces tenemos un DIEZMADO en frecuencia.

Basado en estos algoritmos se desarrolla la transformada rápida de Fourier (FFT) para calcular la DFT. La FFT la utilizaremos en la parte de programación.

Digitalización de las señales de voz.- En un **sistema de tiempo continuo**, las señales de entrada/ salida son continuas. En un **sistema de tiempo discreto**, las señales están representadas en tiempo discreto. En un **sistema digital**, las señales de entrada y salida son digitales. En el **Procesamiento Digital de Señales**, las señales son discretas tanto en amplitud como en el tiempo y pueden realizarse tanto en software como en Hardware, nuestra aplicación esta basada en el software (MATLAB).

Las señales de voz las digitalizamos y las almacenamos en archivos cuyas características mostraremos en el siguiente capítulo o sea en nuestro caso un dígito hablado por un locutor se almacena como un vector. La voz dígito (t) se transforma en $x[n]$ con N muestras donde “ N ” es del orden de 3,000 a 10,000.

Sabemos que $x[n]$ varía en el tiempo pero al aplicarle la FFT pasamos al dominio de la frecuencia y generamos $X[k]$ donde K siendo “ K ” número de elementos del vector X generalmente será 512, estas señales de voz digitalizadas se tratarán en el dominio del tiempo y de la frecuencia.

- **En el dominio del tiempo:** Vamos a empezar realizando el estudio en el dominio del tiempo por ser más sencillo. Existen algunos métodos útiles en el dominio del tiempo. Debido a la naturaleza cambiante de la voz en el tiempo se aplica el análisis a porciones de voz. A cada porción de la señal de voz se le asigna una ventana, en la figura 1200 muestras.

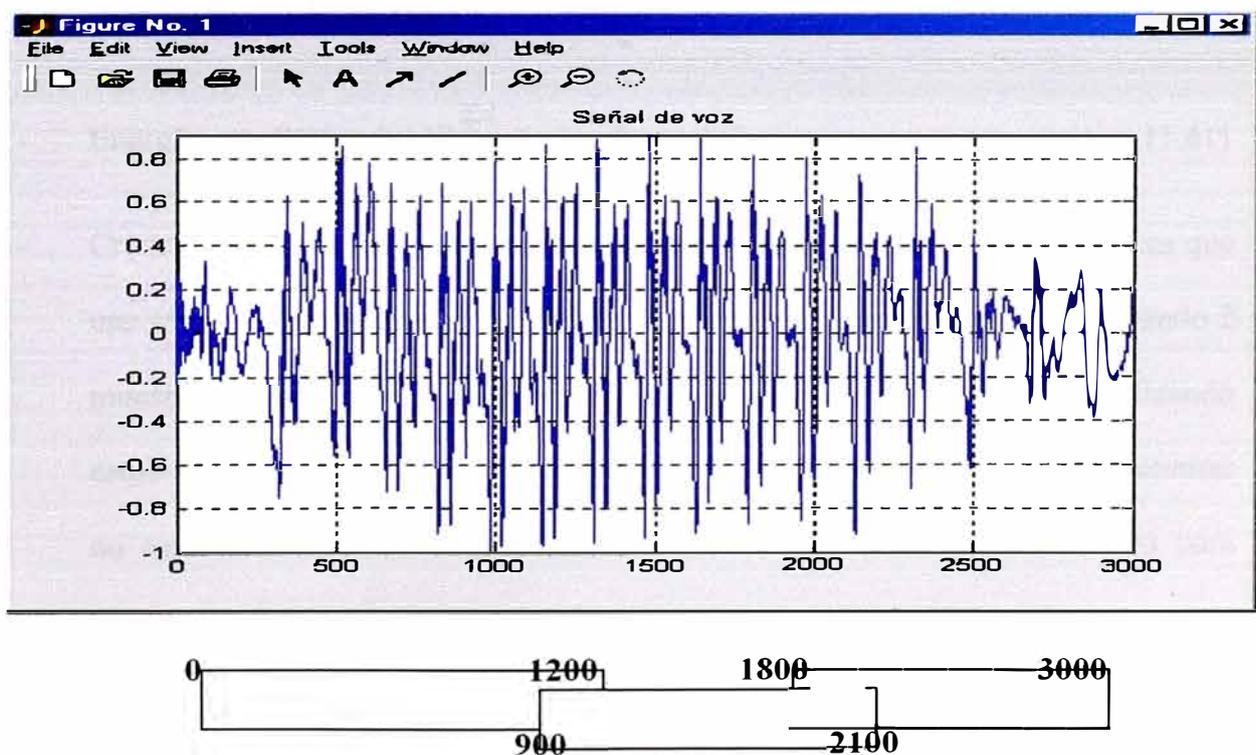


Figura 1.3 Enventanado de la señal de voz para su mejor estudio

Al utilizar el solapamiento desmejoramos los tiempos de respuesta pero proporciona una mejor calidad en los resultados obtenidos. Tenemos tres tipos de ventanas más comunes:

- Rectangulares $w(t) = 1, 0 \leq t \leq L-1$, y $w(t) = 0$, en el resto de la señal Hamming
 $w(t) = 0.54 - 0.46 * \text{coseno}(2\pi t / L), 0 \leq t \leq L-1$
 $w(t) = 0$, en el resto de la señal
- Hanning $w(t) = 0.5 - 0.5 * \text{coseno}(2\pi t / L), 0 \leq t \leq L-1$
 $w(t) = 0$, en el resto de la señal

Para cualquier ventana, su duración determina la cantidad de cambios que se podrán obtener.

En los tiempos las características más comunes son:

- Energía y magnitud:
- Magnitud : $M(n) = (1/N) \sum_{m=0}^{N-1} |x(m)|^p w(n-m)$ (1.40)
- Energía $E(n) = (1/N) \sum_{m=0}^{N-1} x(m)^2 w(n-m)$ (1.41)
- Cruces por cero y máximos: Los cruces por cero indican el número de veces que una señal continua toma el valor cero. Para las señales discretas ocurre cuando 2 muestras consecutivas difieren en signo, o una muestra es nula. Contabilizando estos cruces por cero tenemos una idea de la señal de voz. Podemos encontrar un estimador alternativo contabilizando los máximos. Esto es adecuado para diferenciar los sonidos sonoros del resto de la señal.

- **En el dominio de la frecuencia:** La voz se trata principalmente en el dominio de la frecuencia, llamado también análisis espectral, sabiendo que la voz normalmente se analiza a bajas frecuencias de 4 a 300 Hz. Vamos a empezar realizando el estudio en el dominio del tiempo por ser mas sencillo

CAPÍTULO II TRATAMIENTO DE LA VOZ.

2.1 Captura de sonido y Formato WAV.-

La captura del sonido se realiza mediante el computador y se almacena en memoria o disco. Una tarjeta conocida para esta tarea es la Sound Blaster de Creative.

El proceso consiste:

- 1:** El sonido se trasmite por el aire hasta el micrófono, se transforma la señal acústica en niveles eléctricos, la calidad del micrófono es importante.
 - 2:** La señal analógica pasa a través de un convertidor analógico / digital. Este resulta ser el elemento más importante, sus características establecen la frecuencia máxima a la que se puede trabajar y el número de bits que se utilizan para cuantificar cada muestra.
 - 3:** Mediante un canal de DMA se copia cada muestra a la memoria principal del computador. Posteriormente se podrá copiar el total de las muestras en disco en algún formato estándar de sonido.
- **Formato WAV.-** Describiremos el formato de un archivo WAV de Microsoft. El archivo esta formado por dos bloques: la cabecera tiene un tamaño de 44 bytes y contiene el tipo y organización de las muestras. Su contenido es el siguiente:
 - **Campo 1 :** bytes 0..3. Contiene la palabra “RIFF” en código ASCII.

- **Campo 2** : bytes 4..7. Tamaño total del archivo en bytes menos 8 (no incluye los 2 primeros campos)
- **Campo 3** : bytes 8..15. Contiene la palabra “WAV fmt” en código ASCII.
- **Campo 4** : bytes 16..19. Formato: para PCM vale 16
- **Campo 5** : bytes 20..21 Formato : para PCM vale 1.
- **Campo 6** : bytes 22..23. Indica si es mono(1) o estéreo(2).
- **Campo 7** : bytes 24..27. Frecuencia de muestreo, puede valer: 11.025, 22.050 o 44.100.
- **Campo 8** : bytes 28..31. Número de bytes por segundo que se debe intercambiar con la tarjeta de sonido para una grabación o reproducción.
- **Campo 9** : Bytes 32..33. Numero de Bytes por captura puede ser 1,2 o 4.
- **Campo 10** : Bytes 34..35. Numero de Bytes por muestra, es 8 o 16.
- **Campo 11** : Bytes 36..39. Contiene la palabra “data” en código ASCII.
- **Campo 12** : Bytes 40..43. Nro.total de Bytes que ocupan las muestras.

El registro tendrá la siguiente forma:

Type	tcab WAV = record	{campos que forman una cabecera WAV}
Riff	: array [0..3] of char;	{ “Riff”}
Tamanyo	: longint;	{tamaño del fichero menos 8}
Wave	: array[0..7] of char;	{“WAVEfmt”}
Format1	: longint;	{Formato para PCM vale 16}
Format2	: integer;	{Formato para PCM vale 1 }
Mono	: integer;	{canales: mono 1, estéreo 2}
Frec	: longint;	{Frecuencia de muestreo (hz)}
Bytes_s	: longint;	{Numero de Bytes por segundo}

```

Bytes_c   : integer;           {Numero de Bytes por captura }
Bits m    : integer;           {Numero de bits por muestra   }
Data      : array [0..3] of char; {“data”}
Bytes_muestras_fich : longint;  {Número de Bytes de las muestras}
end;

```

2.2 Cálculos sobre archivos de voz

La primera tarea es aprovechar al máximo los bits utilizados para la codificación de las muestras. Si saturamos en el rango asignado distorsionamos el sonido original disminuyendo su calidad, si el rango es pequeño perdemos información, lo correcto se muestra en la figura 2.1(c).

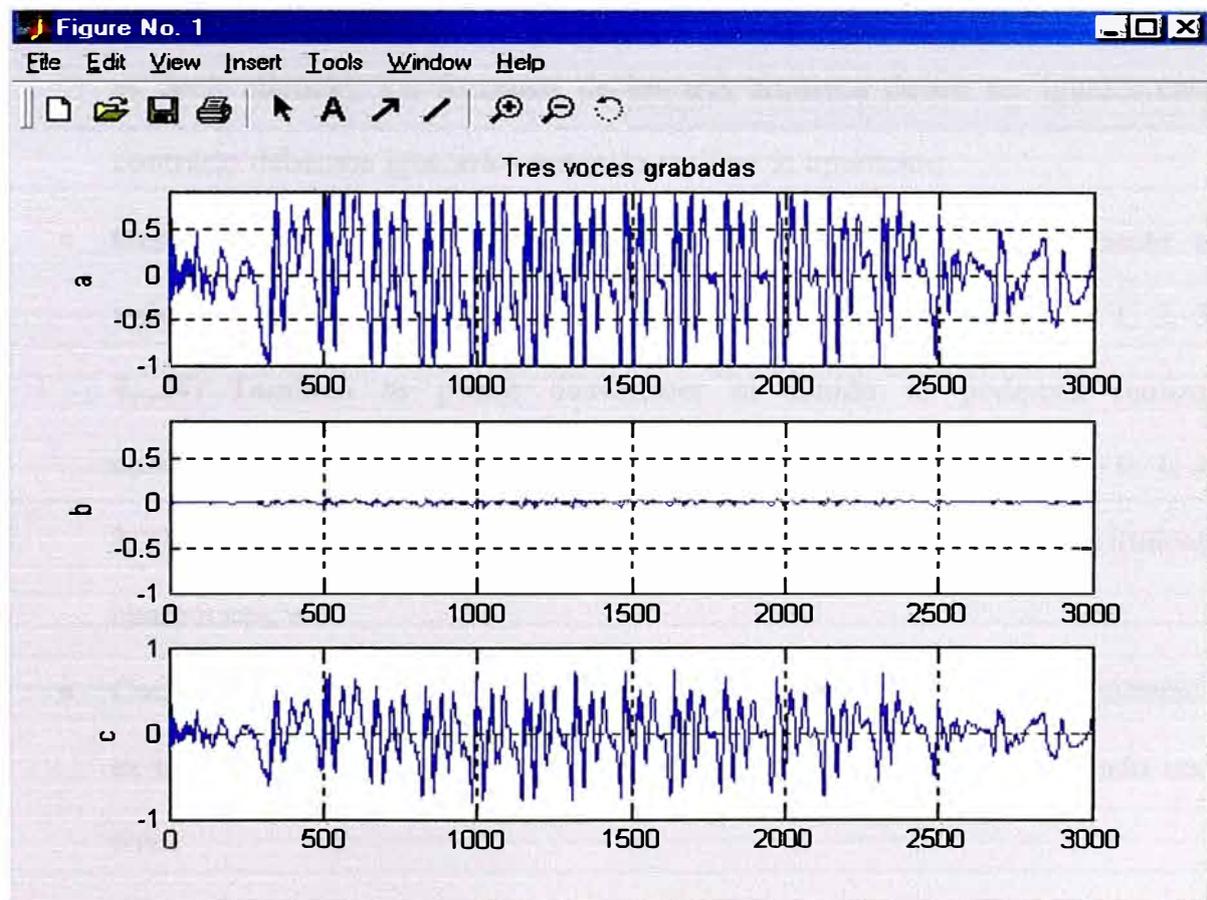


Fig.2.1 (a) Señal saturada de voz (b) Señal con rango muy pequeño (c) Señal correcta para hacer un estudio sobre ella

- **Modificación del volumen:** Podemos multiplicar las muestras por un factor, si este es mayor que 1 aumentamos el volumen, si es menor que 1 disminuimos el volumen. Para obtener una buena señal debemos buscar la muestra de mayor valor y calcular un factor para convertirla en el mayor valor que permita el rango. Después multiplicamos todas las muestras por ese factor.
- **Suma de Sonidos:** Si queremos sumar varios archivos de sonidos debemos sumar sus archivos en forma ordenada. Para ello debemos multiplicar cada sonido por un factor menor que la unidad, la suma de los factores debe ser igual a la unidad. Los sonidos van a sonar en el mismo instante. Debemos considerar que cuando no existen muestras de un sonido debemos tomar ceros es decir silencio, los formatos de los tres archivos deben ser iguales caso contrario debemos igualarlos antes de realizar la operación
- **Desvanecer e intensificar:** Se puede ir intensificar paulatinamente el volumen esto se realiza haciendo: $(\text{muestra}(i) * i) / N$ (donde $i = 0, 1, 2, 3, 4, \dots, N$) También se puede desvanecer el sonido lo podemos realizar aumentando el volumen, haciendo: $\text{muestra}(i) * (N - i) / N$ (donde $i = 0, 1, 2, 3, 4, \dots, N$). También se puede multiplicar por funciones logarítmicas, cuadráticas, etc.
- **Concatenar dos archivos de sonido:** Esta operación estándar como sabemos es tomar dos archivos de sonido con igual formato y unirlos formando uno solo.

- **Cambiar el número de bits por muestra:** Para pasar de 8 bits por muestra a 16 bits, se leerán las muestras mediante una variable de tipo *byte*, se restaran 128 para situar el silencio en el cero, se asignaran a una variable de tipo *integer* y se escribirán. Habrá que cambiar la cabecera con los nuevos datos del archivo. Para pasar de 16 bits por muestra a 8 bits se leerán las muestras mediante una variable de tipo *integer*, se dividirá entre 256 y se redondea para eliminar los decimales (con esto se elimina el byte menos significativo) , se suma 128 para situar bien el silencio, se asigna a una variable de tipo *byte* y se escribirán. Habrá que adecuar la cabecera.
- **Cambiar el número de canales:** Si queremos pasar un archivo de mono a estéreo se leerá cada muestra con una variable de tipo *byte* o *integer* dependiendo del tipo de sonido y se escribirá dos veces, una por cada canal. Recordar que en un archivo estéreo las muestras de los canales están entremezcladas. Para pasar de estéreo a mono se sumaran ambos canales como si fueran sonidos independientes, según se vio anteriormente.
- **Modificar la frecuencia de muestreo:** El objetivo de esta operación es modificar la frecuencia de muestreo alterando en la menor manera posible el sonido original del archivo de audio.

Para disminuir la frecuencia de muestreo (de 44.100Hz. a 22.050 Hz. o de 22.050 Hz. a 11.025 Hz.) se eliminará una muestra de cada dos, de forma ordenada.

Para aumentar la frecuencia de muestreo (de 11.025 Hz. A 22.050 o de 22.050 a 44.100 Hz.) se añadirá una nueva muestra por cada una que exista. Para calcular su valor debemos realizar una interpolación. La forma mas simple es

calculando la media aritmética para mayor calidad se puede realizar interpolaciones con mayor número de muestras.

2.3 Cálculo de patrones de voz.-

- **Cálculos en el dominio del tiempo.-** Los cálculos que se realizan aquí son:
 - Magnitud.
 - Energía
 - Promedio
 - Cruces por cero
 - Numero de máximos

Para poder observar los resultados debemos colocarlos en un gráfico y estos gráficos deben estar normalizados.

- **Magnitud:** Es la sumatoria del valor absoluto de las muestras que se hallan en una ventana.

$$M(i) = (1/Tv) \sum_{k=0}^{Tv-1} |m(k)|, \quad \dots\dots\dots(1.42)$$

donde M (i) es la magnitud de la ventana i, Tv el tamaño de la ventana y m (k) la muestra del sonido capturado.

- **Cálculo de la media:** La media se define como la sumatoria de todas las muestras de una ventana dividida entre el número de muestras. Se calcula así:

$$M(i) = (1/Tv) \sum_{k=0}^{Tv-1} m(k) , \quad \dots\dots\dots(1.43)$$

donde: $M(i)$ es la media de la ventana i, Tv el tamaño de la ventana y m(k) la muestra del sonido capturado.

- **Energía:** Es la sumatoria del cuadrado de las muestras, se divide entre el número de muestras para normalizar los resultados, se calcula con la fórmula:

$$E(i) = (1/Tv) \sum_{k=0}^{Tv-1} m^2(k) \quad \dots\dots\dots(1.44)$$

donde E(i) es la energía de la ventana i, Tv el tamaño de la ventana y m(k) la muestra del sonido capturado.

- **Calculo de los cruces por cero:** Los cruces por cero nos indican el número de veces que la señal atraviesa el nivel cero, en cualquiera de los dos sentidos. Solamente debemos considerar cuando hay un cambio de signo entre dos muestras consecutivas o en el caso de que haya un cero intermedio.
- **Calculo de los máximos:** Mediante este cálculo se busca el número de máximos locales que ocurren en una ventana. Se considera que ocurre un máximo cuando una muestra tiene un valor superior a la anterior y posterior. Podría suceder que encontráramos varios máximos consecutivos.
- **Cálculos en el dominio de la frecuencia.-**
 - **Transformada de Fourier:** En este cálculo cuando se aplica sobre una ventana nos devuelve un conjunto de resultados. Todos estos resultados deben estar normalizados. Tenemos programas que calculan el espectro de un archivo WAV utilizando la TF.
 - **Transformada rápida de Fourier:** Realiza el mismo cálculo o sea el espectro pero con un número de cálculos menores, es más veloz.
 - **Calculo de las máximas frecuencias:** Son las frecuencias más importantes que contribuyen a la formación de un dígito o una cadena de cuatro dígitos en caso de trabajar con una clave.

- **El promedio de frecuencias:** También será un parámetro que contribuirá a diferenciar un dígito vocalizado por un locutor del mismo dígito vocalizado por otro locutor. Y en el caso de una cadena de cuatro dígitos será mas preciso el reconocimiento pues la probabilidad de coincidencias es pequeña.

2.4 Programas básicos para señales de audio.

- **Programa de lectura.-** Lee un archivo de audio y lo gráfica.

```
% Programa: Leer.m
% clc; echo off;
% *** EXTRACCION DEL ARCHIVO ***
[x,Fs,Bits]=wavread('c:\matlabr11\huacha\muestra.wav');
a=length(x);
fprintf('Longitud del vector: '); a
[mi,ii]=min(x); [ma,ii]=max(x);
plot(x); grid; zoom on;
ylabel('Amplitud');
xlabel('Número de muestras');
title('FORMA DE ONDA');
axis([0 a(1) mi ma]);
%
sound(x, Fs);
```

- **Programa FFT.-** Programa que halla la transformada rápida de Fourier.

```
% Programa: Halla la transformada rápida de Fourier
fsx=Fs; figure(2);
Fx=fft(x,512);
w=(0:255)/512*fsx; %plot(w,abs(Fx(1:256)));
subplot(2,1,1); plot(w,abs(Fx(1:256)));
title('FAST FOURIER TRANSFORM de x');
ylabel('Mag. '); grid; zoom on;
subplot(2,1,2); stem(w,abs(Fx(1:256)));
grid; zoom on; %axis([0 2000 0 6]);
ylabel('Mag. ');
xlabel('Frecuencia');
```

- **Programa de Filtrado.-** Programa que filtra el archivo de frecuencias y le aplica un filtro Pasabajo de CHEBYSHEV de 350 Hz. Para posteriormente construir un gráfico en frecuencia.

```

xp=x; figure(4);
subplot(3,1,1); plot(x);
title('APLICANDO FILTRO CHEBYSHEV PASABAJOS Wc=
350Hz');
ylabel('S/FILTRO'); zoom on;
x=xp';
[b,a]=cheby1(4,0.005,1200*2/Fs); %FCorte
[H,w]=freqz(b,a,512);
subplot(3,1,2); plot(w*Fs/(2*pi),abs(H));
ylabel('FILTRO'); zoom on;
xf=filter(b,a,x);
subplot(3,1,3); plot(xf);
ylabel('C/FILTRO');
zoom on; grid; %pause;
% *** RESPUESTA EN FRECUENCIA C/S FILTRO ***
figure(5);
X=fft(x,512);
XF=fft(xf,512);
w=(0:255)/256*(Fs/2);
plot(w,abs([X(1:256)'XF(1:256)']));
title('VOCAL C/S FILTRAR EN FREC. ');
xlabel('Frecuencia');
ylabel('Magnitud de la Trans. de Fourier');
zoom on; grid;
x=xp; xf=xf';

```

- **Programa que modifica velocidad de audio.-** Programa ejecuta el mensaje más lento o más rápido variando el archivo.

```

% variando velocidad
clc;
fprintf('\n\n\nPresione tecla para Mensaje mas lento ... ');
pause; sound(xp,Fs/1.5);
fprintf('\n\n\nPresione tecla para Mensaje mas rápido ... ');
pause; sound(xf,Fs*2);

```

- **Programa de espectro.-** Programa que halla el espectrograma del audio.

```
% Hallando espectro y mostrando la gráfica xx = x(1:6000);  
specgram (xx,256,Fs,256,250);  
title('ESPECTROGRAMA DE x');
```

- **Programa inverso.-** Programa que copia el archivo de audio al revés.

```
% Copia archivo de audio al revés.  
xy=[];  
for i=1:length(x)  
    yy=x(length(x)-i+1);  
    xy=[xy yy];  
end  
sound(xy,Fs);
```

CAPÍTULO III PROGRAMAS PARA RECONOCIMIENTO DE LOCUTORES

3.1 Analisis en el dominio del tiempo

- **Programa 3.1.1**

Lee un dígito en un periodo de tiempo determinado y entrega solamente el dígito sin silencios.

```
% Entrega un vector con la señal limpia  
% Lee archivo cas1.wav. Archivo tiempo01  
%yc1 vector original 154364 componentes  
%y1cas vector sin silencios 3001 componentes, y1casn es y1cas  
normalizado  
clc; echo off;  
%*****PROCEDIMIENTO DE LECTURA*****  
[yc1, Fs, Bits] = wavread('a:cas1.wav');  
lon = size(yc1); fprintf('\n Longitud del vector: %d \n', lon(1));pause;  
%*****CALCULA MAYOR MENOR Y MAYOR  
ABSOLUTO*****  
[mayor, jj]=max(yc1); [menor, jj]=min(yc1);  
if abs(mayor) > abs(menor)  
    mayabs = abs(mayor);  
else  
    mayabs = abs(menor);  
end;  
%*****HALLA ARCHIVOS SIN SILENCIOS*****  
des = (mayabs/100)*20;  
a= size(yc1);  
for i=1:lon(1);  
    if abs(yc1(i)) > des  
        lima = i; break;  
    end;
```

```

end;
limb = lima +3000
k= 0:( limb-lima);
y1cas = yc1(lima :limb);
b = size(y1cas);
fprintf('\n Longitud del vector: %d \n',b(1));
y1casn = y1cas;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb - lima)+1;
    y1casn(i)=y1casn(i)/mayabs;
end;
%*****GRAFICA DE LOS VECTORES *****
subplot(3,1,1); t=1:lon(1); plot(t,yc1);
grid; ylabel('yc1');title('Digito dos de castañeda');
axis([0 a(1) menor mayor]);
subplot(3,1,2);t=k; plot(t,y1cas);grid;
axis([0 (limb-lima) menor mayor]);ylabel('y1cas');
subplot(3,1,3);plot(t,y1casn);grid; axis([0 (limb-lima) -1 1]);
ylabel('y1casn'); xlabel('Numero de muestras');
fprintf('\n\n Presione una tecla para terminar...');pause;

```

Resultado de la ejecución:

```

Longitud del vector: 154364
Longitud del vector: 3001
Presione una tecla para terminar...

```

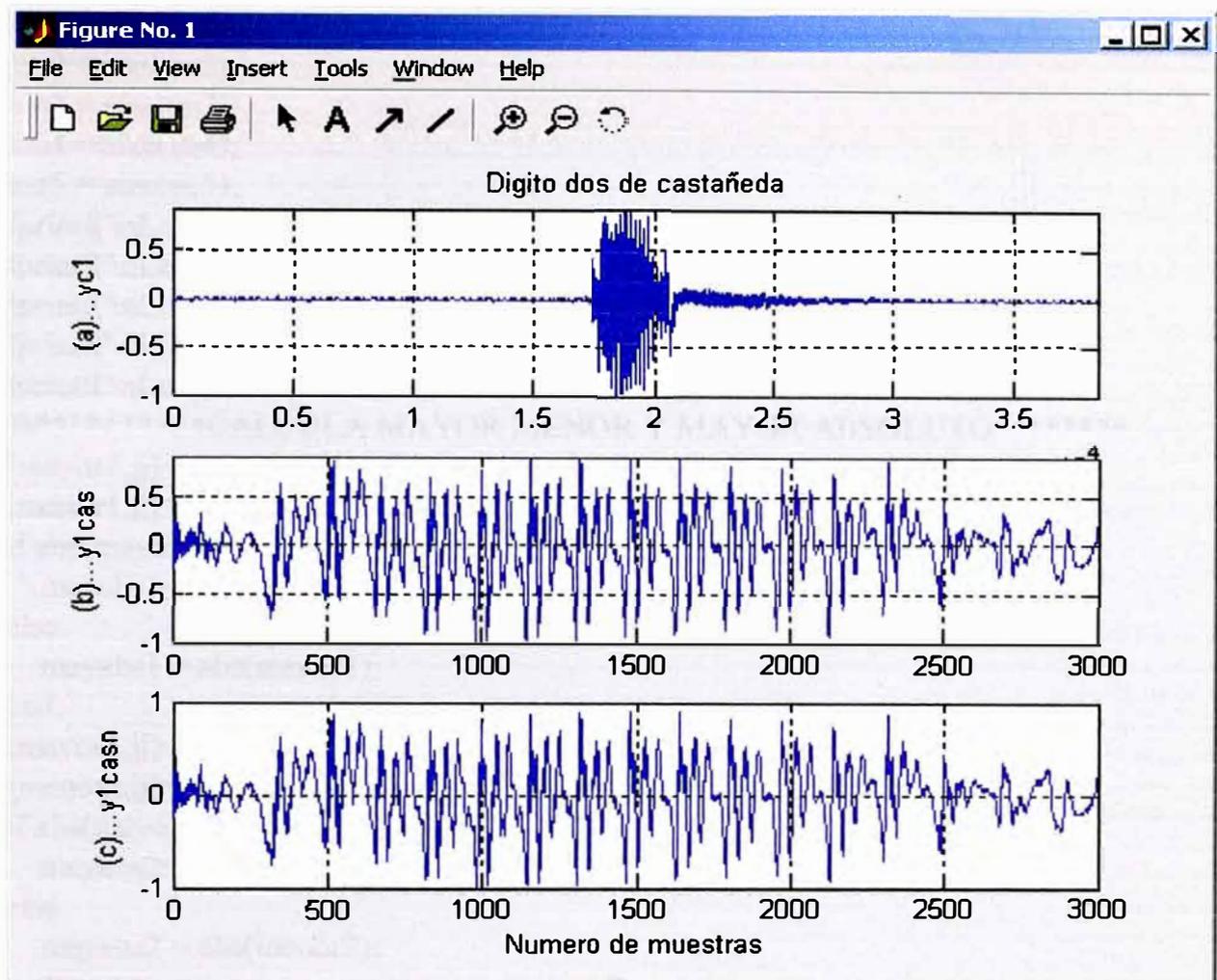


Fig. 3.1. (a) Señal de voz original (b) Señal de voz sin silencios (c) Señal de voz sin silencios y normalizada.

- **Programa 3.1.2.-** Lee 5 veces un dígito del mismo locutor y lo muestra

```
% Lee 5 archivos de voz. Tiermpo2
% Visualiza los 5 archivos de voz sin silencios y normalizados
% Los archivos procesados son cas1, cas2,cas3,cas4 y cas5.wav
% Todos los archivos son del locutor castañeda
clc;
echo off;
%*****PROCEDIMIENTO DE LECTURA*****
[yc1, Fs, Bits] = wavread ('a:cas1.wav');
[yc2, Fs, Bits] = wavread ('a:cas2.wav');
[yc3, Fs, Bits] = wavread ('a:cas3.wav');
[yc4, Fs, Bits] = wavread ('a:cas4.wav');
[yc5, Fs, Bits] = wavread ('a:cas5.wav');
```

```

lon1 = size(yc1);
lon2 = size(yc2);
lon3 = size(yc3);
lon4 = size(yc4);
lon5 = size(yc5);
fprintf('\nLongitud del vector yc1: %d \n',lon1(1));
fprintf('\nLongitud del vector yc2: %d \n',lon2(1));
fprintf('\nLongitud del vector yc3: %d \n',lon3(1));
fprintf('\nLongitud del vector yc4: %d \n',lon4(1));
fprintf('\nLongitud del vector yc5: %d \n',lon5(1));
%*****CALCULA MAYOR MENOR Y MAYOR ABSOLUTO*****
[mayor1,jj]= max(yc1);
[menor1,jj]= min(yc1);
if abs(mayor1) > abs(menor1)
    mayabs1= abs(mayor1);
else
    mayabs1 = abs(menor1);
end;
[mayor2,jj]= max(yc2);
[menor2,jj]= min(yc2);
if abs(mayor2) > abs(menor2)
    mayabs2= abs(mayor2);
else
    mayabs2 = abs(menor2);
end;
[mayor3,jj]= max(yc3);
[menor3,jj]= min(yc3);
if abs(mayor3) > abs(menor3)
    mayabs3= abs(mayor3);
else
    mayabs3 = abs(menor3);
end;
[mayor4,jj]= max(yc4);
[menor4,jj]= min(yc4);
if abs(mayor4) > abs(menor4)
    mayabs4= abs(mayor4);
else
    mayabs4 = abs(menor4);
end;
[mayor5,jj]= max(yc5);
[menor5,jj]= min(yc5);
if abs(mayor5) > abs(menor5)
    mayabs5= abs(mayor5);

```

```

else
    mayabs5 = abs(menor5);
end;
%*****HALLA ARCHIVOS SIN SILENCIOS*****
des = (mayabs1/100)*20;
for i=1:lon1(1);
    if abs(yc1(i)) > des
        lima1 = i;
        break
    end;
end;
limb1=lima1+3000;
k1= 0:(limb1-lima1);
y1cas = yc1(lima1 :limb1);
y1casn = y1cas;
%*****
des = (mayabs2/100)*20;
for i=1:lon2(1);
    if abs(yc2(i)) > des
        lima2 = i;
        break
    end;
end;
limb2=lima2+3000 ;
k2= 0:(limb2-lima2);
y2cas = yc2(lima2 :limb2);
y2casn = y2cas;
% *****
des = (mayabs3/100)*20;
for i=1:lon3(1);
    if abs(yc3(i)) > des
        lima3 = i;
        break
    end;
end;
limb3=lima3+3000;
k3= 0:(limb3-lima3);
y3cas = yc3(lima3 :limb3);
y3casn = y3cas;
% *****
des = (mayabs4/100)*20;
for i=1:lon4(1);
    if abs(yc4(i)) > des

```

```

        lima4 = i;
        break
    end;
end;
limb4= lima4+3000;
k4= 0:(limb4-lima4);
y4cas = yc4(lima4 :limb4);
y4casn = y4cas;
% *****
des = (mayabs5/100)*20;
for i=1:lon5(1);
    if abs(yc5(i)) > des
        lima5 = i;
        break
    end;
end;
limb5= lima5+3000;
k5= 0:(limb5-lima5);
y5cas = yc5(lima5 :limb5);
y5casn = y5cas;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb1 - lima1)+1;
    y1casn(i)=y1casn(i)/mayabs1;
end;
for i=1:(limb2 - lima2)+1;
    y2casn(i)=y2casn(i)/mayabs2;
end;
for i=1:(limb3 - lima3)+1;
    y3casn(i)=y3casn(i)/mayabs3;
end;
for i=1:(limb4 - lima4)+1;
    y4casn(i)=y4casn(i)/mayabs4;
end;
for i=1:(limb5 - lima5)+1;
    y5casn(i)=y5casn(i)/mayabs5;
end;
%*****GRAFICA DE LOS VECTORES *****
% ***vector1***
subplot(5,1,1);
t = k1;
plot(t, y1casn);grid;
axis([0 (limb1-lima1) -1 1]);
ylabel('y1casn');

```

```

xlabel('Numero de muestras');
% ***vector2***
subplot(5,1,2);
t = k2;
plot(t, y2casn);grid;
axis([0 (limb2-lima2) -1 1]);
ylabel('y2casn');
xlabel('Numero de muestras');
% ***vector3***
subplot(5,1,3);
t = k3;
plot(t, y3casn);grid;
axis([0 (limb3-lima3) -1 1]);
ylabel('y3casn');
xlabel('Numero de muestras');
% ***vector4***
subplot(5,1,4);
t=k4;
plot(t, y4casn);grid;
axis([0 (limb4-lima4) -1 1]);
ylabel('y4casn');
xlabel('Numero de muestras');
% ***vector5***
subplot(5,1,5);
t=k5;
plot(t, y5casn);grid;
axis([0 (limb5-lima5) -1 1]);
ylabel('y5casn');
xlabel('Numero de muestras');
fprintf('\n\n Presione una tecla para terminar...');
pause;

```

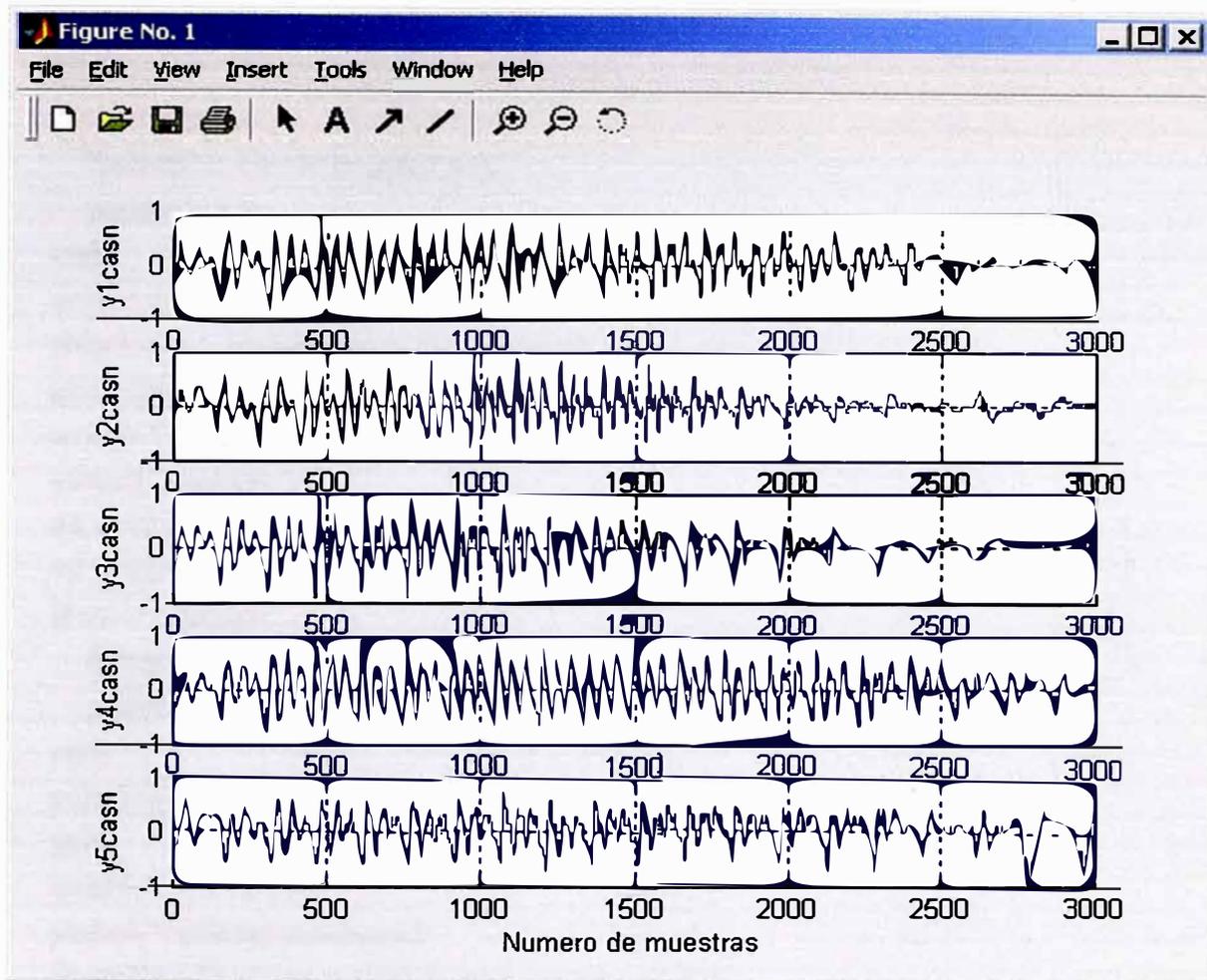


Fig.3.2 Muestra 5 señales de voz del mismo locutor (digito 2 de Castañeda)

- **Programa 3.1.3.-** Muestra la señal de voz por ventanas de un determinado ancho

%Nos muestra un dígito con ventanas de ancho variable

%Programa tiempo03

%Utiliza archivo normalizado: y1casn

clc;

tv=1000; tv=input('ingrese ancho de la ventana ==> ');

figure(1);

tam = size(y1casn);

tamano = tam(1);

cont = 0;

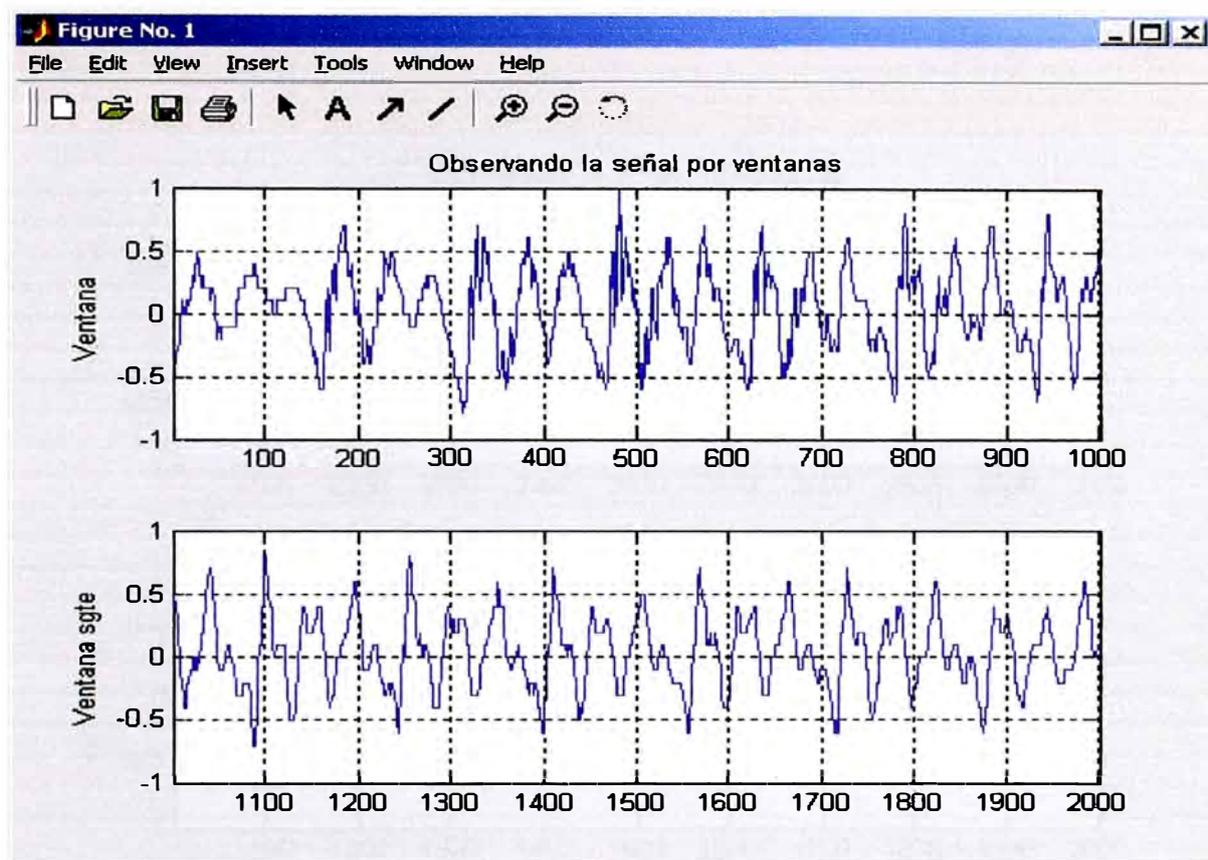
for v= 1 : 2*tv :tamano;

cont = cont + 1;

```

v1 = v; %VALOR INFERIOR DE LA VENTANA
v2 = v + tv: %VALOR SUPERIOR DE LA VENTANA
if v2 > tamaño
    fprintf('\n Fin de la señal \n');
    pause; break;
end;
cont = cont + 1;
subplot(2,1,1); plot([v1:v2],y1 casn(v1:v2));
grid; title('Observando la señal por ventanas');
axis([v1 v2 -1 1]);
ylabel('Ventana ');
v1 = v2; %VALOR INFERIOR DE LA VENTANA
v2 = v2+ tv : %VALOR SUPERIOR DE LA VENTANA
if v2 > tamaño
    fprintf('\n Fin de la señal \n');
    pause; break;
end;
subplot(2,1,2); plot([v1:v2],y1 casn(v1:v2));
grid;
axis([v1 v2 -1 1]);
ylabel('Ventana siguiente');
fprintf('\n Presione una tecla para continuar \n');
pause;
end;

```



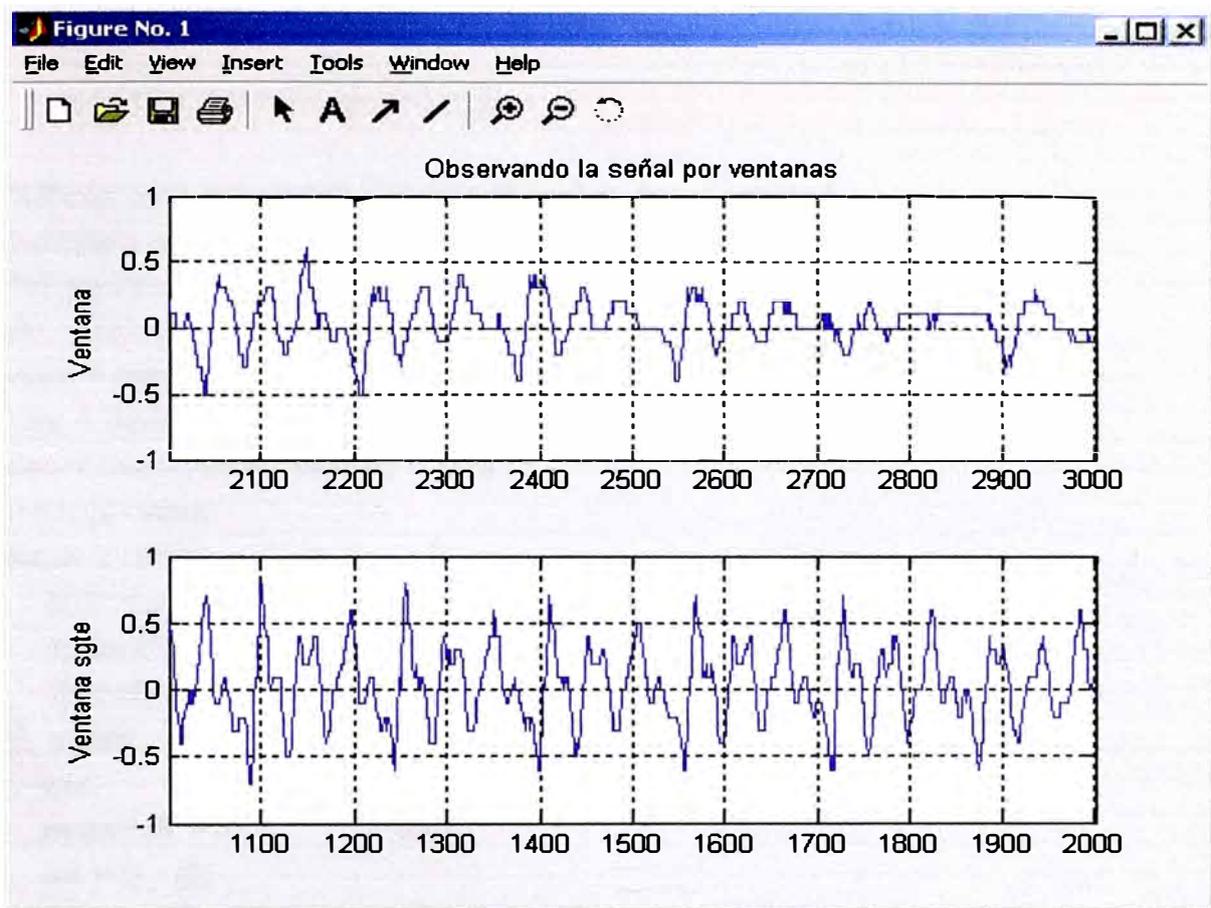


Fig.3.3 (a) Señal de voz de la muestra 1 a 1000 (b) Señal de voz de la muestra 1000 a 2000 (c) Señal de voz de la muestra 2000 a 3000

- **Programa 3.1.4.-** Calcula el parámetro magnitud en el dominio del tiempo con la formula anteriormente indicada

%Programa tiempo04. Calcula el parámetro magnitud

%Utiliza archivo normalizado: y2casn

%Considera tamaño de ventana e incremento de ventana

```

clc; fprintf('Inicio de programa \n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn); tamano = tam(1);
j1=1; j2=vtam;
for i= 1 :100;
    if(j2 >tamano)
        fprintf('\n FIN DEL PROGRAMA \n');
        fprintf('\n Presione enter para mostrar tabla...\n');
        pause; break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1casn(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc; j2 = j2 + vinc;
end;
m=i-1;
clc; fprintf('\n \t\t\t MAGNITUDES \n');fprintf('\t\t\t =====\n\n');
fprintf(' \t\t\t [Indice] \t [Mag]\n');
for i=1:m;
    fprintf(' \t\t\t %d \t\t %8.2f\n',i,magnitudes(i));
end;

```

Ejecución del programa tiempo04:


```
fprintf(' \t\t\t[Índice] \t[Energía]\n');
for i=1:m;
    fprintf(' \t\t\t %d \t %6.2f\n',i,energias(i));
end;
```

Ejecución del programa tiempo05:

```
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
Presione enter para visualizar tabla...
          ENERGIAS
          =====
          [Índice] [Energía]
              1      0.10
              2      0.12
              3      0.11
              4      0.10
              5      0.08
              6      0.08
              7      0.05
              8      0.04
              9      0.02
```

- **Programa 3.1.6.-** Calcula los cruces por cero de una señal de voz de un dígito.

%Programa que calcula cruces por cero

%Programa tiempo06, utiliza archivo normalizado: y1casn

```
clc; fprintf('Inicio de programa \n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn); tamano = tam(1);
j1=1; j2=vtam;
for i= 1 :100;
    if(j2 >tamano)
        fprintf('\n Presione enter para visualizar tabla. \n'); pause; break;
    end;
    estado = 0; cruce = 0;
    for j =j1 : j2;
        if (y1casn(j) ~= 0)
            if (estado == 0)
                estado = y1casn(j);
```

```

else if((ylcasn(j)>0)& (estado<0))| ((ylcasn(j)<0) & (estado>0))
    cruce = cruce +1; estado = ylcasn(j);
    end; end;
end; end;
cruces(i) = cruce;
j1 = j1 + vinc; j2 = j2 + vinc;
end;
m = i -1;
fprintf('\n CRUCES POR CERO \n'); fprintf('-----\n');
fprintf(' [Indice] \t[Cruces]\n');
for i=1:m;
    fprintf(' %d\t %d\n',i, cruces(i));
end;

```

Ejecución del programa tiempo06:

```

ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
Presione enter para visualizar tabla...
CRUCES POR CERO
=====
[Índice] [Cruces]
1         23
2         27
3         23
4         25
5         27
6         24
7         20
8         15
9         13

```

- **Programa 3.1.7.-** Calcula los máximos en una señal de voz que es un dígito

%Programa que calcula máximos

%Programa tiempo07

%Utiliza archivo normalizado: ylcasn

%Considera tamaño de ventana e incremento de ventana

clc;

fprintf('Inicio de programa \n');

vtam= input('ingrese tamaño de la ventana ==> ');

vinc= input('ingrese incremento de la ventana ==> ');

```

%figure(1);
tam = size(y1casn);
tamano = tam(1);
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause; break;
    end;
    estado = -1;
    anterior = y1casn(j1);
    maximo = 0;
    for j=j1+1 : j2;
        if ((y1casn(j)<anterior)&(estado == 1))
            maximo = maximo +1;
            estado = -1;
        else
            if((y1casn(j)> anterior)& (estado ==-1))
                estado = 1;
            end;
        end;
        anterior=y1casn(j);
    end;
    maximos(i)= maximo;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
m=i-1;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf('=====\n');
fprintf(' [Índice] \t[Máximos] \n');
for i=1:m;
    fprintf(' %d\t %d\n',i, maximos(i));
end;

```

Ejecución del programa tiempo07:

```

Inicio de programa
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
FIN DEL CALCULO
Presione enter para visualizar tabla...
MAXIMOS DE LA SENAL
=====
[Índice] [Máximos]
1         47
2         60
3         43
4         32
5         30
6         24
7         19
8         19
9         22

```

3.2 Analisis en el dominio de la frecuencia

- **Programa 3.2.1.-** Utiliza la FFT y pasa la señal del tiempo al dominio de la frecuencia

% Programa halla la transformada de fourier

% Programa frec01

% Vector y1casn

clc; echo off;

%***PROCEDIMIENTO DE LECTURA*******

[yc1, Fs, Bits] = wavread ('a:cas1.wav');

lon = size(yc1);

fprintf('\nLongitud del vector: %d\n',lon(1));pause;

%***CALCULA MAYOR MENOR Y MAYOR ABSOLUTO*******

[mayor,jj]=max(yc1);

[menor,jj]=min(yc1);

if abs(mayor) > abs(menor)

 mayabs = abs(mayor);

else

 mayabs = abs (menor);

end;

```

%*****HALLA ARCHIVOS SIN SILENCIOS*****
des = (mayabs/100)*20;
a= size(ycl);
for i=1:lon(1);
    if abs(ycl(i)) > des
        lima = i;
        break
    end;
end;
limb=lima+3000
k= 0:(limb-lima);
y1cas = ycl(lima :limb);
b = size(y1cas);
fprintf('\nLongitud del vector: %d\n',b(1));
y1casn = y1cas;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb - lima)+1;
    y1casn(i)=y1casn(i)/mayabs;
end;
clc; tam= size(y1casn); tamaño = tam(1);
    fprintf('\n Inicio del programa \n');
% *****Mostramos la señal*****
figure(1);fs =Fs;
subplot(2,1,1);plot([1:tamaño],y1casn);grid;
title('Dígito dos de Castañeda');ylabel('y1casn');zoom on;
%***Mostramos la transformada de Fourier****
Fy1casn=fft(y1casn,512);
w=(0:255)/512*fs;
subplot(2,1,2);plot(w, abs(Fy1casn(1:256)));grid;
ylabel('Fourier de y1casn');zoom on;
%*****Termina proceso*****
fprintf('transformada generada, Presione tecla para terminar.....');

```

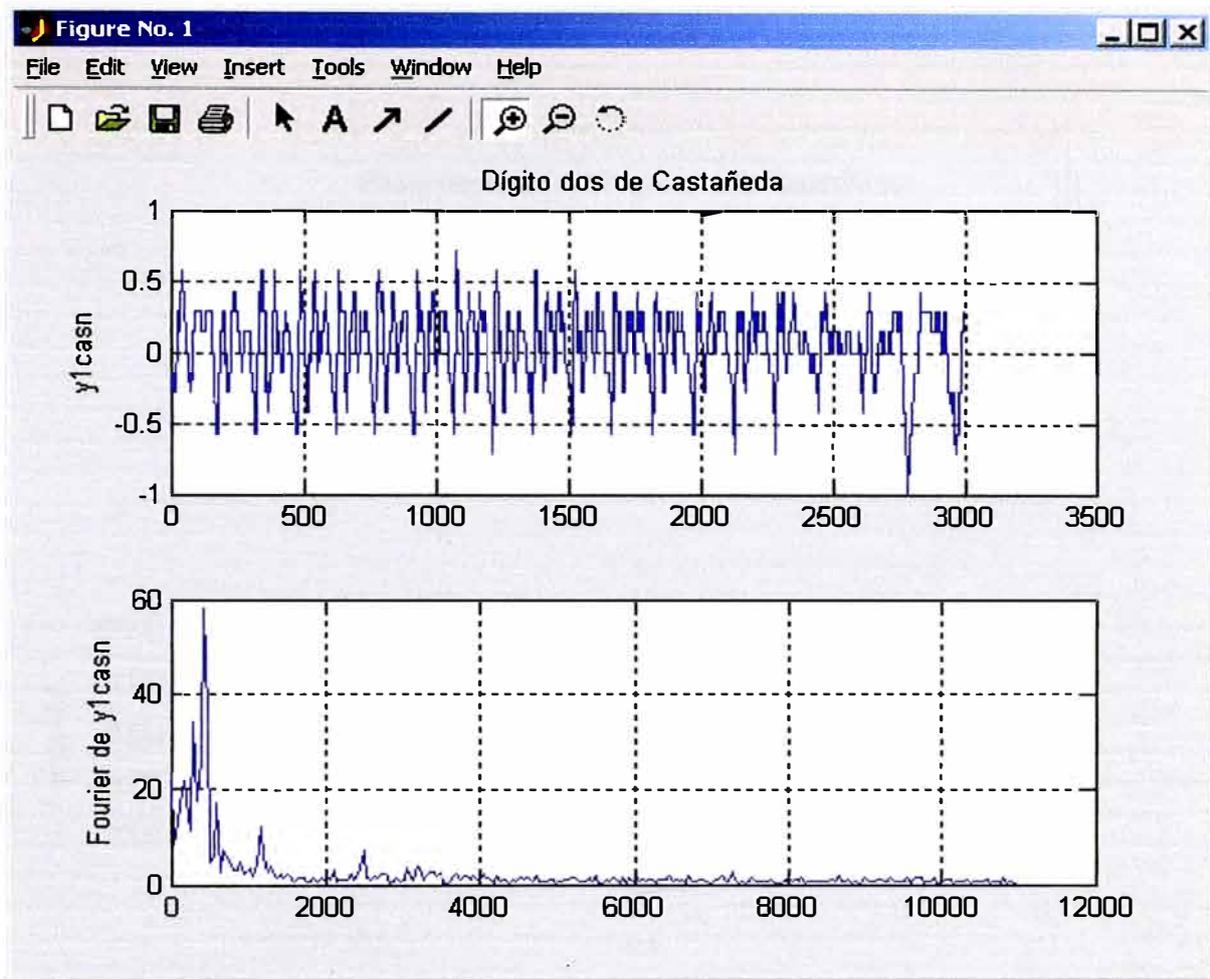


Fig.3.4 (a) Dígito dos de castañeda señal en el tiempo **(b)** Transformada de Fourier, señal en el dominio de la frecuencia.

- **Programa 3.2.2.-** Muestra el espectrograma de el dígito dos de castañeda

```
% Programa grafica espectrograma
```

```
% Programa frec02
```

```
% Vector y1casn
```

```
clc;
```

```
%*****Mostramos la señal*****
```

```
figure(1); subplot(2,1,1);plot(y1casn);grid;
```

```
title('Espectrograma del Dígito dos de Castañeda');
```

```
ylabel('y1casn');zoom on;
```

```
%****Mostramos espectrograma*****
```

```
subplot(2,1,2);
```

```
specgram(y1casn,256, Fs,256, 250);
```

```
%*****Termina proceso*****
```

```
fprintf('Espectrograma mostrado, Presione tecla para terminar.....');
```

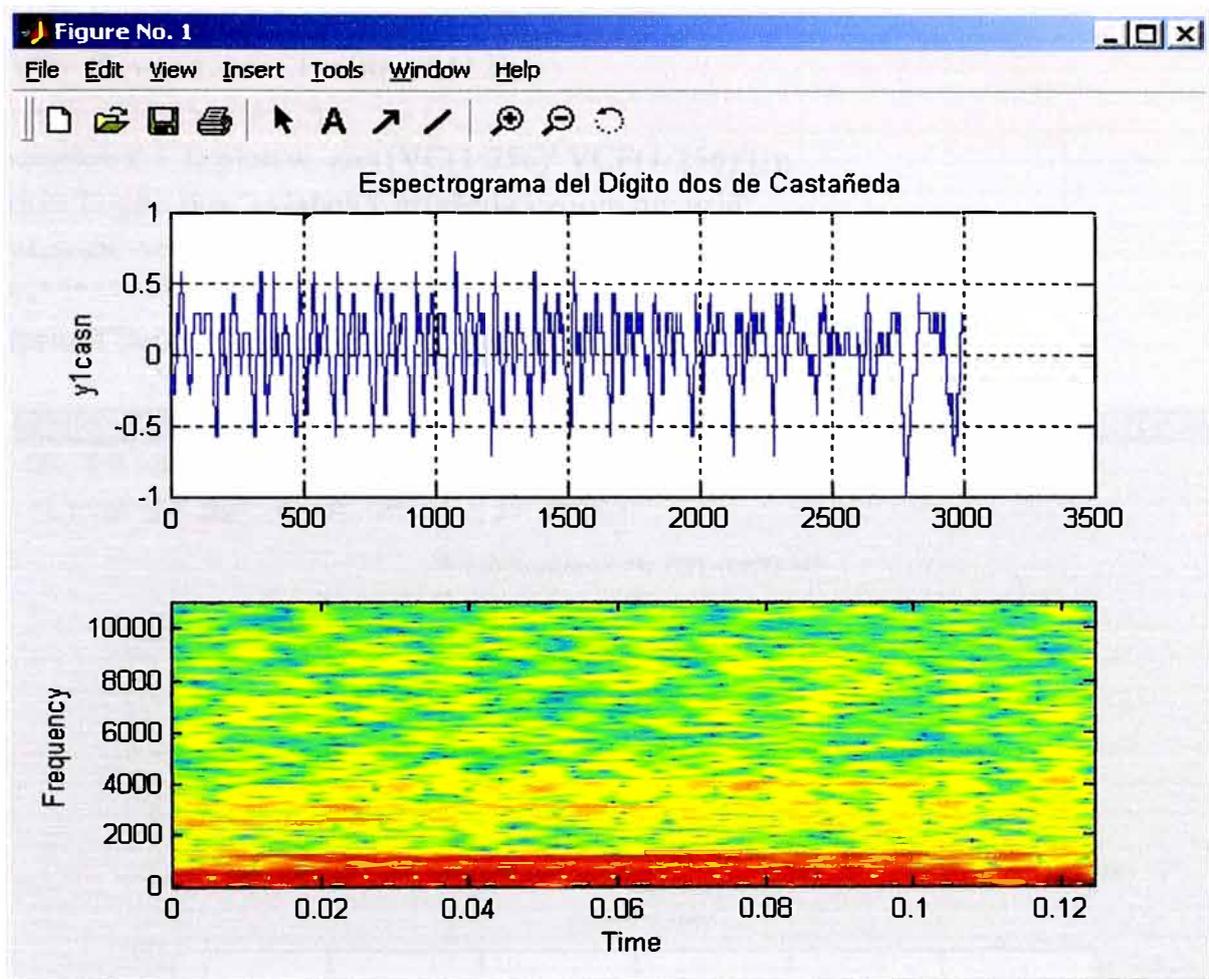


Fig.3.5 (a) Señal de voz en el tiempo **(b)** Espectrograma en la frecuencia

- **Programa 3.2.3.-** Filtra el dígito dos para frecuencias de 100 a 1800 Hz.

% Programa filtra digito

% Programa frec03

% vector y1casn

clc;

vc= y1casn'; % vc es voz de Castañeda

%*****Filtramos la señal*****

figure(1);

[b,a]=butter(4,[100 1800]*2/Fs);

[H, w]= freqz(b,a,512);

subplot(2,1,1);

plot(w*Fs/(2*pi),abs(H));

title('Filtro Butterworth 100 -1800 Hz');grid;zoom on;

```
vcf= filter(b,a,vc);
VC= fft(vc,512);VCF= fft(vcf,512);
w=(0:255)/256*(Fs/2);
subplot(2,1,2);plot(w, abs([VC(1:256)' VCF(1:256)']));
title('Dígito dos ');ylabel('Castañeda');zoom on; grid;
ylcasnf=vcf';
%*****Termina proceso*****
fprintf('Señal filtrada , Presione tecla para terminar.....');
```

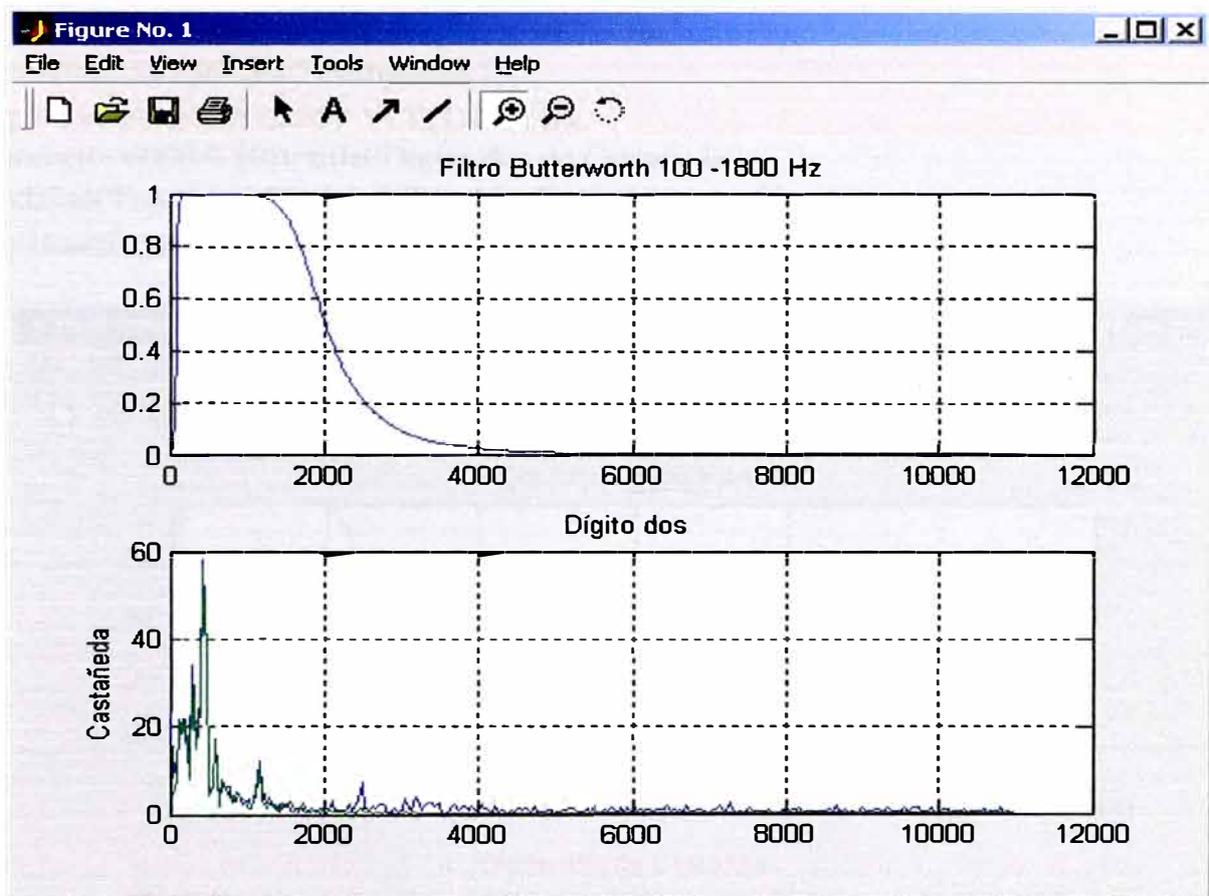


Fig.3.6 (a) Grafico del filtro butterworth de 101 a 11800 Hz. **(b)** Transformada de Fourier de la señal (azul) y la señal filtrada (verde).

- **Programa 3.2.4.-** Filtra el dígito dos con filtro tipo elíptico

% Aplicación de filtro elíptico a vector de voz

% Archivo frec04.m, Vector: ylcase

clc; vc= ylcase';

figure(1);

[b,a]= ellip(4,0.005,40,[100 1800]*2/Fs);

[H,w]=freqz(b,a,512);

subplot(2,1,1); plot(w*Fs/(2*pi), abs(H));

title('Filtro Elíptico pasabanda'); grid; zoom on;

vcf= filter(b,a,vc); VC= fft(vc,512); VCF= fft(vcf,512);

w=(0:255)/256*(Fs/2);subplot(2,1,2);

plot(w, abs([VC(1:256)' VCF(1:256)']));

axis([0 6000 0 50]); title('Dígito dos de Castañeda');

xlabel('Frecuencia'); ylabel('Magnitud'); zoom on; grid; pause;

ylcasnf=vcf;fprintf('\n Señales filtradas: \n\n');

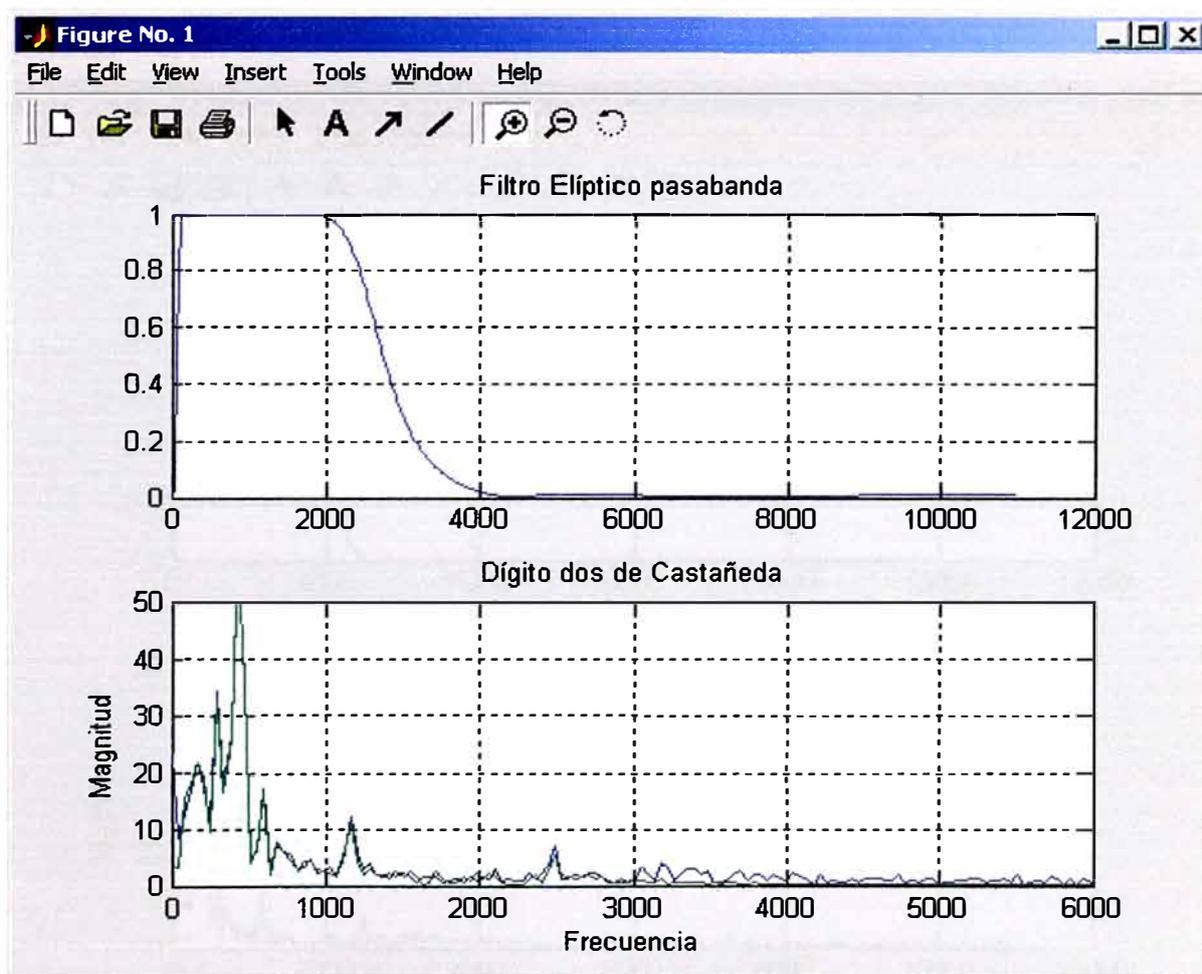


Fig.3.7 (a) Filtro elíptico de 100 a 1800 **(b)** Voz sin filtro(azul), con filtro (verde).

- **Programa 3.2.5.-** Aplica filtro chebychev a señal de voz dígito dos

% Filtrando frecuencias formantes

% Archivo frec05.m

% Aplica filtros pasabajo a y1casn

clc;

vc=y1casn'; figure(1);

[b,a]=cheby1(12,0.005,2000*2/Fs);

[H,w]=freqz(b,a,512);

subplot(2,1,1);plot(w*Fs/(2*pi),abs(H));

title('FILTRO CHEBYSHEV PASABAJOS'); grid; zoom on;

vcf = filter(b,a,vc);VC= fft(vc,512); VCF= fft(vcf,512);

w=(0:255)/256*(Fs/2);

subplot(2,1,2);plot(w, abs([VC(1:256)' VCF(1:256)']));

title('Dígito dos Castañeda'); ylabel('Magnitud');

zoom on; grid; pause;

y1casnf=vcf';

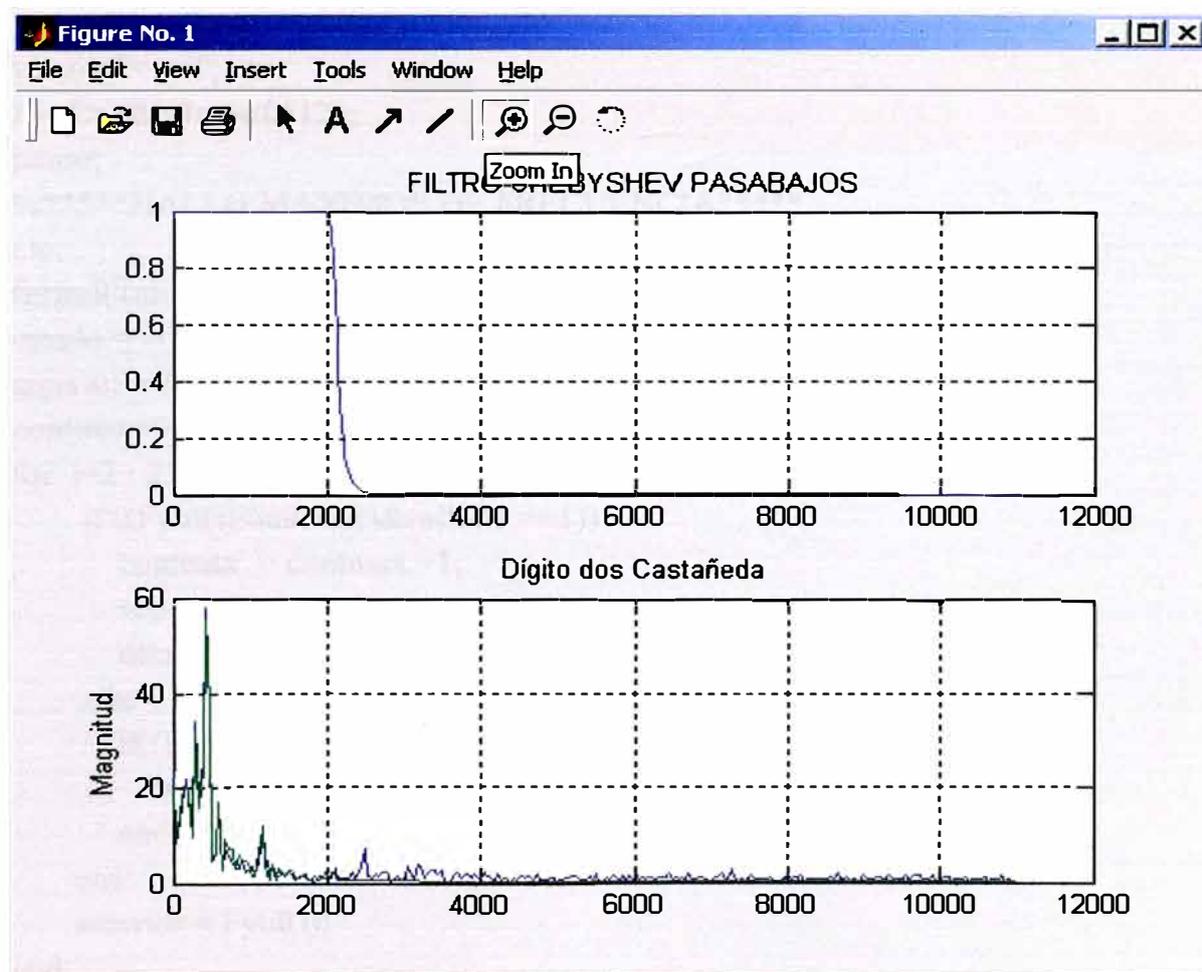


Fig.3.8 (a) Filtro chebyshev de 100 a 1800 **(b)** Voz sin filtro(azul), con filtro (verde).

- **Programa 3.2.6.-** Halla máximos del dígito dos pero en la frecuencia

```

%Hallo máximos de frecuencia
% Archivo frec06.m
% Vector: y1casn
% *****
clc;
vc=y1casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vcf= filter(b,a,vc);
figure(1);
VC= fft(vc,512); VCF= fft(vcf,512);
w=(0:255)/256*(Fs/2);
stem(w, abs(VC(1:256')));
axis ([0 1800 0 30]);
title('Digito dos ');
ylabel('CASTAÑEDA');zoom on;grid;
y1casnf = vcf;
Fynf = fft(y1casnf,512);
pause;
%*****HALLO MAXIMOS DE FRECUENCIA*****
clc;
fprintf('Inicio de programa \n');
estado = -1;
anterior = Fynf(1);
contmax=0;
for i=2 : 256;
    if ((Fynf(i)<anterior)&(estado ==1))
        contmax = contmax +1;
        vcon(contmax)=i-1;
        estado = -1;
    else
        if((Fynf(i)> anterior)& (estado ==-1))
            estado = 1;
        end;
    end;
    anterior = Fynf(i);
end;
for i= 1:contmax;

```

```

kk=vcon(i);
mayores(i)=abs(Fynf(kk));
frec(i)=w(kk);
end;
[mayor,jj]=max(mayores);
K1=0;
for i=1:contmax;
    if mayores(i)>0.20*mayor;
        K1=K1+1;
        mayores1(K1)=mayores(i);
        frec1(K1)=frec(i);
    end;
end;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf('=====\n');
fprintf(' [Índice] \t[Frecuencia]\t[Magnitud]\n');
for i=1:K1;
    fprintf(' %dt \t%8.2ft%6.2fn',i,frec1(i),mayores1(i));
end;

```

Ejecución del programa frec06:

Inicio de programa

MAXIMOS DE LA SENAL

```

=====
[Índice] [Frecuencia] [Magnitud]
1         129.20      21.89
2         215.33      21.58
3         387.60      24.71
4         559.86       6.99
5         775.20       6.22
6        1119.73       5.38

```

- **Programa 3.2.7.- Halla promedio de frecuencias**

```

% Filtra digito dos y halla promedio de frecuencias **
% Archivo frec07.m
% Vectores:
% y1casn
% y1casnf
% vc y vcf
% vcf y vcf
% *****
clc;
vc=y1casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vcf=filter(b,a,vc);
figure(1);
VC=fft(vc,512); VCF=fft(vcf,512);
w=(0:255)/256*(Fs/2);
%subplot(3,1,1);
stem(w,abs(VC(1:256')));axis([0 1800 0 30]);
title('Digito dos ');
ylabel('CASTAÑEDA');zoom on;grid;y1casnf=vcf;
Fynf=fft(y1casnf,512);pause;
%*****HALLO PROMEDIO DE FRECUENCIA*****
clc;
fprintf('Inicio de programa \n');promedio = mean(abs(Fynf))*100;
fprintf('\n Promedio = %8.2f\n',promedio);
fprintf('\n=====FIN      ===== \n');
%*****

```

Ejecución del programa frec07:

Inicio de programa

Promedio = 160.04

=====FIN=====

3.3 Comparación de patrones en el tiempo y en la frecuencia.

- **Programa 3.3.1.-** Lee cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), y entrega la señal limpia y normalizada.

```

% Lee 4 archivos de voz
% Entrega los vectores con la señal limpia
% Lee archivo cas1.wav, cas2.wav,cas3.wav y walter1.wav
%Archivo compa01
%yc1,yc2,yc3,yw1      vectores originales
%y1cas,y2cas,y3cas,y1wal  vectores sin silencios
%y1casn,y2casn,y3casn,y1waln  vectores sin silencios y normalizados
clc;
echo off;
%*****PROCEDIMIENTO DE LECTURA*****
[yc1, Fs, Bits] = wavread ('a:cas1.wav');
[yc2, Fs, Bits] = wavread ('a:cas2.wav');
[yc3, Fs, Bits] = wavread ('a:cas3.wav');
[yw1, Fs, Bits] = wavread ('a:walter1.wav');
lon1 = size(yc1);
lon2 = size(yc2);
lon3 = size(yc3);
lon4 = size(yw1);
fprintf('\nLongitud del vector yc1: %d \n',lon1(1));
fprintf('\nLongitud del vector yc2: %d \n',lon2(1));
fprintf('\nLongitud del vector yc3: %d \n',lon3(1));
fprintf('\nLongitud del vector yw1: %d \n',lon4(1));
%*****CALCULA MAYOR MENOR Y MAYOR ABSOLUTO*****
[mayor1,jj]= max(yc1);
[menor1,jj]= min(yc1);
if abs(mayor1) > abs(menor1)
    mayabs1= abs(mayor1);
else
    mayabs1 = abs(menor1);
end;
[mayor2,jj]= max(yc2);
[menor2,jj]= min(yc2);
if abs(mayor2) > abs(menor2)
    mayabs2= abs(mayor2);
else
    mayabs2 = abs(menor2);

```

```

end;
[mayor3,jj]= max(y3);
[menor3,jj]= min(y3);
if abs(mayor3) > abs(menor3)
    mayabs3= abs(mayor3);
else
    mayabs3 = abs(menor3);
end;
[mayor4,jj]= max(yw1);
[menor4,jj]= min(yw1);
if abs(mayor4) > abs(menor4)
    mayabs4= abs(mayor4);
else
    mayabs4 = abs(menor4);
end;
%*****HALLA ARCHIVOS SIN SILENCIOS*****
des = (mayabs1/100)*20;
for i=1:lon1(1);
    if abs(y3(i)) > des
        limal = i;
        break
    end;
end;
limb1=limal+3000;
k1= 0:(limb1-limal);
y1cas = y3(limal :limb1);
y1casn = y1cas;
%*****
des = (mayabs2/100)*20;
for i=1:lon2(1);
    if abs(y4(i)) > des
        lima2 = i;
        break
    end;
end;
limb2=lima2+3000 ;
k2= 0:(limb2-lima2);
y2cas = y4(lima2 :limb2);
y2casn = y2cas;
%*****
des = (mayabs3/100)*20;
for i=1:lon3(1);
    if abs(y5(i)) > des

```

```

        lima3 = i;
        break
    end;
end;
limb3=lima3+3000;
k3= 0:(limb3-lima3);
y3cas = yc3(lima3 :limb3);
y3casn = y3cas;
% *****
des = (mayabs4/100)*20;
for i=1:lon4(1);
    if abs(yw1(i)) > des
        lima4 = i;
        break
    end;
end;
limb4= lima4+3000;
k4= 0:(limb4-lima4);
y1wal = yw1(lima4 :limb4);
y1waln = y1wal;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb1 - lima1)+1;
    y1casn(i)=y1casn(i)/mayabs1;
end;
for i=1:(limb2 - lima2)+1;
    y2casn(i)=y2casn(i)/mayabs2;
end;
for i=1:(limb3 - lima3)+1;
    y3casn(i)=y3casn(i)/mayabs3;
end;
for i=1:(limb4 - lima4)+1;
    y1waln(i)=y1waln(i)/mayabs4;
end;

```

Ejecucion del programa compa01:

Longitud del vector yc1: 181929 Longitud del vector yc2: 132312 Longitud del vector yc3: 93721 Longitud del vector yw1: 49617
--

- **Programa 3.3.2.-** Analiza cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un tercero de otro locutor (Walter), este análisis es en el tiempo y utiliza el parámetro MAGNITUD.

```

%Programa que calcula magnitud de cuatro vectores
%Programa compa02
%Utiliza archivos: y1casn y2casn y3casn y1waln
%Considera tamaño de ventana e incremento de ventana
clc;
fprintf('Inicio de programa n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn);
tamano1 = tam(1);
tam = size(y2casn);
tamano2 = tam(1);
tam = size(y3casn);
tamano3 = tam(1);
tam = size(y1waln);
tamano4 = tam(1);
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n FIN DEL PROGRAMA \n');
        fprintf('\n Presione enter para mostrar tabla...\n');
        pause;
        break;
    end;
    magnitud1 = 0;
    magnitud2 = 0;
    magnitud3 = 0;
    magnitud4 = 0;
    for j=j1 : j2;
        magnitud1 = magnitud1 + abs(y1casn(j));
        magnitud2 = magnitud2 + abs(y2casn(j));
        magnitud3 = magnitud3 + abs(y3casn(j));
        magnitud4 = magnitud4 + abs(y1waln(j));
    end;

```


- **Programa 3.3.3.-** Analiza cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), este análisis es en el tiempo y utiliza el parámetro PROMEDIO.

%Calcula promedios de los cuatro vectores de voz

%Programa compa03

%Archivo normalizado: y1casn y2casn y3casn y4waln

%Considera tamaño e incremento de ventana

```

clc;
fprintf('Inicio de programa \n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn);
tamano1 = tam(1);
tam = size(y2casn);
tamano2 = tam(1);
tam = size(y3casn);
tamano3 = tam(1);
tam = size(y1waln);
tamano4 = tam(1);
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause;
        break;
    end;
    media1 = 0;
    media2 = 0;
    media3 = 0;
    media4 = 0;
    for j=j1 : j2;
        media1 = media1 + y1casn(j);
        media2 = media2 + y2casn(j);
        media3 = media3 + y3casn(j);
        media4 = media4 + y1waln(j);
    end;
    medias1(i) = media1/vtam;
    medias2(i) = media2/vtam;

```


- **Programa 3.3.4.-** Analiza cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), este análisis es en el tiempo y utiliza el parámetro ENERGIA.

```

%Programa que calcula la energía de cuatro vectores
%Programa compa04
%Archivos normalizados: y1casn y2casn y3casn y4waln
%Considera tamaño de ventana e incremento de ventana
clc;
fprintf('Inicio de programa \n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn);
tamano1 = tam(1);
tam = size(y2casn);
tamano2 = tam(1);
tam = size(y3casn);
tamano3 = tam(1);
tam = size(y1waln);
tamano4 = tam(1);
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause;
        break;
    end;
    energia1 = 0;
    energia2 = 0;
    energia3 = 0;
    energia4 = 0;
    for j=j1 : j2;
        energia1 = energia1 + y1casn(j)*y1casn(j);
        energia2 = energia2 + y2casn(j)*y2casn(j);
        energia3 = energia3 + y3casn(j)*y3casn(j);
        energia4 = energia4 + y1waln(j)*y1waln(j);
    end;
    energias1(i) = energia1/vtam;
    energias2(i) = energia2/vtam;

```

```

energias3(i) = energia3/vtam;
energias4(i) = energia4/vtam;
j1 = j1 + vinc;
j2 = j2 + vinc;
end;
m=i-1;
fprintf('\n\t\t\t ENERGIAS \n');
fprintf('\t\t\t=====\n\n');
fprintf('\t\t\t[Índice] \t[Energia1] \t[Energia2] \t[Energia3] \t[Energia4]\n');
for i=1:m;
    fprintf('\t\t\t %d \t %6.2f\t %6.2ft \t %6.2ft \t %6.2fn',i,
energias1(i),energias2(i),energias3(i),energias4(i));
end;

```

Ejecución del programa compa04:

```

Inicio de programa
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
FIN DEL CALCULO

```

Presione enter para visualizar tabla...

ENERGIAS

=====

[Índice]	[Energia1]	[Energia2]	[Energia3]	[Energia4]
1	0.07	0.06	0.09	0.07
2	0.09	0.09	0.12	0.11
3	0.08	0.09	0.12	0.15
4	0.08	0.10	0.10	0.18
5	0.07	0.08	0.08	0.17
6	0.06	0.04	0.07	0.15
7	0.06	0.02	0.04	0.12
8	0.04	0.01	0.02	0.07
9	0.06	0.01	0.01	0.05

- **Programa 3.3.5.-** Analiza cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), este análisis es en el tiempo y utiliza el parámetro CRUCES POR CERO.

```

%Programa calcula cruces por cero de cuatro vectores
%Programa compa05
%Archivos normalizados: y1casn y2casn y3casn y1waln
% Considera tamaño de ventana e incremento de ventana
clc;
fprintf('Inicio de programa \n');
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
tam = size(y1casn);
tamano1 = tam(1);
tam = size(y2casn);
tamano2 = tam(1);
tam = size(y3casn);
tamano3 = tam(1);
tam = size(y1waln);
tamano4 = tam(1);
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause;
        break;
    end;
    estado1 = 0;
    cruce1 = 0;
    estado2 = 0;
    cruce2 = 0;
    estado3 = 0;
    cruce3 = 0;
    estado4 = 0;
    cruce4 = 0;
    for j=j1 : j2;
        if y1casn(j) ~=0
            if estado1==0
                estado1=y1casn(j);
            else
                if ((y1casn(j)>0)& (estado1<0))| ((y1casn(j)<0) & (estado1>0))
                    cruce1 = cruce1 +1;
                    estado1 = y1casn(j);
                end
            end
        end
    end
end

```

```

        end;
    end;
end;
if y2casn(j) ~=0
    if estado2==0
        estado2= y2casn(j);
    else
        if ((y2casn(j)>0)& (estado2<0))| ((y2casn(j)<0) & (estado2>0))
            cruce2 = cruce2 +1;
            estado2 = y2casn(j);
        end;
    end;
end;
end;
if y3casn(j) ~=0
    if estado3==0
        estado3= y3casn(j);
    else
        if ((y3casn(j)>0)& (estado3<0))| ((y3casn(j)<0) & (estado3>0))
            cruce3 = cruce3 +1;
            estado3 = y3casn(j);
        end;
    end;
end;
end;
if y1waln(j) ~=0
    if estado4==0
        estado4=y1waln(j);
    else
        if ((y1waln(j)>0)& (estado4<0))| ((y1waln(j)<0) & (estado4>0))
            cruce4 = cruce4 +1;
            estado4 = y1waln(j);
        end;
    end;
end;
end;
cruces1(i) = cruce1;
cruces2(i) = cruce2;
cruces3(i) = cruce3;
cruces4(i) = cruce4;
j1 = j1 + vinc;
j2 = j2 + vinc;
end;
m=i-1;
fprintf('\n CRUCES POR CERO \n');
fprintf('=====\n');
fprintf(' [Indice] \t[Cruces1] \t[Cruces2] \t[Cruces3] \t[Cruces4]\n');
for i=1:m;

```

```
fprintf(' %d \t%3d \t%3d \t%3d \t%3d \n',i,
cruces1(i),cruces2(i),cruces3(i),cruces4(i));
end;
```

Ejecución del programa compa05:

```
Inicio de programa
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300

FIN DEL CALCULO

Presione enter para visualizar tabla...

CRUCES POR CERO
=====
```

[Índice]	[Cruces1]	[Cruces2]	[Cruces3]	[Cruces4]
1	17	26	22	21
2	20	26	26	23
3	19	26	20	23
4	19	29	20	24
5	19	26	19	28
6	19	22	19	27
7	15	18	14	25
8	9	13	10	20
9	8	13	4	16

- **Programa 3.3.6.-** Compara cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), esta comparación es en el tiempo y utiliza el parámetro VALORES MAXIMOS.

%Programa que calcula máximos de cuatro vectores

%Programa compa06

%Archivos normalizados: y1casn y2casn y3casn y1waln

%Considera tamaño de ventana e incremento de ventana

clc;

fprintf('Inicio de programa \n');

vtam = input('ingrese tamaño de la ventana ==> ');

vinc = input('ingrese incremento de la ventana --> ');

tam = size(y1casn); tamano1 = tam(1);

tam = size(y2casn); tamano2 = tam(1);

tam = size(y3casn); tamano3 = tam(1);

tam = size(y1waln); tamano4 = tam(1);

```

j1=1; j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause; break;
    end;
    estado1 = -1;
    anterior1 = y1casn(j1);
    maximo1=0;
    estado2 = -1;
    anterior2 = y2casn(j1);
    maximo2=0;
    estado3 = -1;
    anterior3 = y3casn(j1);
    maximo3=0;
    estado4 = -1;
    anterior4 = y1waln(j1);
    maximo4=0;
    for j=j1+1 : j2;
        if ((y1casn(j)<anterior1)&(estado1==1))
            maximo1 = maximo1 + 1;
            estado1= -1;
        else
            if((y1casn(j)> anterior1)& (estado1== -1))
                estado1= 1;
            end;
        end;
        anterior1= y1casn(j);
        if ((y2casn(j)<anterior2)&(estado2==1))
            maximo2 = maximo2 + 1;
            estado2= -1;
        else
            if((y2casn(j)> anterior2)& (estado2== -1))
                estado2= 1;
            end;
        end;
        anterior2=y2casn(j);
        if ((y3casn(j)<anterior3)&(estado3==1))
            maximo3 = maximo3 + 1;
            estado3= -1;
        else
            if((y3casn(j)> anterior3)& (estado3== -1))
                estado3= 1;
            end;
        end;
        anterior3=y3casn(j);
    end;
end;

```

```

if ((y1 waln(j)<anterior4)&(estado4==1))
    maximo4 = maximo4 + 1;
    estado4= -1;
else
    if((y1 waln(j)> anterior4)& (estado4==1))
        estado4= 1;
    end;
end;
anterior4= y1 waln(j);
end;
maximos1(i)= maximo1; maximos2(i)= maximo2;
maximos3(i)= maximo3; maximos4(i)= maximo4;
j1 = j1 + vinc;
j2 = j2 + vinc;
end;
m=i-1;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf(' ----- - -- \n');
fprintf(' [Índice] \t[Maximos1] \t[Maximos2] \t[Maximos3] \t[Maximos4]\n');
for i=1:m;
    fprintf(' %d      %d\t      %d\t      %d\t      %d\n',i,
maximos1(i),maximos2(i),maximos3(i),maximos4(i));
end;

```

Ejecución del programa compa06:

```

Inicio de programa
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
FIN DEL CALCULO
Presione enter para visualizar tabla...
MAXIMOS DE LA SENAL
=====
[Índice] [Maximos1] [Maximos2] [Maximos3] [Maximos4]
1      43      47      36      51
2      41      42      41      42
3      40      45      43      37
4      42      44      34      41
5      23      25      22      40
6      27      24      20      38
7      28      27      28      24
8      24      20      30      20
9      27      33      39      24

```

- **Programa 3.3.7.-** Compara cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), esta comparación es en la frecuencia y utiliza el parámetro MAXIMOS EN FRECUENCIA.

```

%Halla máximos de frecuencia de cuatro vectores
%Programa compa07
%Archivos normalizados: y1casn y2casn y3casn y4waln
clc;
fprintf('Inicio de programa \n');
%PRIMER VECTOR
vc1=y1casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc1f=filter(b,a,vc1);
w=(0:255)/256*(Fs/2);
y1casnf=vc1f';
Fy1nf=fft(y1casnf,512);
%*****HALLO MAXIMOS DE FRECUENCIA*****
estado = -1;
anterior = Fy1nf(1);
contmax=0;
for i=2 : 256;
    if ((Fy1nf(i)<anterior)&(estado ==1))
        contmax = contmax +1;
        vcon(contmax)=i-1;
        estado = -1;
    else
        if((Fy1nf(i)> anterior)& (estado ==-1))
            estado = 1;
        end;
    end;
    anterior = Fy1nf(i);
end;
for i= 1:contmax;
    kk=vcon(i);
    mayores(i)=abs(Fy1nf(kk));
    frec(i)=w(kk);
end;
[mayor1,jj]=max(mayores);
K1=0;
for i=1:m;
    if (mayores(i)>0.30*mayor1)
        K1=K1+1;
        mayores1(K1)=mayores(i);
        frec1(K1)=frec(i);
    end;
end;

```

```

    end;
end;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf('===== \n');
fprintf(' [Índice] \t[Frecuencia]\t[Magnitud]\n');
for i=1:K1;
    fprintf(' %d\t \t%8.2ft%6.2f\n',i,frec1(i),mayores1(i));
end;
%SEGUNDO VECTOR
vc2=y2casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc2f=filter(b,a,vc2);
w=(0:255)/256*(Fs/2);
y2casnf=vc2f;
Fy2nf = fft(y2casnf,512);
pause;
%*****HALLO MAXIMOS DE FRECUENCIA*****
estado = -1;
anterior = Fy2nf(1);
contmax=0;
for i=2 : 256;
    if ((Fy2nf(i)<anterior)&(estado ==1))
        contmax = contmax+1;
        vcon(contmax)=i-1;
        estado = -1;
    else
        if((Fy2nf(i)> anterior)& (estado ==-1))
            estado= 1;
        end;
    end;
    anterior=Fy2nf(i);
end;
for i= 1:contmax;
    kk=vcon(i);
    mayores2(i)=abs(Fy2nf(kk));
    frec(i)=w(kk);
end;
[mayor2,jj]=max(mayores2);
k1=0;
for i=1:m;
    if (mayores2(i)>0.30* mayor2)
        k1= k1+1;
        mayores12(k1)=mayores2(i);
        frec12(k1)=frec(i);
    end;
end;
end;

```

```

fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf(' =====\n');
fprintf(' [Índice] \t[Frecuencia]\t[Magnitud]\n');
for i=1:k1;
    fprintf(' %d\t \t%8.2f\t%6.2f\n',i,frec12(i),mayores12(i));
end;
%*****TERCER VECTOR*****
vc3=y3casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc3f=filter(b,a,vc3);
w=(0:255)/256*(Fs/2);
y3casnf=vc3f;
Fy3nf = fft(y3casnf,512);
%*****HALLO MAXIMOS DE FRECUENCIA*****
estado = -1;
anterior = Fy3nf(1);
contmax=0;
for i=2 : 256;
    if((Fy3nf(i)<anterior)&(estado ==1))
        contmax = contmax+1;
        vcon(contmax)=i-1;
        estado = -1;
    else
        if((Fy3nf(i)> anterior)& (estado ==-1))
            estado = 1;
        end;
    end;
    anterior=Fy3nf(i);
end;
for i= 1:contmax;
    kk=vcon(i);
    mayores3(i)=abs(Fy3nf(kk));
    frec(i)=w(kk);
end;
[mayor3,jj]=max(mayores3);
k1=0;
for i=1:m;
    if (mayores3(i)>0.30*mayor3)
        k1=k1+1;
        mayores13(k1)=mayores3(i);
        frec13(k1)=frec(i);
    end;
end;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf(' =====\n');
fprintf(' [Índice] \t[Frecuencia]\t[Magnitud]\n');

```

```

for i=1:k1;
    fprintf(' %d\t \t%8.2ft%6.2f\n',i,frec13(i),mayores13(i));
end;
%****CUARTO VECTOR****
vc4= y1 waln';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc4f=filter(b,a,vc4);
w=(0:255)/256*(Fs/2);
y1 walnf=vc4f';
Fy4nf = fft(y1 walnf,512);
%*****HALLO MAXIMOS DE FRECUENCIA*****
estado = -1;
anterior = Fy4nf(1);
contmax=0;
for i=2 : 256;
    if ((Fy4nf(i)<anterior)&(estado ==1))
        contmax = contmax+1;
        vcon(contmax)=i-1;
        estado = -1;
    else
        if((Fy4nf(i)> anterior)& (estado ==-1))
            estado = 1;
        end;
    end;
    anterior=Fy4nf(i);
end;
for i= 1:contmax;
    kk=vcon(i);
    mayores4(i)=abs(Fy4nf(kk));
    frec(i)=w(kk);
end;
[mayor4,jj]=max(mayores4);
k1=0;
for i=1:m;
    if (mayores4(i)>0.30*mayor4)
        k1=k1+1;
        mayores14(k1)=mayores4(i);
        frec14(k1)=frec(i);
    end;
end;
fprintf('\n MAXIMOS DE LA SENAL \n');
fprintf(' =====\n');
fprintf(' [Índice] \t[Frecuencia]\t[Magnitud]\n');
for i=1:k1;
    fprintf(' %d\t \t%8.2ft%6.2f\n',i,frec14(i),mayores14(i));
end;

```

Ejecución del programa compa07:

Inicio de programa

MAXIMOS DE LA SENAL(cas1)

[Índice]	[Frecuencia]	[Magnitud]
1	129.20	21.89
2	215.33	21.58
3	387.60	24.71

MAXIMOS DE LA SENAL (cas2)

[Índice]	[Frecuencia]	[Magnitud]
1	301.46	19.12
2	430.66	13.94
3	516.80	41.90

MAXIMOS DE LA SENAL (cas3)

[Índice]	[Frecuencia]	[Magnitud]
1	129.20	19.99
2	387.60	16.80
3	473.73	35.00
4	732.13	13.58

MAXIMOS DE LA SENAL (wall)

[Índice]	[Frecuencia]	[Magnitud]
1	129.20	14.53
2	301.46	31.57
3	430.66	9.57
4	559.86	12.79

- **Programa 3.3.7.-** Compara cuatro archivos, los tres primeros del mismo locutor (Castañeda) y un cuarto de otro locutor (Walter), esta comparación es en la frecuencia y utiliza el parámetro PROMEDIO DE FRECUENCIAS.

```

%Halla promedio de frecuencia de cuatro vectores
%Programa compa08
%arch. normalizados: y1casn y2casn y3casn y1waln
clc;
fprintf('Inicio de programa \n');
%PRIMER VECTOR
vc1 = y1casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc1f=filter(b,a,vc1);
y1casnf= vc1f';
Fyn1f = fft(y1casnf,512);
promedio1 = mean(abs(Fyn1f))*100;
%SEGUNDO VECTOR
vc2 = y2casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc2f=filter(b,a,vc2);
y2casnf=vc2f';
Fyn2f = fft(y2casnf,512);
promedio2 = mean(abs(Fyn2f))*100;
%TERCER VECTOR
vc3= y3casn';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc3f=filter(b,a,vc3);
y3casnf=vc3f';
Fyn3f = fft(y3casnf,512);
promedio3 = mean(abs(Fyn3f))*100;
%CUARTO VECTOR
vc4= y1waln';
[b,a]=butter(4,[100 1800]*2/Fs);
[H,w]=freqz(b,a,512);
vc4f=filter(b,a,vc4);
y1walnf=vc4f';
Fyn4f = fft(y1walnf,512);
promedio4 = mean(abs(Fyn4f))*100;
fprintf('\n PROMEDIOS DE FRECUENCIA \n');
fprintf('      ===== \n');
fprintf(' \t[Promedio1] \t[Promedio2] \t[Promedio3] \t[Promedio4]\n');

```

```
fprintf(' %8.2ft %8.2ft %8.2ft
%8.2ft\n',promedio1,promedio2,promedio3,promedio4);
```

Ejecución del programa compa08:

```
Inicio de programa

PROMEDIOS DE FRECUENCIA
=====
      [Promedio1] [Promedio2] [Promedio3] [Promedio4]
      160.04      151.74      189.89      159.90
>>
```

3.4 Reconocimiento de locutor.

- **Programa 3.4.1.-** Lee archivos de audio y genera los archivos con los dígitos de los tres locutores sin silencios y normalizados.

% Programa de reconocimiento de locutor

% Programa: reconoce01

% Lee archivos de audio y genera archivos de dígitos

% separados y normalizados

clc; echo off;

%***PROCEDIMIENTO DE LECTURA*******

[yc1, Fs, Bits] = wavread ('a:cas1.wav');

[yw1, Fs, Bits] = wavread ('a:ken1.wav');

[yh1, Fs, Bits] = wavread ('a:rob1.wav');

[yi1, Fs, Bits] = wavread ('a:lau1.wav');

[ym1, Fs, Bits] = wavread ('a:mar1.wav');

lon1 =size(yc1);lon2 =size(yw1);lon3 = size(yh1);lon4 = size(yi1);lon5 = size(ym1);

fprintf('\n Longitud del vector yc1: %d \n',lon1(1));

fprintf ('\n Longitud del vector yw1: %d \n',lon2(1));

fprintf('\n Longitud del vector yh1: %d \n',lon3(1));

fprintf ('\n Longitud del vector yi1: %d \n',lon4(1));

fprintf('\n Longitud del vector ym1: %d \n',lon5(1));

%***CALCULA MAYOR MENOR Y MAYOR ABSOLUTO*******

[mayor1,jj]= max(yc1); [menor1,jj]= min(yc1);

if abs(mayor1) > abs(menor1)

 mayabs1= abs(mayor1);

```

else
    mayabs1 = abs(menor1);
end;
[mayor2,jj]= max(yw1); [menor2,jj]= min(yw1);
if abs(mayor2) > abs(menor2)
    mayabs2= abs(mayor2);
else
    mayabs2 = abs(menor2);
end;
[mayor3,jj]= max(yh1); [menor3,jj]= min(yh1);
if abs(mayor3) > abs(menor3)
    mayabs3= abs(mayor3);
else
    mayabs3 = abs(menor3);
end;
[mayor4,jj]= max(yi1); [menor4,jj]= min(yi1);
if abs(mayor4) > abs(menor4)
    mayabs4= abs(mayor4);
else
    mayabs4 = abs(menor4);
end;
[mayor5,jj]= max(ym1); [menor5,jj]= min(ym1);
if abs(mayor5) > abs(menor5)
    mayabs5= abs(mayor5);
else
    mayabs5 = abs(menor5);
end;
%*****HALLA ARCHIVOS SIN SILENCIOS*****
des = (mayabs1/100)*20;
for i=1:lon1(1);
    if abs(yc1(i)) > des
        lima1 = i;
        break
    end;
end;
limb1=lima1+3000; k1= 0:(limb1-lima1);
y1cas = yc1(lima1 :limb1);
y1casn = y1cas;
%*****
des = (mayabs2/100)*20;
for i=1:lon2(1);
    if abs(yw1(i)) > des
        lima2 = i;

```

```

        break
    end;
end;
limb2= lima2+3000; k2= 0:(limb2-lima2);
ylwal = ywl(lima2 :limb2);
ylwaln = ylwal;
%*****
des = (mayabs3/100)*20;
for i=1:lon3(1);
    if abs(yh1(i)) > des
        lima3 = i;
        break
    end;
end;
limb3=lima3+3000; k3= 0:(limb3-lima3);
ylhar = yh1(lima3 :limb3);
ylharn = ylhar;
%*****
des = (mayabs4/100)*20;
for i=1:lon4(1);
    if abs(yil(i)) > des
        lima4 = i;
        break
    end;
end;
limb4=lima4+3000; k4= 0:(limb4-lima4);
ylisa = yil(lima4 :limb4);
ylisan = ylisa;
%*****
des = (mayabs5/100)*20;
for i=1:lon5(1);
    if abs(ym1(i)) > des
        lima5 = i;
        break
    end;
end;
limb5=lima5+3000; k5=0:(limb5-lima5);
ylmic = yml(lima5 :limb5);
ylmicn = ylmic;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb1 - lima1)+1;
    ylcasn(i)=ylcasn(i)/mayabs1;
end;

```

```

for i=1:(limb2 - lima2)+1;
    y1waln(i)=y1waln(i)/mayabs2;
end;
for i=1:(limb3 - lima3)+1;
    y1harn(i)=y1harn(i)/mayabs3;
end;
for i=1:(limb4 - lima4)+1;
    y1isan(i)=y1isan(i)/mayabs4;
end;
for i=1:(limb5 - lima5)+1;
    y1micn(i)=y1micn(i)/mayabs5;
end;
fprintf('\n ARCHIVOS CREADOS');

```

Ejecución del programa reconoce01:

```

Longitud del vector yc1: 50912

Longitud del vector yw1: 70176

Longitud del vector yh1: 53664

Longitud del vector yi1: 45408

Longitud del vector ym1: 46784

ARCHIVOS CREADOS

```

- **Programa 3.4.2.-** Genera los patrones de voz de cada locutor y los coloca en una estructura de datos.

```

% Programa de reconocimiento de locutor
% Programa: reconoce02
% Crea vector de patrones de voz
% Procesa:
%   Entrada: y1casn, y1waln
%   Salida:
%       p1 patron 1
%       p2 patron 2
%       p3 patron 3
clc;
fprintf('\n Inicio de Programa \n');
a= size(y1casn);tamano1 = a(1);

```

```

b= size(y1waln);tamano2 = b(1);
c= size(y1harn);tamano3 = c(1);
d= size(y1isan);tamano4 = d(1);
e= size(y1micn);tamano5 = e(1);
%*****Calculo de magnitudes*****
vtam = input('ingrese tamaño de la ventana ==> ');
vinc = input('ingrese incremento de la ventana ==> ');
%****Para "dos" de Castañeda****
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano1)
        fprintf('\n Presione enter ..\n');
        pause;
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1casn(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
sumal(1)=0;
for i=1:3;
    sumal(1) = sumal(1) + magnitudes(i);
end;
%****Para "dos" de kenny****
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano2)
        fprintf('\n Presione enter ..\n');
        pause;
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1waln(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
sumal(2)=0;
for i=1:3;

```

```

    suma1(2)= suma1(2)+magnitudes(i);
end;
%***Para "dos" de Robinson****
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano3)
        fprintf('\n Presione enter .\n');
        pause;
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1harn(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
suma1(3)=0;
for i=1:3;
    suma1(3)= suma1(3)+magnitudes(i);
end;
%***Para "dos" de Laureano****
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano4)
        fprintf('\n Presione enter .\n');
        pause;
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1isan(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
suma1(4)=0;
for i=1:3;
    suma1(4)= suma1(4)+magnitudes(i);
end;
%***Para "dos" de Marcos****
j1=1;
j2=vtam;

```

```

for i= 1 :100;
    if(j2 >tamano5)
        fprintf('\n Presione enter .\n');
        pause;
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(y1micn(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
sumal(5)=0;
for i=1:3;
    sumal(5)= sumal(5)+magnitudes(i);
end;
%*****
fprintf('\n Primer parámetro completo \n');
%*****Calculo de máximos de tiempo*****
%****Para "dos" de Castañeda, Kenny, Robinson, Laureano y Marcos****
j1=1;
j2=vtam;
for i= 1 :20;
    if(j2 >tamano1)
        fprintf('\n FIN DEL CALCULO \n');
        fprintf('\n Presione enter para visualizar tabla...\n');
        pause;
        break;
    end;
    estado1 = -1;
    anterior1 = y1casn(j1);
    maximo1=0;
    estado2 = -1;
    anterior2 = y1waln(j1);
    maximo2=0;
    estado3 = -1;
    anterior3 = y1harn(j1);
    maximo3=0;
    estado4 = -1;
    anterior4 = y1isan(j1);
    maximo4=0;
    estado5 = -1;
    anterior5 = y1micn(j1);
    maximo5=0;
    for j=j1+1 : j2;

```

```

if ((y1casn(j)<anterior1)&(estado1==1))
    maximo1 = maximo1 + 1;
    estado1= -1;
else
    if((y1casn(j)> anterior1)& (estado1== -1))
        estado1= 1;
    end;
end;
anterior1= y1casn(j);
if ((y1waln(j)<anterior2)&(estado2==1))
    maximo2 = maximo2 + 1;
    estado2= -1;
else
    if((y1waln(j)> anterior2)& (estado2== -1))
        estado2= 1;
    end;
end;
anterior2=y1waln(j);
if ((y1harn(j)<anterior3)&(estado3==1))
    maximo3 = maximo3 + 1;
    estado3= -1;
else
    if((y1harn(j)> anterior3)& (estado3== -1))
        estado3= 1;
    end;
end;
anterior3=y1harn(j);
if ((y1isan(j)<anterior4)&(estado4==1))
    maximo4 = maximo4 + 1;
    estado4= -1;
else
    if((y1isan(j)> anterior4)& (estado4== -1))
        estado4= 1;
    end;
end;
anterior4= y1isan(j);
if ((y1micn(j)<anterior5)&(estado5==1))
    maximo5 = maximo5 + 1;
    estado5= -1;
else
    if((y1micn(j)> anterior5)& (estado5== -1))
        estado5= 1;
    end;
end;
anterior5= y1micn(j);
end;
maximos1(i)= maximo1;

```

```

    maximos2(i)= maximo2;
    maximos3(i)= maximo3;
    maximos4(i)= maximo4;
    maximos5(i)= maximo5;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
media1 = 0;
media2 = 0;
media3 = 0;
media4 = 0;
media5 = 0;
for k = 1:3;
    media1 = media1 + maximos1(k);
    media2 = media2 + maximos2(k);
    media3 = media3 + maximos3(k);
    media4 = media4 + maximos4(k);
    media5 = media5 + maximos5(k);
end;
suma2(1)=media1;
suma2(2)=media2;
suma2(3)=media3;
suma2(4)=media4;
suma2(5)=media5;
fprintf('\n Segundo Parámetro completo \n'):
%*****Calculo de frecuencias Formantes*****
%****Para "dos" de Castañeda****
temp1=y1casn';
temp2=y1waln';
temp3=y1harn';
temp4=y1isan';
temp5=y1micn';
[q,p]= cheby1(12,0.005,2000*2/Fs);
[H,w]= freqz(q,p,512);
temp1f=filter(q,p,temp1);
temp2f=filter(q,p,temp2);
temp3f=filter(q,p,temp3);
temp4f=filter(q,p,temp4);
temp5f=filter(q,p,temp5);
y1casnf=temp1f;
y1walnf=temp2f;
y1harnf=temp3f;
y1isanf=temp4f;
y1micnf=temp5f;
F1=fft(y1casnf,512);
F2=fft(y1walnf,512);
F3=fft(y1harnf,512);

```

```

F4=fft(y1isanf,512);
F5=fft(y1micnf,512);
cont = 0; acum1=0; estado = -1; anterior = abs(F1(1));
for i =2:512;
    if(abs(F1(i))<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado =-1;
    else
        if(abs(F1(i))>anterior)& (estado == -1)
            estado =1;
        end;
    end;
    anterior= abs(F1(i));
end;
[maximo1 ,jj] = max(abs(F1));
cuenta3=0;
for i= 1:cont;
    k=pos(i);
    if abs(F1(k))>(0.30*maximo1)
        cuenta3 = cuenta3 +1;
        if (cuenta3<=3)
            acum1 = acum1+abs(F1(k));
        end;
    end;
end;
suma3(1)=acum1;
%****Para "dos" de Kenny****
cont=0; acum2=0; estado = -1; anterior = abs(F2(1));
for i =2:512;
    if(abs(F2(i))<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado=-1;
    else
        if(abs(F2(i))>anterior)& (estado == -1)
            estado=1;
        end;
    end;
    anterior=abs(F2(i));
end;
[maximo2 ,jj] = max(abs(F2));
cuenta3 = 0;
for i= 1:cont;
    k=pos(i);
    if(abs(F2(k))>0.30*maximo2)
        cuenta3 = cuenta3 +1;
    end;
end;

```

```

    if (cuenta3 <= 3)
        acum2 = acum2+ abs(F2(k));
    end;
end;
end;
suma3(2)=acum2;
%****Para "dos" de Robinsón****
cont =0; acum3=0; estado = -1; anterior = abs(F3(1));
for i =2:512;
    if(abs(F3(i))<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado =-1;
    else
        if(abs(F3(i))>anterior)& (estado == -1)
            estado =1;
        end;
    end;
    anterior= abs(F3(i));
end;
[maximo3,jj] = max(abs(F3));
cuenta3 = 0;
for i= 1:cont;
    k=pos(i);
    if abs(F3(k))>(0.30*maximo3)
        cuenta3 = cuenta3 +1;
        if (cuenta3 <= 3)
            acum3 = acum3+ abs(F3(k));
        end;
    end;
end;
end;
suma3(3)=acum3;
%****Para "dos" de Laureano****
cont =0; acum4=0; estado = -1; anterior = abs(F4(1));
for i =2:512;
    if(abs(F4(i))<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado =-1;
    else
        if(abs(F4(i))>anterior)& (estado == -1)
            estado=1;
        end;
    end;
    anterior=abs(F4(i));
end;
end;
[maximo4,jj] = max(abs(F4));

```

```

cuenta3 = 0;
for i= 1:cont;
    k=pos(i);
    if(abs(F4(k))>0.30*maximo4)
        cuenta3 = cuenta3 +1;
        if (cuenta3 <= 3)
            acum4 = acum4+ abs(F4(k));
        end;
    end;
end;
suma3(4)=acum4;
%****Para "dos" de Marcos****
cont=0; acum5=0; estado = -1; anterior = abs(F5(1));
for i =2:512;
    if(abs(F5(i)<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado=-1;
    else
        if(abs(F5(i))>anterior)& (estado == -1)
            estado=1;
        end;
    end;
    anterior=abs(F5(i));
end;
[maximo5,jj] = max(abs(F5));
cuenta3 = 0;
for i= 1:cont;
    k=pos(i);
    if(abs(F5(k))>0.30*maximo5)
        cuenta3 = cuenta3 +1;
        if (cuenta3 <= 3)
            acum5 = acum5+ abs(F5(k));
        end;
    end;
end;
suma3(5)=acum5;
fprintf('\n Tercer Parámetro completo \n');
fprintf('\n Generando vector de patrones \n');
for i= 1:5;
    p1(i) = suma1(i);
    p2(i) = suma2(i);
    p3(i) = suma3(i);
end;
fprintf('\n Vector de Patrones Generado \n');
fprintf(' \t[Magnitudes] \t[Maximos t] \t[Maximos f] \n');
for i= 1:5;

```

```
fprintf(' %9.4ft %9.4ft %9.4f\n',p1(i),p2(i),p3(i));
end;
```

Ejecución del programa reconoce02:

```
ingrese tamaño de la ventana ==> 500
ingrese incremento de la ventana ==> 300
```

Presione enter .

Primer parámetro completo

Segundo Parámetro completo

Tercer Parámetro completo

Generando vector de patrones

Vector de Patrones Generado

[Magnitudes]	[Máximos t]	[Máximos f]	
0.9891	186.0000	158.2483	PATRON 1
0.8382	97.0000	127.5137	PATRON 2
0.5816	49.0000	97.2312	PATRON 3
0.6383	70.0000	63.3660	PATRON 4
0.7293	157.0000	82.1064	PATRON 5

- **Programa 3.4.3.-** Reconoce locutor, programa final.

```
% Programa de reconocimiento de locutor
```

```
% Programa: reconoce03.m
```

```
% Compara patrón y reconoce locutor
```

```
clc;
```

```
echo off;
```

```
%*****PROCEDIMIENTO DE LECTURA*****
```

```
[y, Fs, Bits] = wavread ('a:cas10.wav');
```

```
ta = size(y);
```

```
tam = ta(1);
```

```
%*****CALCULA MAYOR MENOR Y MAYOR ABSOLUTO*****
```

```
[mayor,jj]=max(y);
```

```
[menor,jj]=min(y);
```

```
if abs(mayor) > abs(menor)
```

```
    mayabs = abs(mayor);
```

```
else
```

```
    mayabs = abs (menor);
```

```

end;
%*****HALLA ARCHIVOS SIN SILENCIOS*****
des = (mayabs/100)*20;
for i=1:tam;
    if abs(y(i)) > des
        lima = i;
        break
    end;
end;
limb=lima+3000;
k= 0:(limb-lima);
ys = y(lima :limb);
b = size(ys);
ysn = ys;
%*****NORMALIZA DIGITOS DOS*****
for i=1:(limb - lima)+1;
    ysn(i)=ysn(i)/mayabs;
end;
c= size(ysn);tamano = c(1);
%vtam = input('Ingrese tamaño de la ventana ==>');
%vinc = input('Ingrese incremento de ventana ==>');
vtam = 500;
vinc= 300;
%*****Calculo de Magnitudes*****
%****Para "dos" de locutor****
j1=1;
j2=vtam;
for i= 1 :100;
    if(j2 >tamano)
        fprintf('\n Calculo de magnitudes ..\n');
        break;
    end;
    magnitud = 0;
    for j=j1 : j2;
        magnitud = magnitud + abs(ysn(j));
    end;
    magnitudes(i) = magnitud/vtam;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
acum=0;
for i=1:3;
    acum = acum + magnitudes(i);
end;
pat1=acum;
%*****Calculo de promedio de frecuencias*****
%****Para "dos" de locutor****

```

```

j1=1;
j2=vtam;
for i= 1 :20;
    if(j2 >tamano l)
        fprintf('\n calculo de maximos \n');
        break;
    end;
    estado1 = -1;
    anterior1 = ysn(j1);
    maximo1=0;
    for j=j1+1 : j2;
        if ((ysn(j)<anterior1)&(estado1==1))
            maximo1 = maximo1 + 1;
            estado1 = -1;
        else
            if((ysn(j)> anterior1)& (estado1==-1))
                estado1= 1;
            end;
        end;
        anterior1= ysn(j);

    end;
    maximos1(i)= maximo1;
    j1 = j1 + vinc;
    j2 = j2 + vinc;
end;
medial = 0;
for k = 1:3;
    medial = medial + maximos1(k);
end;
pat2= medial;
%*****Calculo de frecuencias Formantes*****
%****Para "dos" de locutor****
temp =ysn';
[q,p]= cheby1(12,0.005,2000*2/Fs);
[H,w]= freqz(q,p,512);
tempf = filter(q,p,temp);
ysnf= tempf;
F1 = fft(ysnf,512);
cont=0; acum=0; estado = -1; anterior = abs(F1(1));
for i =2:512;
    if(abs(F1(i))<anterior) & (estado==1)
        cont = cont +1;
        pos(cont) = i-1;
        estado=-1;
    else
        if(abs(F1(i))>anterior)& (estado == -1)

```

```

        estado=1;
    end;
end;
anterior = abs(F1(i));
end;
[maximo ,jj] = max(abs(F1));
cuenta3 = 0;
for i= 1:cont;
    k=pos(i);
    if abs(F1(k))>(maximo*0.30)
        cuenta3 = cuenta3 +1;
        if (cuenta3 <= 3)
            acum = acum+abs(F1(k));
        end;
    end;
end;
pat3=acum;
pat1
pat2
pat3

%****Buscando Locutor****
encontrado = 0;
for i= 1:5;
    p1i= p1(i) -0.20*p1(i); p1s=p1(i)+0.20*p1(i);
    p2i= p2(i) -0.20*p2(i); p2s=p2(i)+0.20*p2(i);
    p3i= p3(i) -0.20*p3(i); p3s=p3(i)+0.20*p3(i);
    cond1=(pat1 >= p1i) & (pat1 <=p1s);
    cond2=(pat2 >= p2i) & (pat2 <=p2s);
    cond3=(pat3 >= p3i) & (pat3 <=p3s);
    if (cond1 & cond2 & cond3)
        encontrado = 1;
        break;
    end;
end;
if (encontrado ==1)
    if (i==1)
        % sound(castaneda,Fs);
        fprintf('castañeda reconocido \n');
    end;
    if(i==2)
        %sound(Kenny,Fs);
        fprintf('Kenny reconocido \n');
    end;
    if (i==3)
        % sound(Robinson,Fs);
        fprintf('Robinsón reconocido \n');
    end;
end;

```

```
end;
if(i==4)
    %sound(Laureano,Fs);
    fprintf('Laureano reconocido \n');
end;
if (i==5)
    % sound(Marcos,Fs);
    fprintf('Marcos reconocido \n');
end;
else
    % sound(noacc,Fs);
    fprintf('No encontrado \n');
end;
```

Ejecución del programa reconoce03:

Calculo de magnitudes

Calculo de máximos

Castañeda reconocido

CAPÍTULO IV RESULTADOS

4.1 Resultados del estudio comparativo de los patrones de tres archivos nombre.wav del mismo locutor y otro de un locutor diferente.

Los resultados los vamos a analizar a partir de la comparación de patrones.

a.- El parámetro magnitud en el dominio del tiempo, programa COMPA02

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	0.23	0.20	0.24	0.2
	0.24	0.24	0.28	0.27
	0.24	0.25	0.29	0.31
	0.24	0.25	0.27	0.35
	0.22	0.22	0.24	0.34
	0.19	0.16	0.21	0.31
	0.19	0.11	0.14	0.28
	0.15	0.07	0.10	0.22
	0.17	0.06	0.09	0.21
SUMA	1.87	1.56	1.86	2.49

Tabla 4.1 Tres vectores magnitud de locutor 1 y un vector de locutor 2

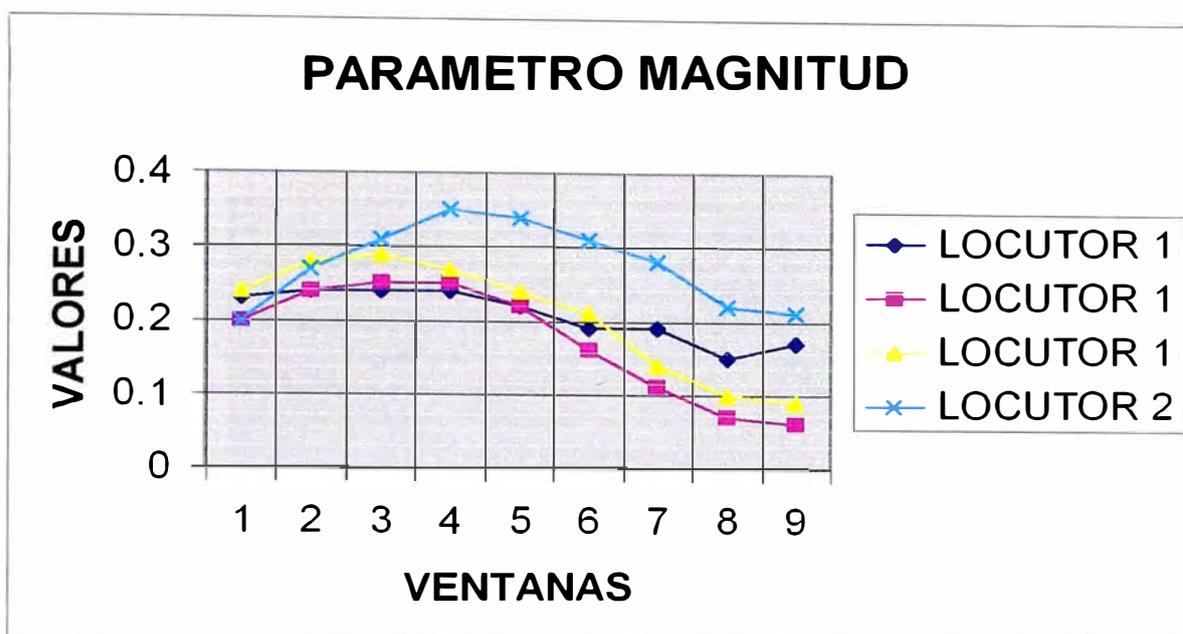


Figura 4.1 Grafico de la tabla 4.1

Comentario

Observamos que hasta la quinta ventana la voz del locutor 1 de los tres archivos tienen un patrón coincidente y se diferencia del locutor 2.

La suma 1 y la suma 3 de locutor 1 coinciden prácticamente y difieren grandemente de locutor 2 **es un parámetro mas que bueno.**

b.- El parámetro media en el dominio del tiempo, PROGRAMA compa03.

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	0.05	0.03	0.03	0.04
	0.05	0.02	0.03	0.05
	0.06	0.03	0.05	0.05
	0.04	0.02	0.03	0.03
	0.05	0.03	0.03	0.03
	0.05	0.01	0.03	0.03
	0.04	0.02	0.02	0.03
	0.05	0.01	0.02	0.04
	0.02	0.00	0.04	0.02
SUMA	0.41	0.17	0.28	0.32

Tabla 4.2 Tres vectores del parámetro media de locutor 1 y uno de locutor 2

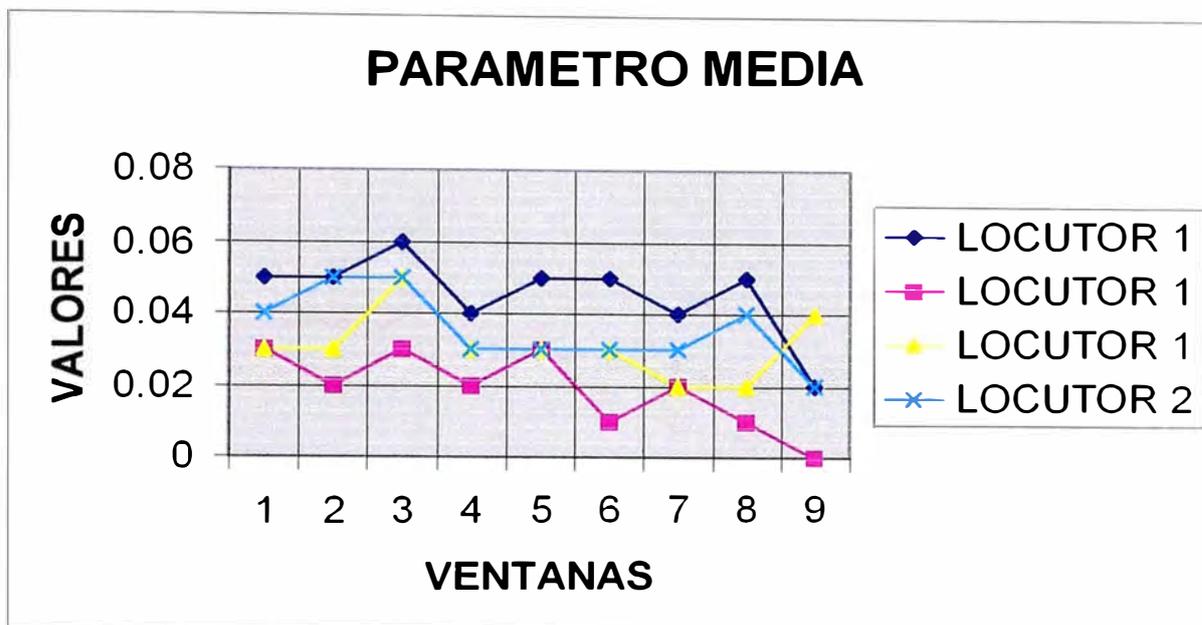


Figura 4.2 Grafico de la tabla 4.2

Comentario

Vemos que este patrón no contiene información relevante para el reconocimiento.

La sumas no coinciden **no es un buen parámetro.**

c.- El parámetro ENERGIA en el dominio del tiempo, PROGRAMA compa04

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	0.07	0.06	0.09	0.07
	0.09	0.09	0.12	0.11
	0.08	0.09	0.12	0.15
	0.08	0.10	0.10	0.18
	0.07	0.08	0.08	0.17
	0.06	0.04	0.07	0.15
	0.06	0.02	0.04	0.12
	0.04	0.01	0.02	0.07
	0.06	0.01	0.01	0.05
SUMA	0.61	0.50	0.65	1.07

Tabla 4.3 Tres vectores de energía del locutor 1 y uno del locutor 2

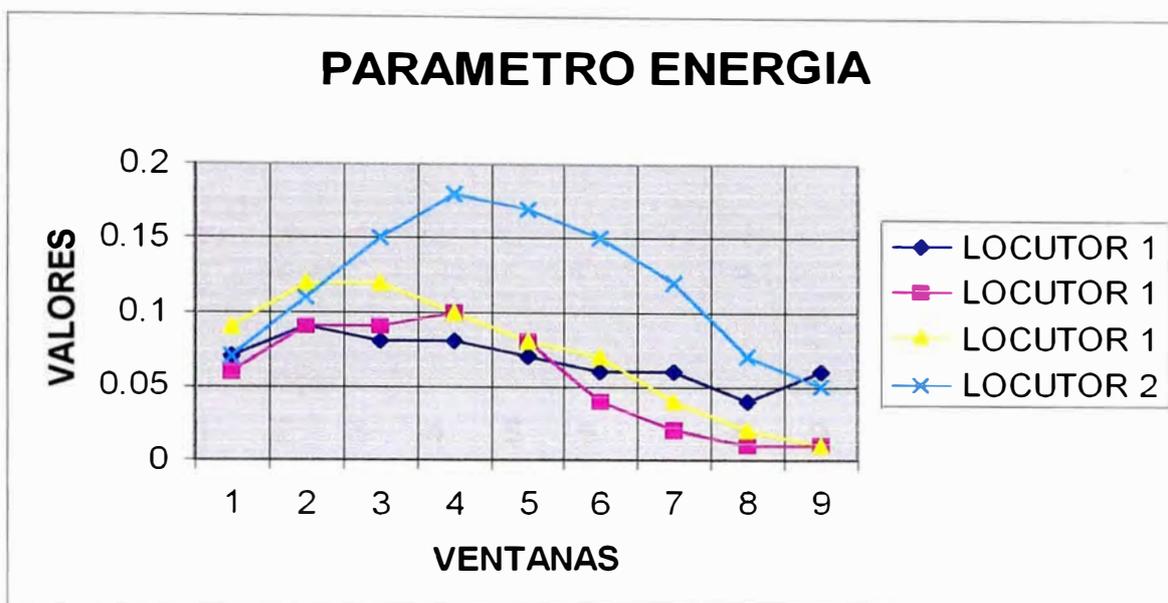


Figura 4.3 Grafico de la tabla 4.3

Comentario

Observamos que la voz del locutor 1 de los tres archivos tienen un patrón más o menos coincidente y se diferencia del locutor 2.

La suma 1 y la suma 3 de locutor 1 coinciden prácticamente y grandemente de locutor 2 es un parámetro bueno.

d.- El parámetro cruces por cero en el dominio del tiempo, PROGRAMA compa05

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	17	26	22	21
	20	26	26	23
	19	26	20	23
	19	29	20	24
	19	26	19	28
	19	22	19	27
	15	18	14	25
	9	13	10	20
	8	13	4	16
SUMA	145	199	154	207

Tabla 4.4 Tres vectores de numero de cruces por cero de locutor 1 y uno de locutor 2

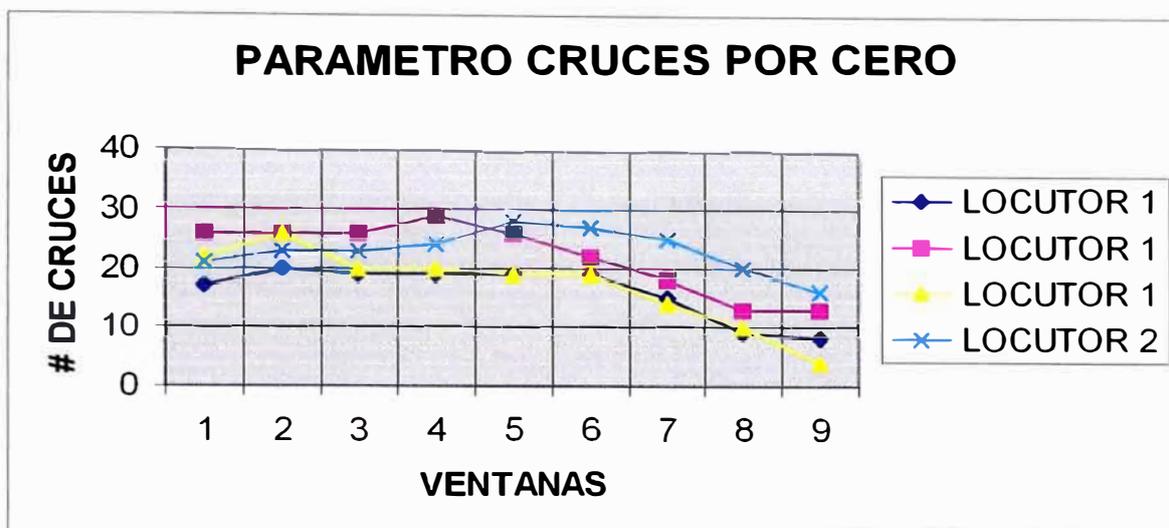


Figura 4.4 Grafico de la tabla 4.4

Comentario

Observamos que la voz del locutor 2 se confunde con locutor 1 pero al final se diferencian, teniendo en la zona central un regular comportamiento.

La suma 1 y la suma 3 de locutor 1 coinciden prácticamente, pero también coinciden la suma 2 de locutor 1 con locutor 2 es un parámetro regular.

e.- El parámetro numero de máximos en el dominio del tiempo, PROGRAMA compa06

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	43	47	36	51
	41	42	41	42
	40	45	43	47
	42	44	34	41
	23	25	22	40
	27	24	20	38
	28	27	28	24
	24	20	30	20
	27	23	39	24
SUMA	295	297	293	327

Tabla 4.5 Tres vectores de numero de máximos de locutor 1 y uno de locutor 2

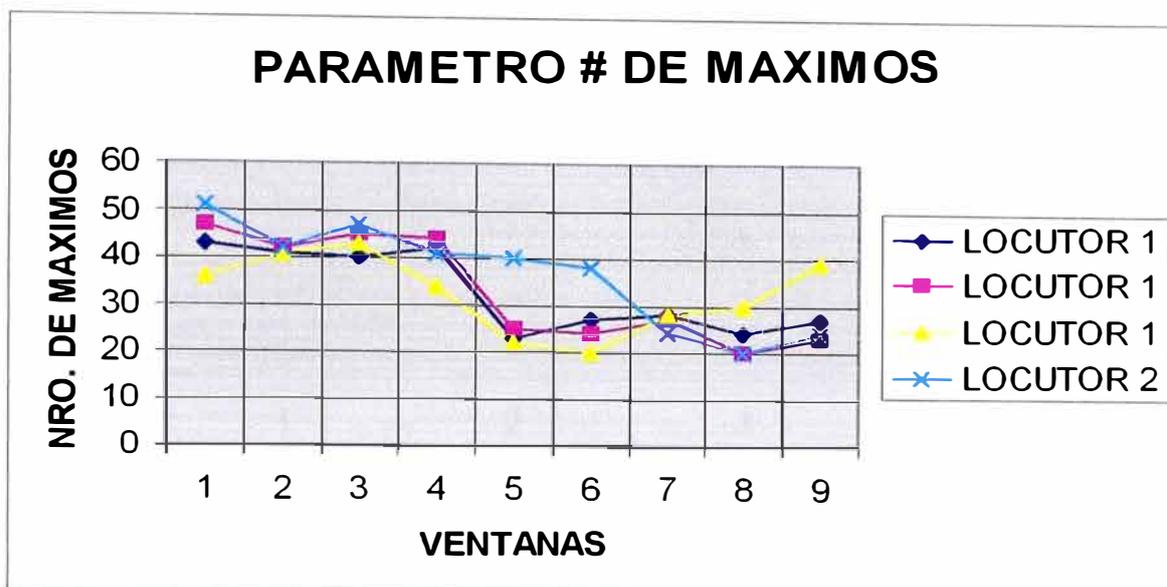


Figura 4.5 Grafico de la tabla 4.5

Comentario

Observamos que la voz del locutor 1 en sus tres archivos coinciden regularmente se diferencian algo de locutor2.

La suma 1, la suma 2 y la suma 3 de locutor 1 coinciden prácticamente, pero están muy cerca de locutor 2 **es un parámetro bueno.**

f.- El parámetro frecuencias máximas o formantes en el dominio de la frecuencia, PROGRAMA compa07

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR 1	LOCUTOR 1	LOCUTOR 1	LOCUTOR 2
	21.89	19.12	19.99	14.53
	21.58	13.94	16.80	31.57
	24.71	41.90	35.00	9.57
SUMA	68.18	74.96	71.79	55.67

Tabla 4.6 Tres vectores de frecuencias máximas de locutor 1 y uno de locutor 2.

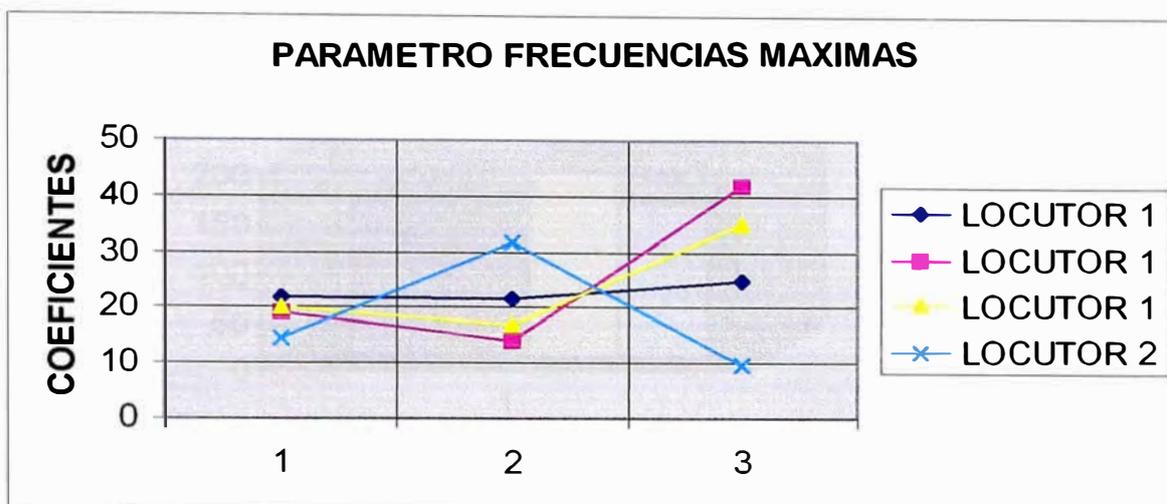


Figura 4.6 Grafico de la tabla 4.6

Comentario

Observamos que la voz del locutor 1 en sus tres archivos coinciden bastante y se diferencian de locutor2.

La suma 1, la suma 2 y la suma 3 de locutor 1 coinciden prácticamente, y se diferencian de locutor 2 **es un parámetro bastante bueno.**

g.- El parámetro promedio de frecuencias en el dominio de la frecuencia,
PROGRAMA compa08

# de archivo	Archivo 1	Archivo 2	Archivo 3	Archivo 1
Voz de:	LOCUTOR1(v1)	LOCUTOR1(v2)	LOCUTOR1(v3)	LOCUTOR 2
	160.04	151.74	189.89	159.90

Tabla 4.7 Promedio de frecuencias de locutor 2 (3) y de locutor 1(1)

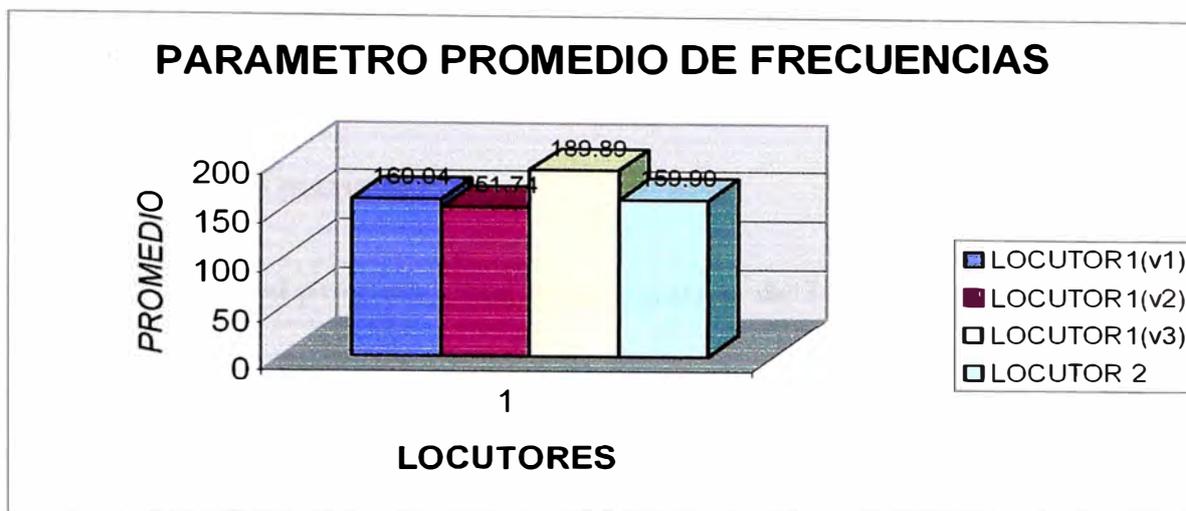


Figura 4.7 Grafico de la tabla 4.7

Comentario

Es un mal parámetro sus valores son muy dispersos. **Mal parámetro.**

h.- Resumen del Estudio Comparativo de los Parámetros

Después del estudio comparativo de los parámetros tenemos el siguiente cuadro resumen.

Parámetro	Calificación	Obs.
Magnitudes	Muy bueno	ok
Medias	Malo	
Energías	Bueno	ok
Cruces por cero	Regular	
# De Máximos	Bueno	ok
Frec. Máximas	Muy bueno	ok
Prom. Frecuencias	Malo	

Debemos escoger tres parámetros para mejorar la posibilidad identificar al locutor, pues con pocos parámetros nos confundiremos de locutor y con muchos parámetros la posibilidad de identificación de un locutor es mínima.

Como el parámetro magnitud y energía son similares desechamos energía y entonces los parámetros considerados serán.

- *Magnitudes*
- *# de máximos*
- *frecuencias máximas*

4.2 Resultados del programa que halla el patrón de 5 locutores y del programa que reconoce finalmente al locutor.

Ejecutadas las pruebas obtenemos los siguientes resultados mostrados en la tabla 4.8, considerando cuatro locutores y su reconocimiento con errores del 10%, 15%, 20% y 25%. Además debemos considerar tres posible resultados.

1. Reconocimiento del locutor.
2. No reconocimiento del locutor.
3. Confusión con otro locutor.

	ERROR DEL 10%			ERROR DEL 15%			ERROR DEL 20%			ERROR DEL 25%		
	Acier.	Error	Conf									
CASTAÑEDA	1	8	0	4	5	0	6	3	0	3	6	0
KENNY	1	8	0	1	8	0	1	8	0	1	5	3
LAUREANO	1	8	0	3	6	0	4	5	0	5	4	0
MARCO	0	9	0	1	8	0	3	6	0	4	5	0
	3	33	0	9	27	0	14	22	0	13	20	3

Tabla 4.8 Resultados del sistema que reconoce locutores, considerando 4 locutores

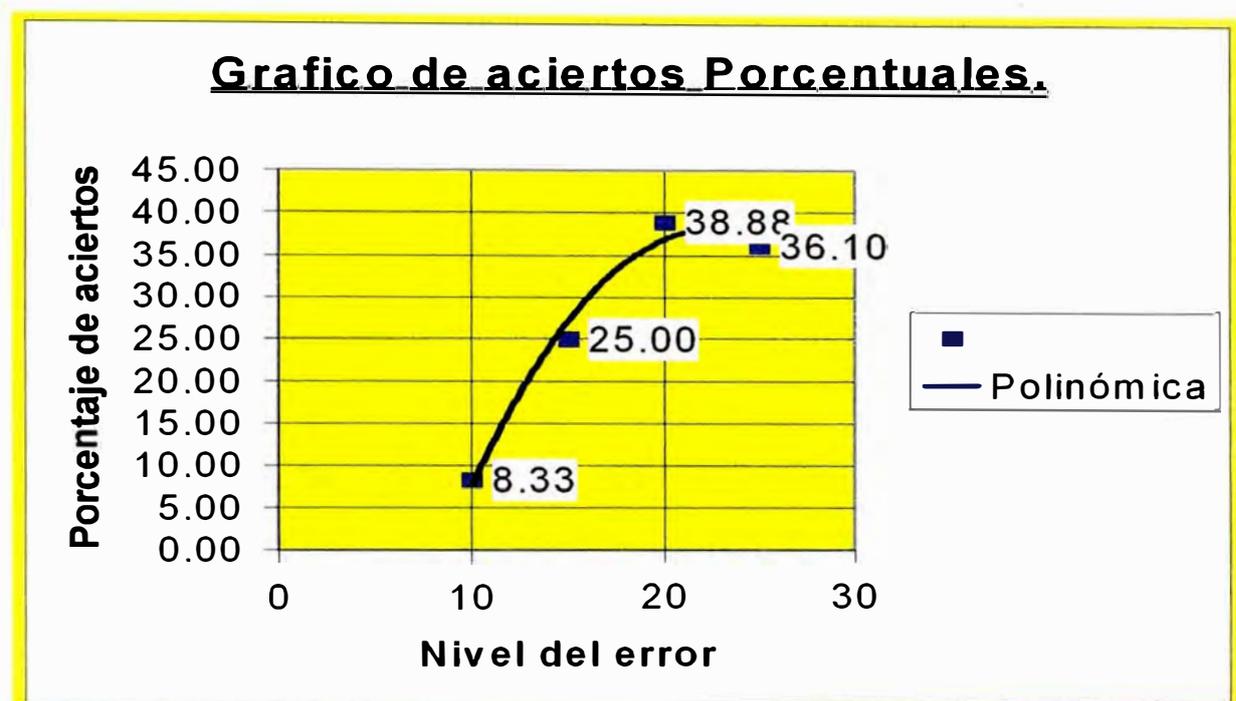


Figura 4.8 Porcentaje de aciertos versus niveles de error.

Vemos claramente que al 20% de error es el mayor porcentaje de aciertos y es del 38.88%

CONCLUSIONES

1. En el tiempo, es fundamental escoger ventanas e incrementos de ventanas del orden adecuado entre las 500 y 1000 muestras.
2. Los parámetros más importantes en el son las magnitudes, los máximos y los cruces por cero.
3. En la frecuencia, es fundamental las frecuencias máximas.
4. Los análisis en el tiempo son más precisos en la primera mitad
5. El sistema de reconocimiento de locutor, no es un método óptimo, pues con un error de 20% su porcentaje de reconocimiento solo llega al 38.88%.
6. Podemos concluir también, que los parámetros en el tiempo aunque varían sus valores en algo, la forma de las curvas es similar para los mismos locutores.
7. Si tomamos varias veces la voz del locutor y hallamos el patrón utilizando los promedios el porcentaje de reconocimiento aumenta.

BIBLIOGRAFÍA

- [1] Jesús Bernal Bermúdez, Jesús Bobadilla Sancho, Pedro Gómez Vilda. “Reconocimiento de voz y fonética acústica”, Universidad Politécnica de Madrid – España, 2000
- [2] HweiP.Su, “Análisis de Fourier”, Universidad de los Andes-Colombia, 1970
- [3] C. Sidney Burrus, James H. McClellan, Alan V. Oppenheim, Thomas W.Parks, Ronald W. Shafer, Hans W.Schuessler, “ Ejercicios de tratamiento de la señal utilizando MATLAB”, Prentice Hall, Madrid 1998.
- [4] Freddy Sotelo Valer, Héctor Laura Castañeda, Robert Castro Salguero “Reconocimiento de locutor y síntesis de voz”, Universidad Nacional de Ingeniería- Lima, 2000.
- [5] Alan V. Oppenheim, Ronald W. Schafer, Jhon R. Buck. “Tratamiento de señales en tiempo discreto”, Prentice Hall, Madrid 2000.
- [6] Dr. Jorge Del Carpio Salinas “Tratamiento Digital de Señales”, Universidad Nacional de Ingeniería- Lima, 2003.