

ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERÍA ELECTRICA

TESIS DE GRADO

ESTUDIO Y SIMULACION DE METODOS DE  
RECUPERACION DE SINCRONISMO DE SEÑALES PSK

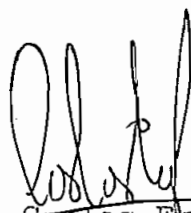
Tesis previa a la obtención del título de  
INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES

DILO BOLIVAR QUISPE QUILAPANTA

1994

CERTIFICACION

Certifico que el presente trabajo ha sido realizado en su totalidad por el Sr. Dilo Bolívar Quispe Quilapanta

  
Ing ← Carlos Egas

DIRECTOR

## AGRADECIMIENTO

A los profesores de la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional, en especial a los que integran el Departamento de Electrónica y Telecomunicaciones, quienes han aportado con sus valiosos conocimientos en mi realización como profesional.

Al Ing. Carlos Egas, por sus acertados consejos en la elaboración de esta tesis, y a todas las personas que de una u otra manera colaboraron con el desarrollo de la misma.

## INTRODUCCION

## INTRODUCCION

Las comunicaciones que utilizan PSK, a pesar de su alto costo por exigir detección sincrónica, son ampliamente utilizadas debido a su grado de inmunidad al ruido, reducida potencia y altas velocidades de transmisión.

La detección coherente es uno de los métodos de demodulación de señales PSK. Esta técnica necesita la presencia en el circuito demodulador, de un bloque que permita recuperación de portadora y por tanto del sincronismo de la señal recibida.

El presente trabajo fue planteado con los siguientes objetivos:

- Hacer un estudio de diferentes métodos de recuperación de sincronismo para señales PSK.
- Implementar un programa que permita, simular el comportamiento de los circuitos estudiados.
- Hacer un estudio de Filtros y su discretización, ya que son los componentes de los sistemas de recuperación de sincronismo, que mayor influencia tienen sobre los mismos.

Para la simulación se necesita establecer un modelo matemático de los componentes que intervienen en los diferentes circuitos que se van a estudiar de tal manera que los resultados sean comparables con los resultados prácticos.

En el capítulo I, se exponen algunos conceptos básicos de la modulación y demodulación PSK. Además se analiza el Lazo Igualador de Fase como bloque, así como cada componente individual de éste.

El capítulo II está dedicado al análisis y estudio de

los métodos de recuperación de sincronismo de señales PSK.

El capítulo III se dedica al estudio de los diferentes métodos de discretizar sistemas analógicos como los métodos de Integración Numérica, Transformación Bilineal y la Transformación Invariante al Impulso. Se analiza también la discretización de cada componente utilizada para la simulación.

El capítulo IV contiene los algoritmos que se desarrolla para simular el comportamiento del lazo igualador de fase y los métodos estudiados en el capítulo II. Finalmente los resultados obtenidos se muestran en el capítulo V.

Para apreciar el fenómeno de recuperación de sincronismo de señales PSK, son favorables para este propósito los sistemas con un número de fases reducido; razón por la que se pone énfasis en el estudio de estos sistemas.

Por la estructuración de los programas se usa un lenguaje de alto nivel como el C.

# CAPITULO I

## CONCEPTOS FUNDAMENTALES

TERMA	PAGINA
1.1. Modulación y Demodulación PSK	
1.1.1. Modulación	3
1.1.2. Demodulación	3
1.1.3. Tipos básicos de modulación digital	3
1.1.4. Modulación de fase (PSK)	4
1.1.5. Técnicas de modulación de fase multinivel	4
1.1.6. Detección de señales discretas	7
1.2. Lazo igualador de fase	11
1.3. Oscilador Controlado por Voltaje	12
1.4. Filtros Analógicos	
1.4.1. Introducción	14
1.4.2. Filtros de Butterworth	15
1.4.3. Filtros de Tchebyscheff	17
1.4.4. Filtros Elípticos	19
1.4.5. Comparación entre las aproximaciones	23
1.5. Comparador de fase	24

## CAPITULO I

### CONCEPTOS FUNDAMENTALES

#### 1.1 MODULACION Y DEMODULACION PSK

##### 1.1.1 MODULACION

Modulación es la alteración sistemática de una onda portadora de acuerdo con el mensaje .

La modulación es necesaria para acoplar la información con el medio de transmisión. Para esto se modifica los parámetros de una onda portadora, de tal manera que sus características se adapten mejor al medio de transmisión que la onda original en banda base.

##### 1.1.2. DEMODULACION

Se dice Demodulación al proceso inverso a la Modulación. Al modular una portadora sinusoidal con una secuencia binaria de banda base, existe un desplazamiento de la señal modulada resultante a una frecuencia adecuada de transmisión. En el receptor debe realizarse el proceso inverso o demodulación de la señal para recuperar la secuencia binaria original.

##### 1.1.3. TIPOS BASICOS DE MODULACION DIGITAL

Existen tres técnicas básicas de modulación para señales digitales, usando una portadora sinusoidal.

Los estilos de modulación digital numérica se derivan de las tradiciones AM(Amplitude Modulation), FM(Frecuency



Modulation), PM(Phase Modulation) y estas son ASK(Amplitude Shift Keying), FSK(Frecuency Shift Keying) y PSK(Phase Shift Keying) respectivamente.

La modulación digital ASK resulta de las variaciones en amplitud de la portadora de acuerdo a los niveles de la señal en banda base, provocada por un circuito de producto. El caso FSK se traduce como la emisión de una frecuencia para cada dígito binario. En PSK el efecto es el cambio de la fase de portadora.

#### 1.1.4. MODULACION DE FASE (PSK)

Se ha mencionado ya que se dice modulación PSK cuando la portadora cambia su fase de acuerdo al estado de la señal de banda base.

$$S(t) = \cos\left(\omega_p t + \frac{b_n(t) d\phi}{2}\right) \quad (1.1)$$

Donde:  $n$  Es el número de fases.

$d\phi = 2\pi/n$  Es la separación entre fases consecutivas.

$b_n(t)$  Es la señal simétrica en banda base y puede tomar los valores  $\pm 1, \pm 3, \dots$

#### 1.1.5. TECNICAS DE MODULACION DE FASE MULTINIVEL.

El ancho de banda requerido para transmitir una secuencia digital de banda base podría reducirse por medio de modulación multinivel.

Con la conformación ideal de Nyquist puede transmitirse dos símbolos/s/Hz por el canal de ancho de banda de Nyquist de  $B$  hertz. Si se usa un conjunto  $M=2^n$  símbolos, donde  $n$  es el número de sucesivos pulsos binarios que se han combinado para formar el símbolo adecuado que se va a transmitir, pueden transmitirse usando la banda de Nyquist  $2n$  bits/s/Hz.

Consideremos un sistema en el cual se combinan dos pulsos binarios sucesivos y el resultado es el conjunto de cuatro pares binarios 00, 01, 10 y 11, que se usa para modular una onda senoidal de alta frecuencia con cuatro posibles fases, una para cada uno de los pares binarios. La señal  $i$ -ésima, de las cuatro posibles puede escribirse.

$$S_i(t) = \cos(\omega_p t + \theta_i) \quad i = 1, 2, 3, 4 \quad (1.2)$$

Posibles elecciones para las cuatro fases angulares son:

$$\begin{aligned} \theta_i &= 0, \pm \frac{\pi}{2}, \pi \\ \theta_i &= \pm \frac{\pi}{4}, \pm \frac{3\pi}{4} \end{aligned} \quad (1.3)$$

En ambos casos las fases están espaciadas  $\pi/2$  radianes. Las señales de este tipo se denominan señales PSK cuaternarias (QPSK), y son un caso especial de las señales multi-PSK (MPSK).

En general, como ya se ha mencionado  $n$  pulsos binarios sucesivos se acumulan y uno de los  $M = 2^n$  símbolos se pone a la salida. Si la velocidad binaria es  $R$  bits/s, cada intervalo de pulso binario es de  $1/R$  segundos de duración; el símbolo correspondiente de salida es entonces de  $T = n/R$

segundos de duración.

Las señales de (1.2) pueden representarse, como una combinación de dos señales en cuadratura, en la siguiente forma:

$$S_i(t) = a_i \cos(\omega_p t) + b_i \text{sen}(\omega_p t) \quad (1.4)$$

Donde  $a_i$  y  $b_i$  son los coeficientes dados por la secuencia de bits de la información.

Se puede representar las señales en un diagrama bidimensional llamado diagrama de constelaciones. El eje horizontal correspondiente a la ubicación de  $a_i$  se denomina eje de fase, el eje vertical donde se localiza  $b_i$  se dice eje en cuadratura. Las cuatro señales QPSK aparecen entonces como se muestra en la figura 1.1.

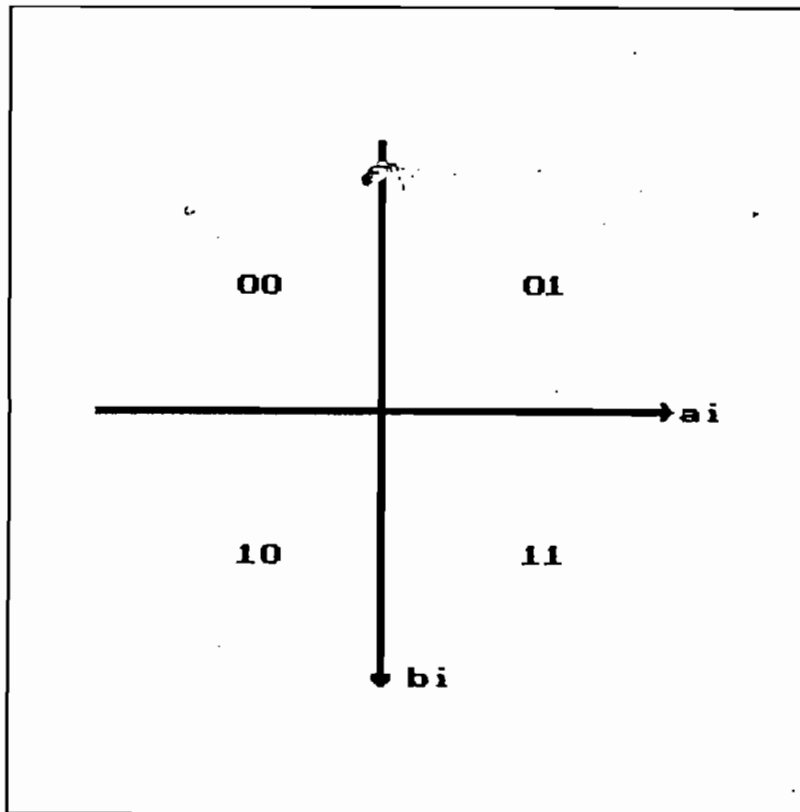


Figura 1.1 : Diagrama de Constelaciones para QPSK.

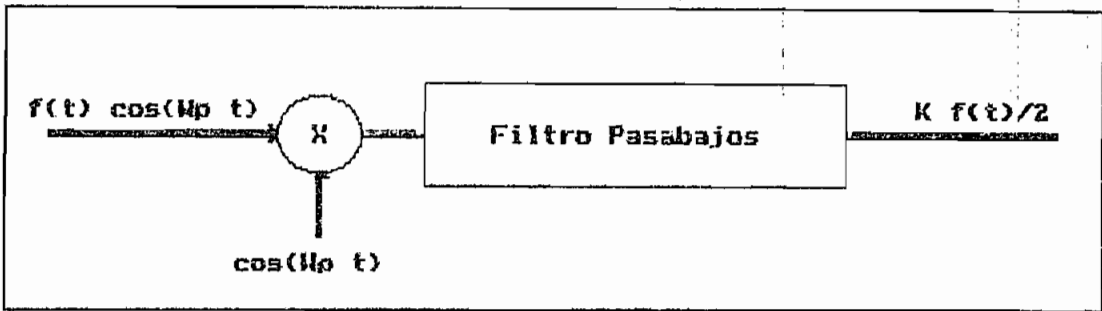
Las representaciones en fase (coseno) y en cuadratura (seno) de la señales  $S_1(t)$  de QPSK sugieren una posible forma de general dichas señales, se toman dos pulsos binarios de entrada sucesivos, los que se almacenan, y el par de números  $a_1$  y  $b_1$ , tomados cada  $T = 2/R$  se usan para modular dos términos de portadora en cuadratura.

#### 1.1.6. DETECCION DE SEÑALES DISCRETAS.

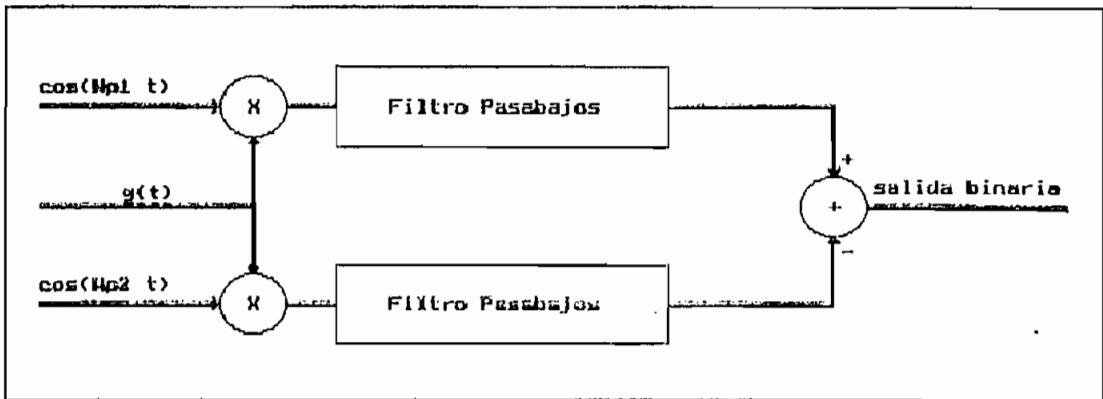
El sistema PSK se comporta mejor en un ambiente ruidoso que los sistema FSK y OOK(On-Off Keying). Se podría pensar entonces que PSK es la técnica óptima para comunicación, sin embargo presenta algunos problemas en los procesos de demodulación que se llevan a cabo.

Existen dos técnicas básicas para la demodulación, detección sincrónica o coherente y detección de envolvente. La detección coherente consiste en la multiplicación de la señal que llega por otra de la misma frecuencia que la portadora la que se genera localmente en el receptor, a continuación la señal multiplicada resultante se hace pasar por un filtro pasabajos.

El proceso de detección sincrónica de señales se muestra en la figura 1.2.



a: Detección de señales OOK o PSK.



b: Detección de señales FSK

Figura 1.2 Detección Sincrónica.

Las señales FSK requieren de dos ondas senoidales, una para cada frecuencia transmitida. Este procedimiento es justamente el inverso al procedimiento original de modulación que se efectúa en el transmisor y que sirve para trasladar las señales binarias de regreso sincrónico hasta las frecuencias de banda base.

Si la señal binaria de alta frecuencia tiene la forma de  $f(t)\cos(W_p t)$ . [Si  $f(t)=\pm 1$ , se tiene la señal 2-PSK; si  $f(t)$  es 1 o 0, se tiene el caso OOK]. Si se multiplica esta señal por  $E_{osc}=k\cos(W_p t)$ , donde  $k$  es una constante arbitraria propia del multiplicador, se obtiene  $kf(t)\cos^2(W_p t) = (k/2)[1+\cos(2W_p t)]f(t)$ . Pero el termino

$f(t)\cos(2W_p t)$  representa la función  $f(t)$  trasladada hasta la frecuencia  $2f_p$ , la que es rechazada por el filtro pasabajos y la salida es finalmente  $(k/2)f(t)$ , justamente la secuencia binaria de banda base que se desea.

Pero se ha supuesto que la portadora local  $E_{osc}=\cos(W_p t)$  esta sincronizada con la señal que llega al receptor, esto es que tienen exactamente la misma frecuencia y fase. Si la señal sinusoidal generada localmente es de una frecuencia  $\cos[(W_p+\Delta W)t]$ , la multiplicación que se genera es  $kf(t)\cos[(W_p+\Delta W)t]\cos(W_p t)=k/2\{\cos[(2W_p+\Delta W)t]+\cos(\Delta W t)\}f(t)$ . La salida del filtro pasabajos tiene entonces la forma  $[kf(t)/2]\cos(\Delta W t)$  y si  $\Delta W$  esta en la banda de paso del filtro, lo detectado no es el mensaje original, sino que sigue modulado pero a otra frecuencia ( $\Delta W$ ).

Si la señal local estuviera a la frecuencia correcta  $W_p$ , pero  $\phi$  radianes defasada respecto de la señal de entrada, es decir  $\cos(W_p t+\phi)$ , la salida del filtro pasabajos sería  $[kf(t)/2]\cos\phi$ . Esta es la salida de banda base deseada, pero se encuentra atenuada en amplitud. En particular a medida que  $\phi$  aumenta,  $\cos\phi$  disminuye. Para  $\phi$  próxima a  $\pi/2$ , la salida sería muy cercana a cero. A medida que  $\phi$  aumenta mas allá de  $\pi/2$ , la señal de salida cambia de signo. Si la señal de banda base es una secuencia polar NRZ, la señal entera cambia de polaridad, todos los unos se transforman en ceros, mientras que los ceros se transforman en unos. Por ello la portadora generada localmente no solo debe ser de la misma frecuencia, sino que también igual en fase. Esta es la razón para denominar este método detección

sincrónica.

El sincronismo de fase es muy difícil de obtener, particularmente si la transmisión se realiza a grandes distancias. Esto significa que un reloj del receptor que proporcione el sincronismo deberá encadenarse o amarrarse al reloj del transmisor dentro de una fracción de un ciclo de la portadora, sin importar lo que ello cueste.

A pesar de la dificultad de mantener la sincronía de fase, los sistemas usando PSK y la detección sincrónica con lazo de amarre de fase han sido usadas con éxito en las comunicaciones.

La otra forma común de detección, llamada detección de envolvente, evita los problemas de tiempo y de fase de la detección sincrónica. En este caso la señal de alta frecuencia que llega se hace pasar a través de un dispositivo no lineal y un filtro pasabajos si se ha modulado en amplitud, en caso de haberse modulado en frecuencia se pasa la señal previamente por un sistema que la transforma en una señal modulada en amplitud y frecuencia.

La señal PSK tiene una envolvente constante y la información está en la fase, de tal manera que no puede emplearse un detector de envolvente. Entonces, el sistema de PSK requiere forzosamente la detección sincrónica.

Puede evitarse el uso de la detección sincrónica adoptando la detección de envolvente, con sistemas de transmisión FSK u OOK, en lugar del óptimo sistema PSK (en presencia de ruido).

## 1.2.-LAZO IGUALADOR DE FASE

Un lazo igualador de fase (PLL), tiene tres componentes básicos:

- 1.- Detector de Fase (PD)
- 2.- Un filtro pasabajos(FPB)
- 3.- Un oscilador controlado por voltaje (VCO).

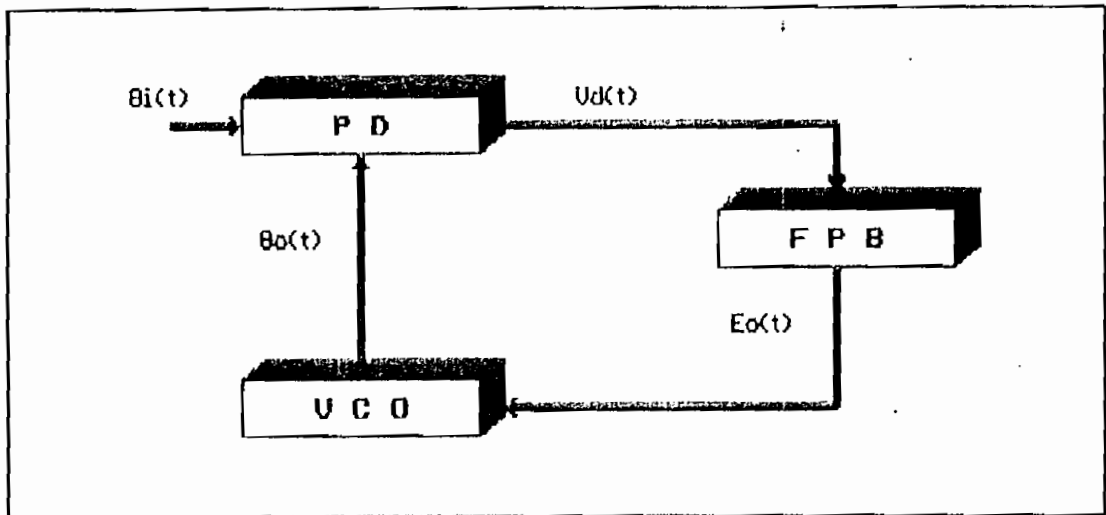


Figura 1.3 : Diagrama de Bloques del PLL

El detector de fase compara la fase de la señal periódica de entrada al lazo, con la fase del VCO, así la salida del detector de fase es una medida de la diferencia de fases entre las dos entradas. Este voltaje es filtrado en el filtro del lazo y aplicado como señal de control al VCO.

La frecuencia de oscilación del VCO es tal que tiende a reducir la diferencia de fases entre el mismo y la señal de entrada al lazo. Cuando el lazo está igualado el voltaje de control es tal que la frecuencia del VCO es igual a la



frecuencia promedio de la señal de entrada.

Si la señal de entrada al lazo tiene una fase  $\theta_1(t)$  y la salida del VCO tiene una fase  $\theta_o(t)$ . La salida del comparador de fase es proporcional a la diferencia de fase que existe entre sus entradas, así  $V_d(t) = K_d(\theta_1(t) - \theta_o(t))$ ; Donde  $K_d$  es el factor de ganancia del PD y esta dado en [voltios/radianes], tomando la transformada de Laplace  $V_d(s) = K_d(\theta_1(s) - \theta_o(s))$

Si la función de transferencia del filtro es  $F(s)$ , la señal filtrada es  $E_o(s) = F(s)V_d(s) = K_d F(s)[\theta_1(s) - \theta_o(s)]$ .  $E_o(t)$  controla la frecuencia del VCO, una desviación del VCO de su frecuencia central está determinada como  $\Delta\omega = K_o E_o(t)$ , donde  $K_o$  es la constante de ganancia del VCO y está dada en [radianes/seg/voltio]. Ya que la frecuencia es la derivada de la fase, la operación del VCO puede ser descrita como  $d\theta_o(t)/dt = K_o E_o(t)$ , tomando la transformada de Laplace  $\theta_o(s) = K_o E_o(s)/s$ .

Reemplazando en esta última expresión  $E_o(s)$ , se obtiene la ecuación fundamental del PLL.

$$\frac{\theta_o(s)}{\theta_1(s)} = \frac{K_d K_o F(s)}{s + K_o K_d F(s)} \quad (1.5)$$

### 1.3.- OSCILADOR CONTROLADO POR VOLTAJE (VCO)

El VCO es un dispositivo que entrega a su salida una señal periódica, cuya frecuencia es función lineal del voltaje de control,  $f_o = f_p + k_v v_c$ . Donde  $f_p$  es la frecuencia nominal del oscilador, es decir la frecuencia de la señal de salida cuando el control de voltaje ( $v_c$ ) es cero. De esta

forma un oscilador convencional puede ser considerado como un VCO con  $v_c=0$ .

Una característica importante del VCO es la sensibilidad en fase o frecuencia, que puede ser considerada en dos formas. Sensibilidad en forma general refiriéndose al cambio de frecuencia sobre períodos de minutos, horas, días, años, etc. Esta sensibilidad de frecuencia es normalmente limitada por los coeficientes de temperatura de los circuitos componentes. En otros términos se habla de sensibilidad en períodos muy pequeños de tiempo, donde sensibilidad de fase es usada sinónimamente con sensibilidad de frecuencia; y se refiere a como reacciona el VCO con su frecuencia de oscilación a pequeños desplazamientos de fase del sistema de lazo abierto. La figura 1.4 muestra las respuestas de dos sistemas a un cambio de fase. Si a la frecuencia de oscilación ( $f_0$ ) el defasamiento es de  $180^\circ$ , si por alguna circunstancia se da un defasamiento de  $10^\circ$  entre las señales, el sistema  $GH2(w)$  tiene una sensibilidad mas alta que  $GH1(w)$ , pues  $Af2 > Af1$ , es decir el cambio de frecuencia por grado de error de fase es mayor en  $GH2$  que en  $GH1$ .

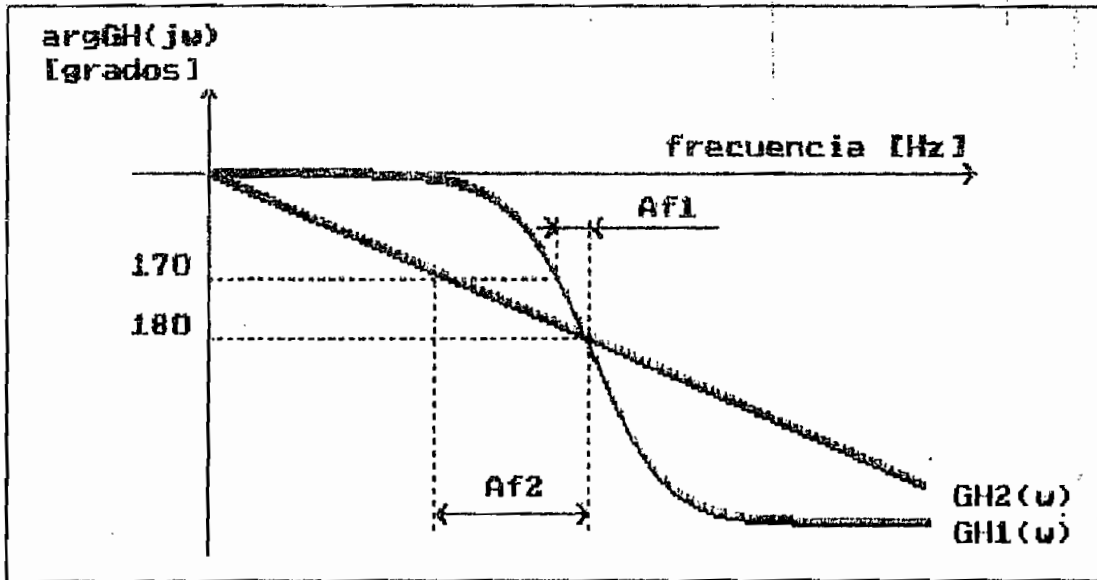


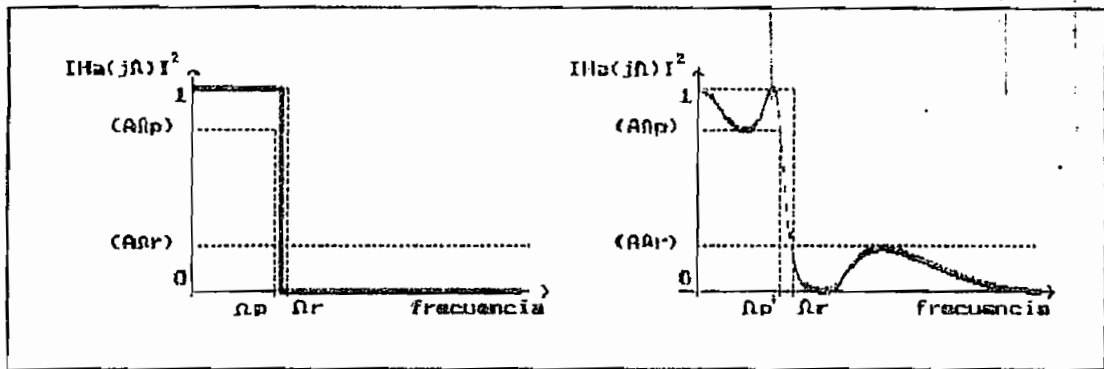
Figura 1.4 : Respuesta del VCO en lazo abierto.

## 1.4.- FILTROS ANALOGICOS DE BUTTERWORTH, TCHEBYSCHIEFF Y ELIPTICOS

### 1.4.1.- INTRODUCCION

Un filtro en general tiene por objeto separar ciertas señales útiles de otras indeseables mezcladas entre sí, por consiguiente para estructurar un filtro se necesita conocer el espectro de frecuencia que tiene la señal útil. Un filtro ideal permite que se transmita todas las componente útiles sin atenuación, sin defasamiento y eliminando todas las señales indeseables. Pero en la práctica un filtro presenta las siguientes imperfecciones:

- Una banda de paso con una atenuación que no es nula.
- Una banda de corte cuya atenuación no es infinita.
- Una banda de transición en donde el paso a la banda de corte no es brusca, sino de forma progresiva.



a. filtro ideal

b. filtro real

Figura 1.5 : Respuesta de frecuencia de un Filtro Pasabajos

Banda de Paso :  $0 < \Omega < \Omega_p$

Banda de Transición :  $\Omega_p < \Omega < \Omega_r$

Banda de Atenuación (Corte o Supresión) :  $\Omega_r < \Omega < \infty$

Son numerosos los procedimientos matemáticos existentes para conseguir la función de transferencia que representa un filtro. Entre estos se analizan los filtros de Butterworth, Tchebyscheff y Elípticos.

#### 1.4.2.-FILTROS DE BUTTERWORTH.

Se caracterizan por tener una curva de respuesta plana en la banda de paso. Para satisfacer esta exigencia, la función característica de Butterworth de grado  $N$ , sus  $2N-1$  primeras derivadas deberán ser nulas para  $\Omega=0$ , así su función de transferencia toma la forma:

$$[H_a(j\Omega)]^2 = \frac{1}{1 + \left(\frac{j\Omega}{j\Omega_c}\right)^{2N}} \quad (1.6)$$

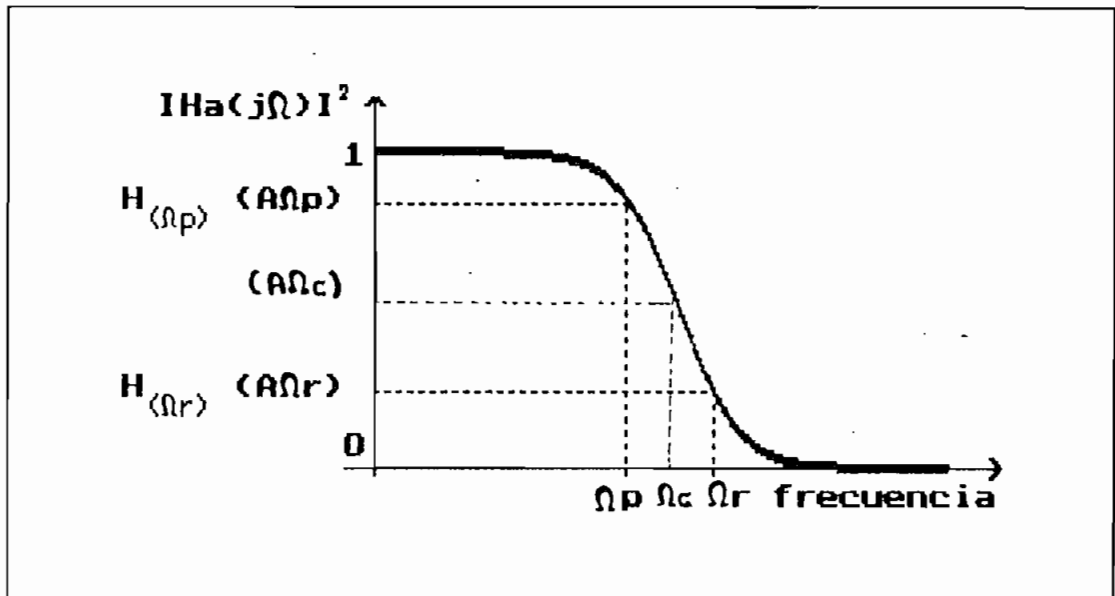


Figura 1.6: Respuesta de frecuencia del filtro Pasabajos de Butterworth.

Para obtener los polos de  $|H_a(j\Omega)|^2$  se iguala el denominador a cero, así los polos están dados por:

$$S_p = \Omega_c e^{\frac{2p+N-1}{2N} \pi j} \quad \text{para } p = 1, 2, \dots, 2N \quad (1.7)$$

Donde  $\Omega_c$  es la frecuencia en la que  $|H_a(j\Omega)|^2$  toma el valor  $-3\text{dB}$ .

$$\Omega_c = \Omega_p \left( 10^{\frac{A\Omega_p}{10}} - 1 \right)^{-\frac{1}{2N}} \quad (1.8)$$

En el plano  $S$ , los polos se ubican simétricamente respecto al eje imaginario y equiespaciados. Para la estructuración del filtro por estabilidad se escoge aquellos que se encuentran en el semiplano izquierdo.

Dado  $\Omega_p$ ,  $\Omega_r$ , Atenuación en  $\Omega_p$ , y Atenuación en  $\Omega_r$ , el orden del filtro esta dado por:

$$N = \frac{1}{2} \frac{\log \frac{10^{0.1 A \Omega_r} - 1}{10^{0.1 A \Omega_p} - 1}}{\log \frac{\Omega_r}{\Omega_p}} \quad (1.9)$$

Una vez calculados los polos con la expresión (1.7), la función de transferencia del filtro queda:

$$H_N(s) = \frac{H_0}{\prod_{i=1}^N (s - p_i)} \quad (1.10)$$

#### 1.4.3. FILTROS DE TCHEBYSCHEFF

El filtro de Tchebyscheff Directo, se caracteriza porque su atenuación en la banda de paso oscila, mientras que la atenuación aumenta de forma continua en la banda de supresión.

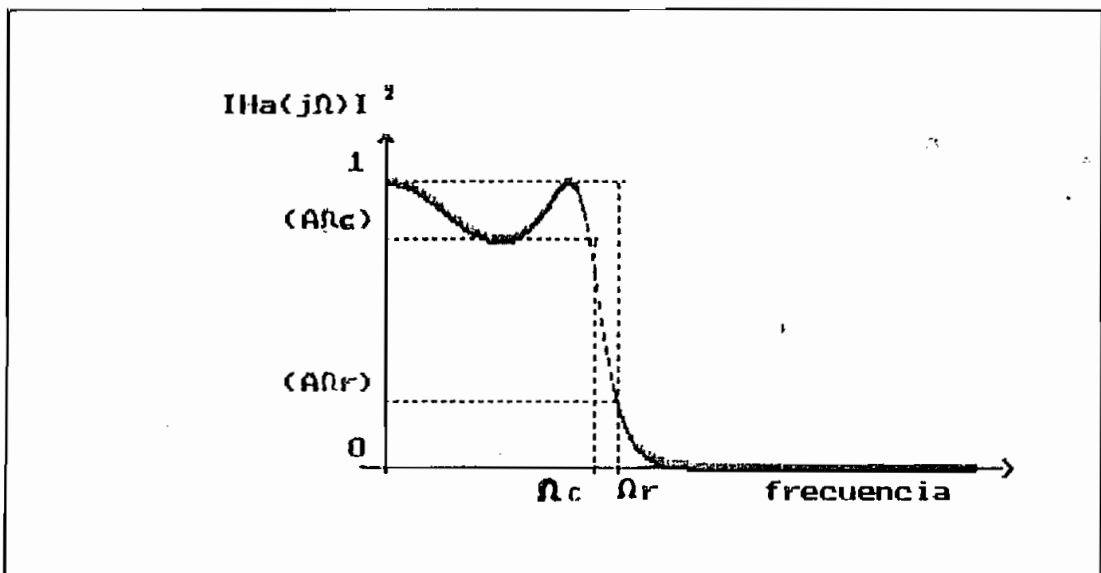


Figura 1.7 : Respuesta de frecuencia del filtro pasabajos Directo de Tchebyscheff.

Su función característica es:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \epsilon^2 V_N^2(\Omega / \Omega_c)} \quad (1.11)$$

Con  $\epsilon \leq 1$

Donde  $V_N(x)$  son los polinomios de Tchebyscheff, cuyo valor oscila  $N$  veces entre  $+1$  y  $-1$  cuando la variable  $x$  oscila entre  $+1$  y  $-1$ .

$$V_N(x) = \cos[N \cos^{-1}(x)] \quad (1.12)$$

$$V_{N+1}(x) = 2x V_N(x) - V_{N-1}(x) \quad (1.13)$$

Como consecuencia del comportamiento de  $V_N^2(x)$ ,  $|H_a(j\Omega)|^2$  para  $0 < \Omega < \Omega_c$  oscila entre  $1$  y  $1/(1+\epsilon^2)$  y para  $\Omega > \Omega_c$ ,  $V_N^2(x)$  crece indefinidamente, en consecuencia,  $|H_a(j\Omega)|^2$  tiende monótonamente a cero.

Se debe tener en cuenta que  $\Omega_c$  es la máxima frecuencia a la cual  $|H_a(j\Omega)|^2$  es igual a  $1/(1+\epsilon^2)$  mas no necesariamente la frecuencia de  $-3\text{dB}$ .

Un filtro de Tchebyscheff queda totalmente definida al especificar  $\Omega_c$ ,  $\epsilon$  y  $N$ . Igualando el denominador de (1.11) a cero se determina los polos del filtro.

$$S_p = \Omega_c \left[ \frac{e^\alpha - e^{-\alpha}}{2} \cos\left(\frac{2p+N-1}{2N}\right)\pi + j \frac{e^\alpha - e^{-\alpha}}{2} \text{sen}\left(\frac{2p+N-1}{2N}\right)\pi \right] \quad (1.14)$$

Donde

$$\alpha = \frac{1}{N} \ln \left( \frac{1}{\epsilon} + \sqrt{\frac{1}{\epsilon^2} + 1} \right) \quad (1.15)$$

$$S_p = \sigma_p + j \omega_p = \text{sen } u_p \text{ senh } v_p + j \cos u_p \cosh v_p \quad (1.16)$$

(1.16) demuestra que las raíces de la función de

transferencia se encuentran en una elipse cuya ecuación es:

$$\frac{\sigma_p^2}{\sinh v} = \frac{W_D^2}{\cosh v} = 1 \quad (1.17)$$

El orden del filtro está dado por:

$$N \geq \frac{\cosh^{-1} \left( \frac{\sqrt{10^{0.1A\Omega_r-1}}}{\sqrt{10^{0.1A\Omega_c-1}}} \right)}{\cosh^{-1}(\Omega_r/\Omega_c)} \quad (1.18)$$

Determinados los polos  $S_p$ , la función de transferencia del filtro queda:

$$H_N(s) = \frac{H_o}{D(s)} = \frac{H_o}{\prod_{l=1}^N (s - s_{pl})} \quad (1.19)$$

Con:

$$H_o = \begin{cases} 10^{-0.05 A\Omega_c} \prod_{l=1}^N (-s_{pl}) & \text{para } N \text{ par} \\ \prod_{l=1}^N (-s_{pl}) & \text{para } N \text{ impar} \end{cases} \quad (1.20)$$

#### 1.4.4.- FILTROS ELIPTICOS

Aunque la aproximación de Tchebyscheff directa tiene una respuesta mejor que la aproximación de Butterworth en la banda de transición, se puede mejorar la característica de esta banda usando aproximación elíptica. En ésta la banda de paso oscila entre 0 y atenuación máxima ( $A\Omega_p$ ), igualmente la banda de atenuación oscila entre infinito y atenuación mínima ( $A\Omega_r$ ).



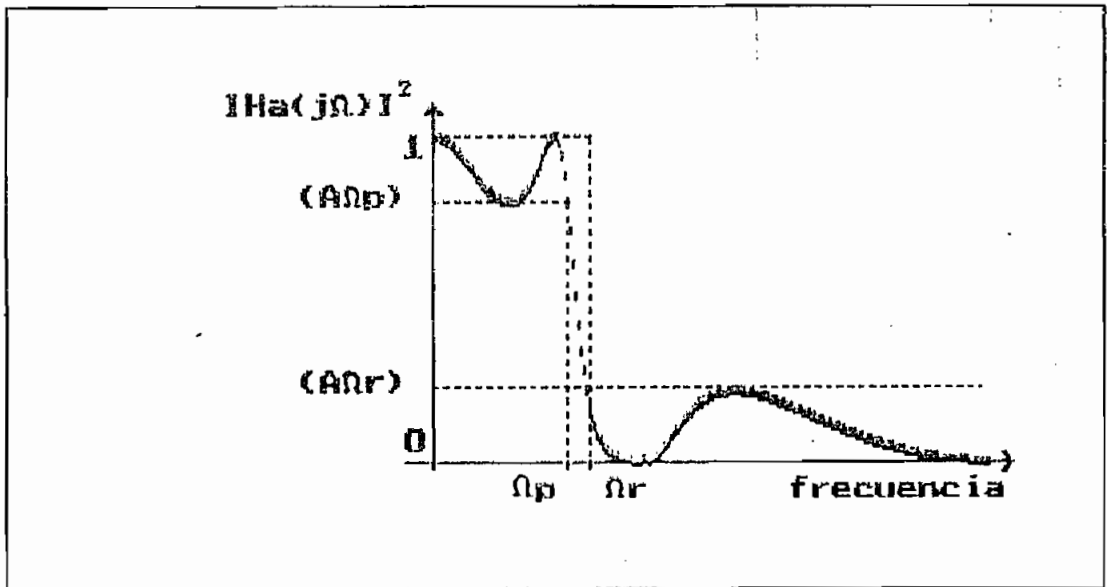


Figura 1.8 : Respuesta de frecuencia de un filtro pasabajos Elíptico.

Los filtros de Bessel, Butterworth, Tchebyscheff y Legendre presentan sus funciones de atenuación en forma de polinomio  $A^2(\Omega) = h(1 + \epsilon N(\Omega^2))$ . Por esta razón para cualquier valor finito de frecuencia, la atenuación tiene valor finito, consecuentemente la función de transferencia  $H(\Omega)$  no tiene ceros de transmisión.

Si la atenuación toma la forma  $A^2(\Omega) = h(1 + N(\Omega^2)/P(\Omega^2))$ , cuando  $P(\Omega^2) = 0$  se tiene  $A(\Omega) = \infty$ . De esta forma se introduce frecuencia de atenuación infinita o ceros de transmisión. La presencia de estos tiene ventajas importantes como:

- Suprime las frecuencias particularmente indeseables, por ejemplo la portadora en un filtro de demodulación.
- Hace que la banda de transición sea mas brusca (mas ideal), situando un cero inmediatamente después de la frecuencia de corte, sin necesidad de aumentar el orden del

filtro.

La función característica de un filtro Elíptico es:

$$H^2(\Omega) = \frac{1}{1 + \epsilon^2 \operatorname{sn}^2(\Omega, k_1)} \quad (1.21)$$

Donde  $y = \operatorname{sn}(\Omega, k_1)$  es el integral elíptico de primer orden.

$$\Omega = \int_0^{\operatorname{Arc} \operatorname{sen}(y)} \frac{d\theta}{(1 - k_1^2 \operatorname{sen}^2\theta)^{1/2}} \quad (1.22)$$

Siendo  $k$  la selectividad ( $k = \Omega_p / \Omega_r$ ), el orden del filtro se obtiene de:

$$n = \frac{K(k) K(\sqrt{1 - k_1^2})}{K(k_1) K(\sqrt{1 - k^2})} \quad (1.24)$$

Donde  $K(k)$  es el integral elíptico COMPLETO de primer orden.

$$K(k) = \int_0^{\pi/2} \frac{d\theta}{(1 - k^2 \operatorname{sen}^2\theta)^{1/2}} \quad (1.25)$$

Un filtro pasabajos elíptico puede ser representado por la siguiente función de transferencia.

$$H_n(s) = \frac{H_0}{D_0(s)} \prod_{i=1}^r \frac{s^2 + A_{0i}}{s^2 + B_{1i} s + B_{0i}} \quad (1.26)$$

$$\text{Donde} \quad r = \begin{cases} \frac{n-1}{2} & \text{para } n \text{ impar} \\ \frac{n}{2} & \text{para } n \text{ par} \end{cases}$$

$$\text{y} \quad D_0(s) = \begin{cases} s + \sigma_0 & \text{para } n \text{ impar} \\ 1 & \text{para } n \text{ par} \end{cases}$$

Conocido  $k$ ,  $f_p$ ,  $f_r$ , Atenuación en  $f_p$  y Atenuación en

$f_r$ , las constantes de (1.26) pueden ser determinadas mediante el siguiente procedimiento matemático.

$$k_1 = \sqrt{1 - k^2} \quad (1.27)$$

$$q_0 = \frac{1}{2} \left( \frac{1 - \sqrt{k_1}}{1 + \sqrt{k_1}} \right) \quad (1.28)$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \quad (1.29)$$

$$D = \frac{10^{0.1 \Lambda \Omega p - 1}}{10^{0.1 \Lambda \Omega r - 1}} \quad (1.30)$$

$$n \geq \frac{\log 16D}{\log (1/q)} \quad (1.31)$$

$$\lambda = \frac{1}{2n} \ln \frac{10^{0.05 \Lambda \Omega r} + 1}{10^{0.05 \Lambda \Omega r} - 1} \quad (1.32)$$

$$\sigma_0 = \left[ \frac{2q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh[(2m+1)\lambda]}{1+2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cosh 2m\lambda} \right] \quad (1.33)$$

$$W = \sqrt{(1 + k \sigma_0^2) \left(1 + \frac{\sigma_0^2}{k}\right)} \quad (1.34)$$

$$\Omega_i = \frac{2q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \operatorname{sen} \frac{(2m+1)\pi\nu}{n}}{1+2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cos \frac{2m\pi\nu}{n}} \quad (1.35)$$

Con

$$v = \begin{cases} i & \text{para } n \text{ impar} \\ i - \frac{1}{2} & \text{para } n \text{ par} \end{cases} \quad i=1,2,\dots,r$$

$$V_i = \sqrt{(1 - k \Omega_i^2) \left(1 - \frac{\Omega_i^2}{k}\right)} \quad (1.36)$$

$$A_{oi} = \frac{1}{\Omega_i^2} \quad (1.37)$$

$$B_{oi} = \frac{(\sigma_0 V_i)^2 + (\Omega_i W)^2}{(1 + \sigma_0^2 \Omega_i^2)^2} \quad (1.38)$$

$$B_{1i} = \frac{2\sigma_0 V_i}{1 + \sigma_0^2 \Omega_i^2} \quad (1.39)$$

$$H_0 = \begin{cases} \sigma_0 \prod_{i=1}^r \frac{B_{oi}}{A_{oi}} & \text{para } n \text{ impar} \\ 10^{-0.05ANp} \prod_{i=1}^r \frac{B_{oi}}{A_{oi}} & \text{para } n \text{ par} \end{cases} \quad (1.40)$$

#### 1.4.5.- COMPARACION ENTRE LAS FUNCIONES DE APROXIMACION ESTUDIADAS.

Los filtros de Butterworth presentan una respuesta transitoria conveniente, debido a la ubicación de los polos pero presenta una banda de transición deficiente, así estos filtros se los puede utilizar en aplicaciones en las que no se necesita una gran precisión y en los que el criterio fundamental sea la simplicidad de estructuración y

construcción.

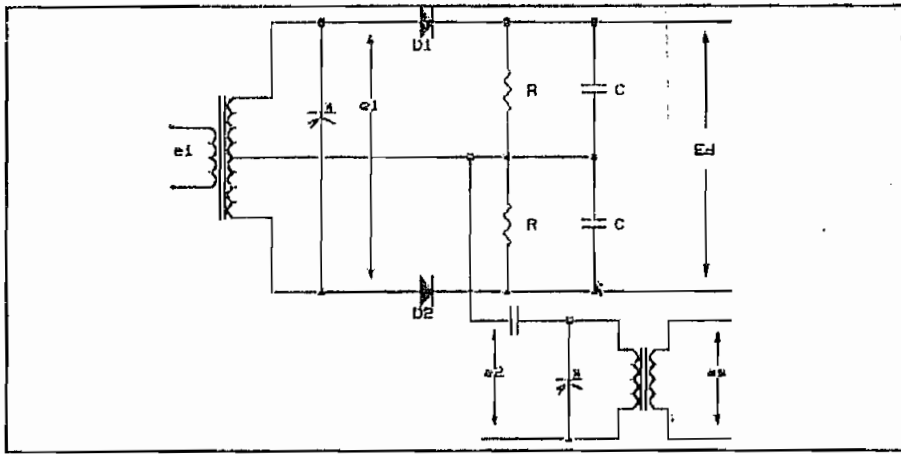
Los filtros de Tchebyscheff, presentan un corte mas brusco que los Butterwoth, se utiliza en aplicaciones donde la pendiente de transición es importante y además no sea posible usar filtros elípticos por ejemplo debido a la dificultad de hallar los ceros de transmisión.

De esta manera se observa que los filtros que mas se aproximan a lo ideal son los elípticos, pero su comportamiento en régimen transitorio presenta deformaciones elevadas de la señal filtrada.

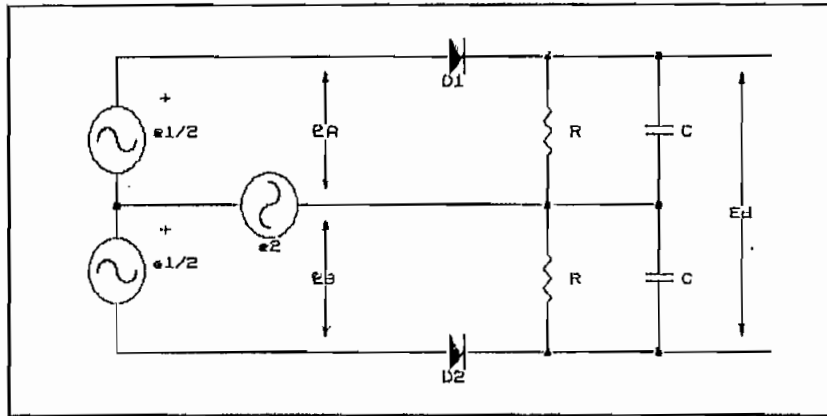
#### 1.5.- COMPARADOR DE FASE

Un detector de fase puede ser considerado un multiplicador analógico ideal. Sin embargo éste, es usado únicamente para bajas frecuencias y muchos PLL trabajan en altas frecuencias. Aunque un multiplicador es un modelo matemático conveniente de un detector de fase, se tiene otras opciones como las siguientes:

Un comparador de fase muy popular se presenta a continuación.



a. Circuito completo



b: Circuito equivalente

Figura 1.9 : Comparador de fase

Sean las entradas al circuito:

$$e_1 = E_1 \text{ sen } (\omega t) \quad (1.41)$$

$$e_2 = E_2 \text{ cos } (\omega t + \theta) \quad (1.42)$$

El voltaje aplicado al diodo y resistencia para cada malla es:

$$e_A = e_2 + \frac{e_1}{2} \quad (1.43)$$

$$e_B = e_2 - \frac{e_1}{2} \quad (1.44)$$

En el diagrama de fasores se puede notar el estado de las señales.

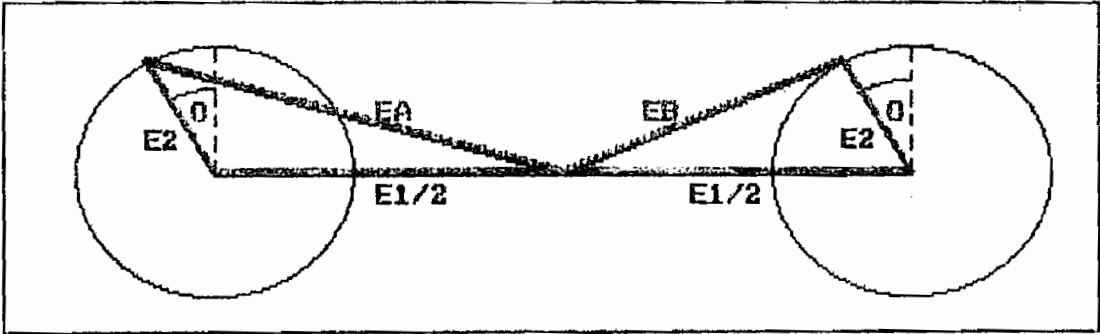


Figura 1.10 : Diagrama de fasores

De acuerdo a la ley de los cosenos se tiene

$$E_A^2 = \frac{E_1^2}{4} + E_2^2 + E_1 E_2 \operatorname{sen}\theta \quad (1.45)$$

$$E_B^2 = \frac{E_1^2}{4} + E_2^2 - E_1 E_2 \operatorname{sen}\theta \quad (1.46)$$

La salida de voltaje del detector de fase  $E_d$  es igual a la diferencia de los dos voltajes rectificadas  $E_d = E_A - E_B$ .

Desarrollando :

$$E_A^2 - E_B^2 = 2 E_1 E_2 \operatorname{sen}\theta \quad (1.47)$$

$$E_d = E_A - E_B = \frac{2 E_1 E_2 \operatorname{sen}\theta}{E_A + E_B} \quad (1.48)$$

Si  $E_2$  es despreciable frente a  $E_1/2$ ,  $E_A + E_B = E_1$ , de tal forma que,  $E_d = 2E_2 \operatorname{sen}\theta$ . Esta última ecuación demuestra que  $E_d$  es proporcional a  $E_2$  pero independiente de  $E_1$ . También es claro que cuando  $\theta = 0^\circ$   $E_d = 0$ , y que  $E_d = 2E_2$  cuando  $\theta = \pm 90^\circ$ .

Otro comparador de fase es:

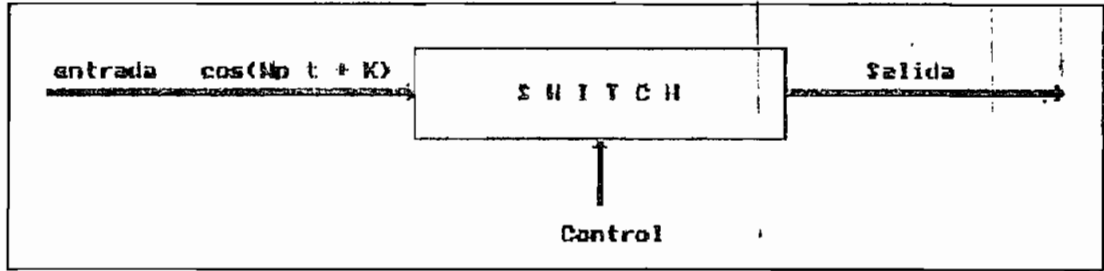
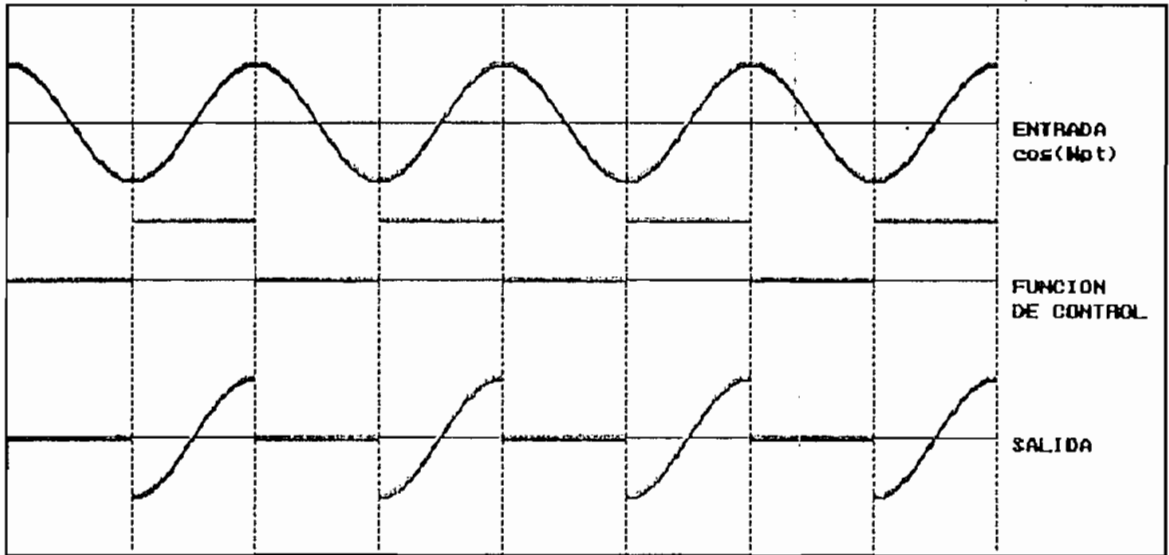
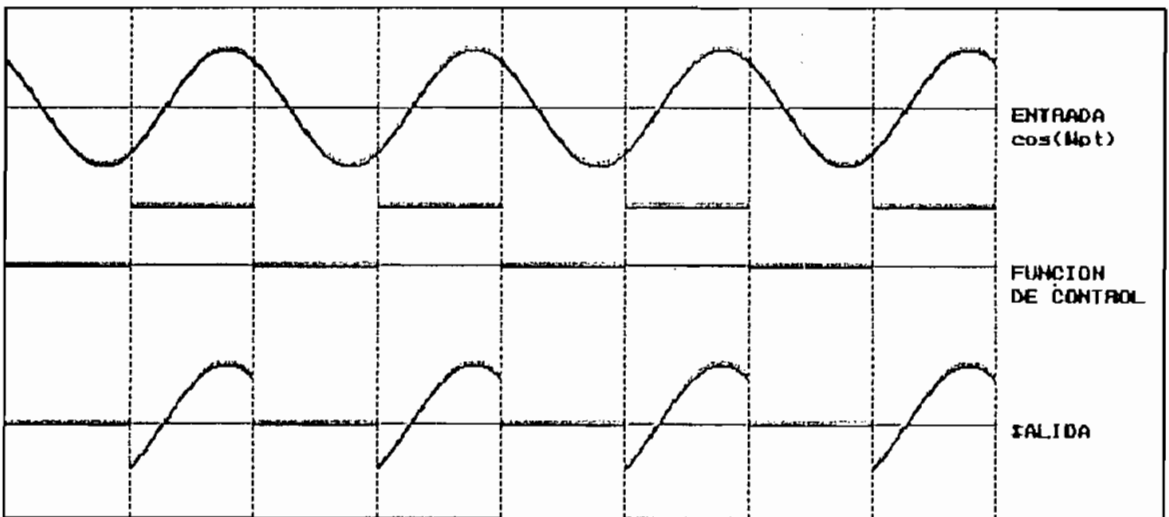


Figura 1.11 : Comparador de fase con Integrador.

Cuya respuesta se observa en las curvas:



a: error de fase cero ( $\psi = 0$ )



b: error de fase diferente de cero ( $\psi \neq 0$ )

Figura 1.12: Respuesta del PD tipo switch



Si la entrada es una función sinusoidal

$$e_s = E_s \cos(W t + \phi) \quad (1.47)$$

La salida del switch es la misma función de entrada para  $0 < Wt < \pi$  y es cero para  $\pi < Wt < 2\pi$ . Pasando por un filtro pasabajos (Integrador), la componente continua de la salida es:

$$E_d = \frac{E_s}{2\pi} \int_0^\pi \cos(W t + \phi) dWt \quad (1.48)$$

$$E_d = - \frac{E_s \operatorname{sen}\phi}{\pi} \quad (1.49)$$

Cuando la diferencia de fase entre las señales de entrada al switch es cero, se tiene  $E_d=0$ .

## CAPITULO I I

### METODOS DE DETECCION DE SINCRONISMO DE SEÑALES PSK

TEMA	PAGINA
2.1. Bucle con multiplicación	29
2.2. Bucle de costas general	31
2.3. Bucle con remodulación por el oscilador local	33
2.4. Bucle con remodulación por la señal recibida	34
2.5. Bucle de costas con detección dentro de un brazo	35
2.6. Bucle de costas numérico	37
2.7. Bucle de costas con remodulación para QPSK	38

## CAPITULO II

## METODOS DE DETECCION DE SINCRONISMO DE SEÑALES PSK

## 2.1.- BUCLE CON MULTIPLICACION

Cuando se usa un sistema de recuperación de portadora con un PLL simple se tiene inconvenientes como:

- Para evitar la influencia de los cambios de fase producidos por la señal modulada, el filtro pasabajos debe ser muy estrecho; es decir que la integración realizada por estos filtros deberá ser mucho mas lenta que el período entre dos cambios sucesivos de fase, mientras mas precisión se requiera en el enganche de fase. La que lleva a un tiempo de enganche elevado.

- Las distintas fases tendrán que ser equiprobables, para evitar una progresiva polarización del VCO, hacia la fase mas probable.

Para mejorar el comportamiento respecto a los inconvenientes presentados al usar un PLL simple como recuperador, se plantea el siguiente bucle:

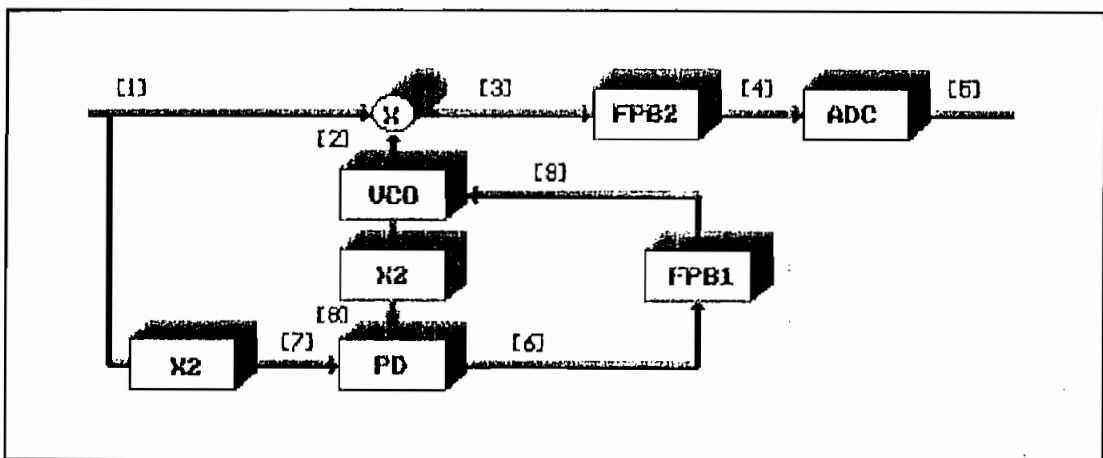


Figura 2.1 : Diagrama de bloques del Bucle con Multiplicación

Con esta estructura el PLL ya no se polariza hacia una determinada fase errónea, aunque en la señal modulada esta fase sea mas probable. Tampoco es necesario que el tiempo de integración sea tan elevado por lo que el tiempo de enganche, puede reducirse notablemente.

Sea la señal modulada en fase que llega al bucle  $e_1(t)$

$$e_1(t) = A \cos(\omega_p t + \phi(t)) \quad (2.1)$$

Sea la señal generada en el bucle por el bloque recuperador de portadora  $e_2(t)$ .

$$e_2(t) = B \cos(\omega_p t + k) \quad (2.2)$$

Las respuestas de los bloques duplicadores de frecuencia son:

$$e_7(t) = A_1 \cos(2\omega_p t + 2\phi(t)) \quad (2.3)$$

$$e_8(t) = B_1 \cos(2\omega_p t + 2k) \quad (2.4)$$

El comparador de fase tiene componentes Armónicos de  $W_p$  mas una componente continua.

$$e_6 = A_1 B_1 K_{PD} [ f(n W_p) + \text{sen}(2\phi(t) + 2k) ] \quad (2.5)$$

Con un buen desempeño del filtro pasabajos, a su salida se tiene:

$$e_9(t) = A_1 B_1 K_{PD} \text{sen}(2\phi(t) + 2k) \quad (2.6)$$

Para este caso  $\phi(t) = \pi$  o  $0$  para los estados lógicos 0 y 1 de banda base, así:

$$e_9(t) = A_1 B_1 K_{PD} \text{sen}(2k) \quad (2.7)$$

Donde  $k$  representa la diferencia de fase entre las señales recibida  $e_1(t)$  y la creada por el VCO  $e_2(t)$ .

El VCO tendrá una respuesta que tiende a disminuir el valor de  $k$ .

Para el detector sincrónico:

$$e_3(t) = A B \cos(\omega_p t + \phi(t)) \cos(\omega_p t + k)$$

$$e_3(t) = \frac{A B}{2} [ \cos(2 \omega_p t + \phi(t) + k) + \cos(\phi(t) - k) ] \quad (2.8)$$

Luego del filtro:

$$e_4(t) = \frac{A B}{2} [ \cos(\phi(t) - k) ] \quad (2.9)$$

Como consecuencia de los valores que toma  $\phi(t)$ ,  $e_4(t)$  es:

$$e_4(t) = \begin{cases} \frac{A B}{2} \cos(k) & \text{para } \phi(t) = 0 \\ -\frac{A B}{2} \cos(k) & \text{para } \phi(t) = \pi \end{cases} \quad (2.10)$$

Como  $k$  tiende a cero cuando el bucle está sincronizado, la señal demodulada en términos ideales es:

$$e_4(t) = \begin{cases} \frac{A B}{2} & \text{para } \phi(t) = 0 \\ -\frac{A B}{2} & \text{para } \phi(t) = \pi \end{cases} \quad (2.11)$$

Finalmente Ecuando.

$$e_5(t) = \begin{cases} +m & \text{para } \phi(t) = 0 \\ -m & \text{para } \phi(t) = \pi \end{cases} \quad (2.12)$$

Donde  $m$  es el valor de voltaje del nivel lógico recuperado.

## 2.2.- BUCLE DE COSTAS GENERAL

En el lazo de Costas la portadora es extraída

multiplicando el valor del voltaje de entrada en dos detectores de fase ( multiplicadores analógicos ), con la producida por el VCO y la misma defasada 90 grados. Filtrando estos resultados y multiplicándolos se obtiene la señal de control del VCO .

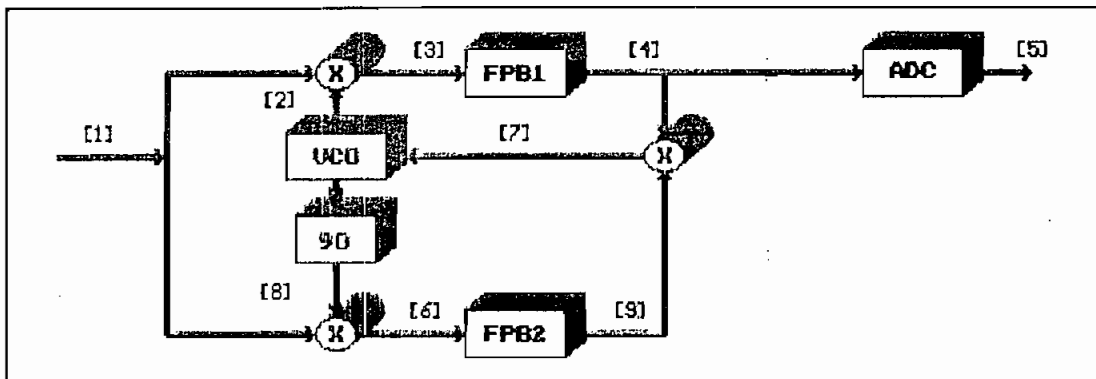


Figura 2.2 : Diagrama de bloques del bucle de COSTAS GENERAL

Si la señal modulada es

$$e_1(t) = X(t) A \cos(\omega_p t + \phi) \quad (2.13)$$

Para el caso analizado  $X(t)=1$  para el nivel lógico uno y  $X(t)=-1$  para el nivel cero lógico.

Las respuestas del VCO y del Defasador.

$$e_2(t) = C \cos(\omega_p t + k) \quad (2.14)$$

$$e_8(t) = C \text{sen}(\omega_p t + k) \quad (2.15)$$

Luego de los multiplicadores analógicos .

$$e_3(t) = \frac{A C}{2} X(t) \cos(\theta - k) + \frac{A C}{2} X(t) \cos(2\omega_p t + \theta + k) \quad (2.16)$$

$$e_6(t) = - \frac{A C}{2} X(t) \text{sen}(\theta - k) + \frac{A C}{2} X(t) \text{sen}(2\omega_p t + \theta + k) \quad (2.17)$$

Filtrando.

$$e_1(t) = \frac{A C}{2} X(t) \cos(\phi - k) \quad (2.18)$$

$$e_2(t) = -\frac{A C}{2} X(t) \sin(\phi - k) \quad (2.19)$$

Obteniéndose así, la señal de error de fase del lazo.

$$e_3(t) = -\frac{A^2 C^2}{8} X^2(t) \sin[2(\phi - k)] \quad (2.20)$$

Esta señal, de control del VCO, hará variar la fase hasta que  $\phi = k$  obteniéndose que  $e_1(t) = ACX(t)/2$  que es la señal de información demodulada.

### 2.3.- BUCLE CON REMODULACION POR EL OSCILADOR LOCAL

Otro de los sistemas para evitar la influencia de los cambios de fase producidos por la señal modulada es el siguiente .

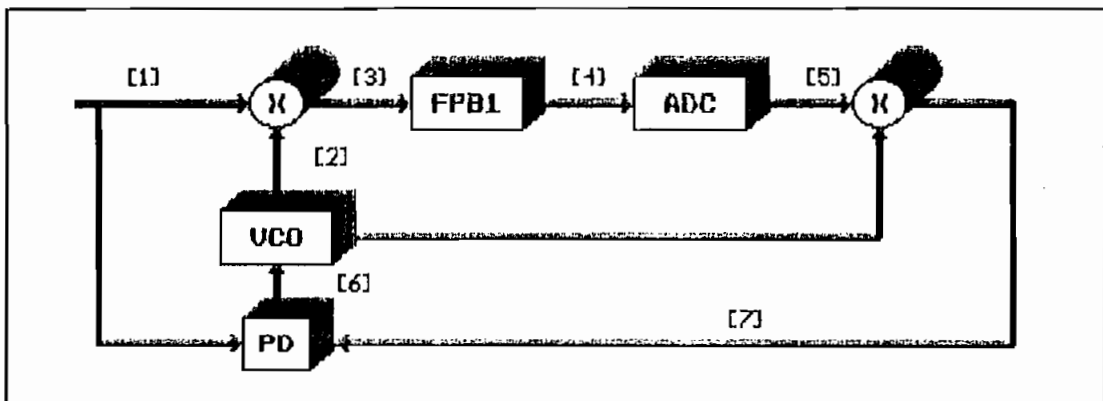


Figura 2.3 : Diagrama de bloques del bucle con REMODULACION por el OSCILADOR LOCAL

Una vez que la portadora recuperada  $e_2(t)$  está sincronizada con la señal original usada en la modulación,

las señales moduladas  $e_1(t)$  y  $e_7(t)$  son exactamente iguales, excepto en el corto intervalo que se da desde el cambio de fase de  $e_1(t)$ , el demodulador cambia el nivel de datos y éste varía la modulación de  $e_7(t)$ .

Como se puede apreciar la señal recuperada  $e_2(t)$ , se utiliza como portadora para la demodulación coherente y para volver a modular la señal de datos recuperados en el demodulador, de allí el nombre del bucle.

Siendo la señal modulada recibida y la respuesta del VCO las ondas (2.13) y (2.14) respectivamente. Las ondas producto de estas dos  $e_3(t)$ , la filtrada  $e_4(t)$  y la ecualizada  $e_5(t)$  serán iguales a las del bucle anterior, es decir el bloque demodulador es el mismo para los dos casos.

Utilizando la señal detectada  $e_5(t)$  y la portadora recuperada  $e_2(t)$ , se hace una remodulación, obteniéndose  $e_7(t)$  que es la señal a compararse con la entrada al bucle, para obtener el error de fase  $e_6(t)$ , que constituye la señal de control de VCO.

$$e_7(t) = e_5(t) C \cos(\omega_p t + k) \quad (2.21)$$

$$e_6(t) = \frac{A C}{2} X(t) e_5(t) [f(n\omega_p t) + \text{sen}(\phi - k)] \quad (2.22)$$

## 2.4 BUCLE CON REMODULACION POR LA SEÑAL RECIBIDA

Otra opción con remodulación esta dado por diagrama :



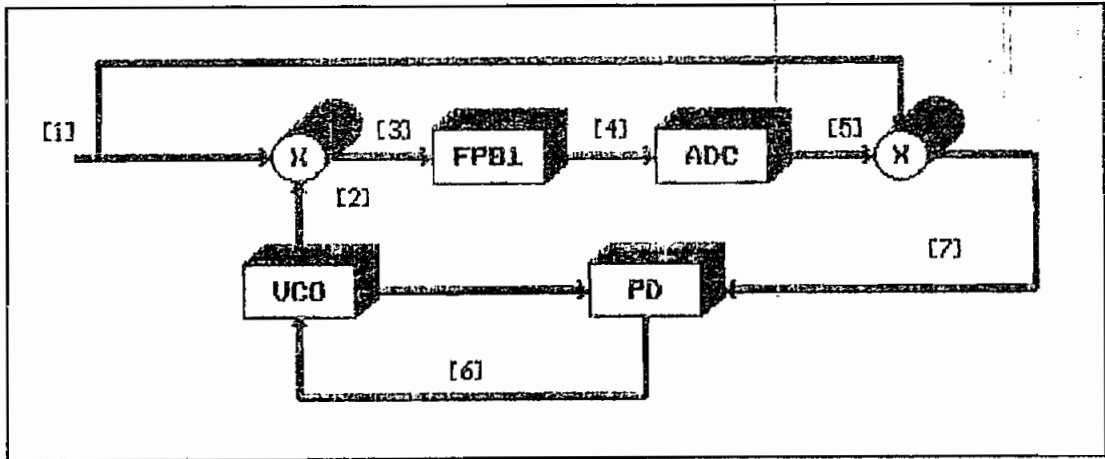


Figura 2.4 : Diagrama de bloques del bucle con REMODULACION por la SEÑAL RECIEIDA

Siendo las señales  $e_1(t)$  y  $e_2(t)$  las mismas que la del bucle con remodulación anterior, el bloque demodulador cuyas ondas involucradas  $e_3(t)$ ,  $e_4(t)$  y  $e_5(t)$  tienen su comportamiento igual que en el bucle mencionado, mientras que las demás señales tienen la forma:

$$e_7 = A e_5(t) X(t) \cos(\omega_p t + \theta) \quad (2.23)$$

$$e_6(t) = \frac{A C}{2} e_5(t) X(t) [f(n\omega_p t) + \text{sen}(\theta - k)] \quad (2.24)$$

Como en el caso anterior utiliza la señal de datos recuperada en el bloque demodulador para realizar una nueva modulación, pero esta vez se utiliza como portadora la señal modulada que llega al receptor. Esta remodulación se prefiere cuando se recibe en el bucle una señal aceptable en cuanto a deformaciones.

## 2.5. BUCLE DE COSTAS CON DECISION DENTRO DE UN BRAZO

Por razones de construcción física asociada con la reducción de offset dc, es deseable tener un limitador a la

salida del canal en fase, de este modo, se puede reemplazar el multiplicador analógico (cuatro cuadrantes), con un simple switch controlado o un multiplicador de dos cuadrantes, el que típicamente presenta un offset menor.

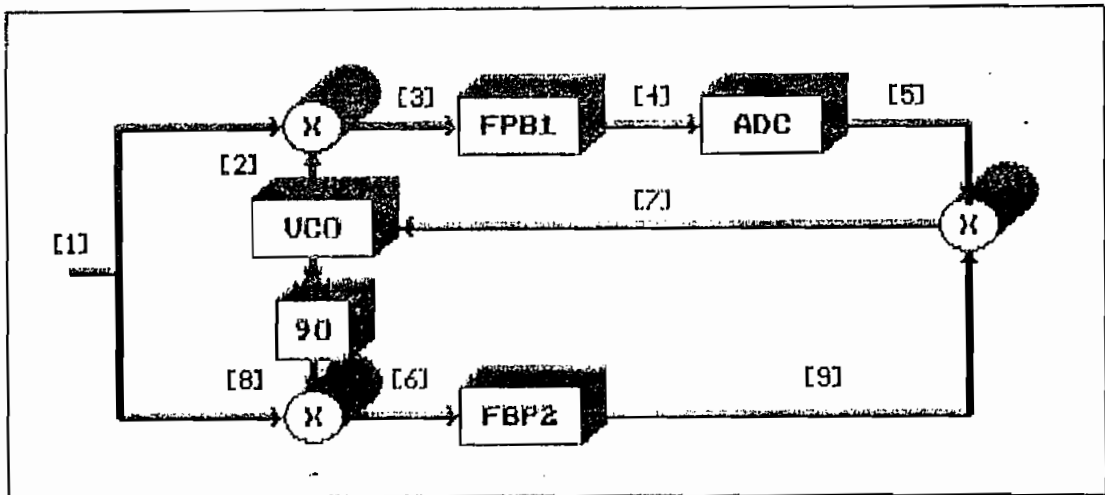


Figura 2.5 : Diagrama de bloques del bucle de COSTAS con DETECCIÓN dentro de un brazo

El bloque demodulador es idéntico al correspondiente en el bucle de Costas General ya analizado, igualmente las respuestas de los bloques del brazo en cuadratura son idénticas a las del mencionado bucle. Pero al tener la equalización dentro de un brazo, la señal de control del VCO es:

$$e_7(t) = - \frac{A C}{2} X(t) e_5(t) \text{sen}(\phi - k) \quad (2.25)$$

Esta señal tiene la componente  $\text{sen}(\phi - k)$  que tiende a sincronizar el lazo, igual que en el bucle anteriormente referido.

## 2.6.- BUCLE DE COSTAS NUMERICO

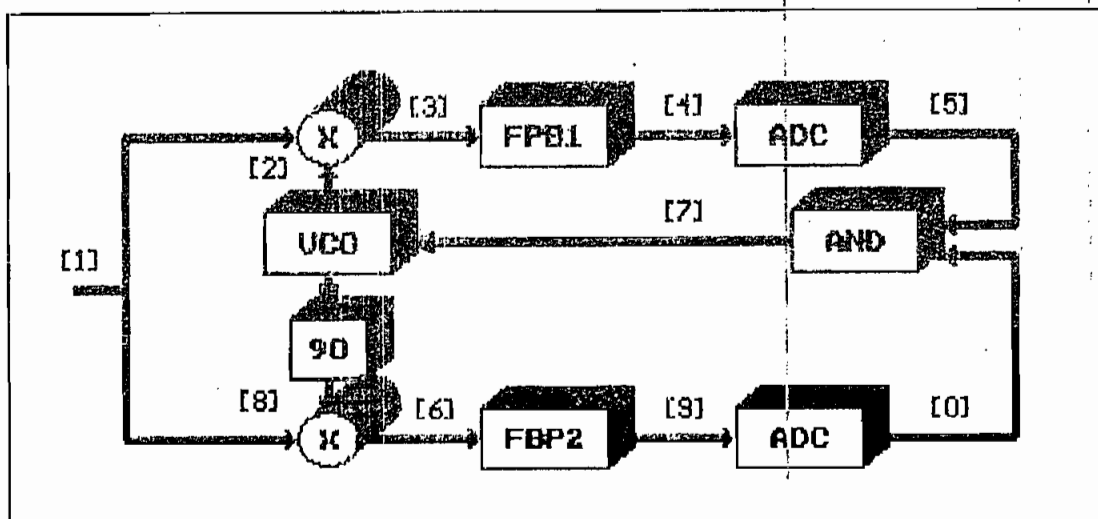


Figura 2.6 : Diagrama de bloques del bucle de COSTAS NUMERICO

Las señales  $e_1(t)$ ,  $e_2(t)$ ,  $e_3(t)$ ,  $e_4(t)$  y  $e_5(t)$  correspondientes al bloque demodulador y  $e_6(t)$ ,  $e_7(t)$ ,  $e_8(t)$  son idénticas a las analizadas en el Bucle de Costas general. Pero para obtener el error de fase se ha ubicado ecualizadores tanto en el brazo en Cuadratura, como en el en fase evitándose así, el usar un multiplicador analógico, al que se lo ha substituido por una simple compuerta AND; digitalizándose esta parte del lazo, de ahí su nombre.

Tabla 2.1: Comportamiento de la compuerta AND.

$e_5(t)$	$e_6(t)$	$e_7(t)$
0	0	0
0	1	0
1	0	0
1	1	1

Recordemos las señales  $e_6(t)$  y  $e_4(t)$  que nos son útiles

para el análisis de la señal de error de fase.

$$e_4(t) = \frac{A C}{2} X(t) \cos(\phi - k)$$

$$e_5(t) = -\frac{A C}{2} X(t) \operatorname{sen}(\phi - k)$$

Tanto  $e_4(t)$  como  $e_5(t)$  dependen de la relación  $\phi - k$ , considerando un valor positivo como 1 lógico y un valor negativo como 0 lógico, se tiene la siguiente tabla.

Tabla 2.2: Influencia del error de fase.

	$e_5(t)$	$e_4(t)$	$e_7(t)$
$0^\circ < \phi - k < 90^\circ$	1	0	0
$90^\circ < \phi - k < 180^\circ$	0	0	0
$180^\circ < \phi - k < 270^\circ$	0	1	0
$270^\circ < \phi - k < 360^\circ$	1	1	1

Cuando el bucle no está sincronizado  $\phi$  es diferente de  $k$ . La tabla anterior demuestra que este bucle tendrá un ajuste de fase inmediato únicamente en un rango de desfase, en el que  $e_7(t)$  es diferente de cero ( $270^\circ < \phi - k < 360^\circ$ ).

Si el bucle está en fase  $e_4(t)$  es cero, por lo que la señal de error también es nula.

## 2.7 .- BUCLE DE COSTAS PARA QPSK

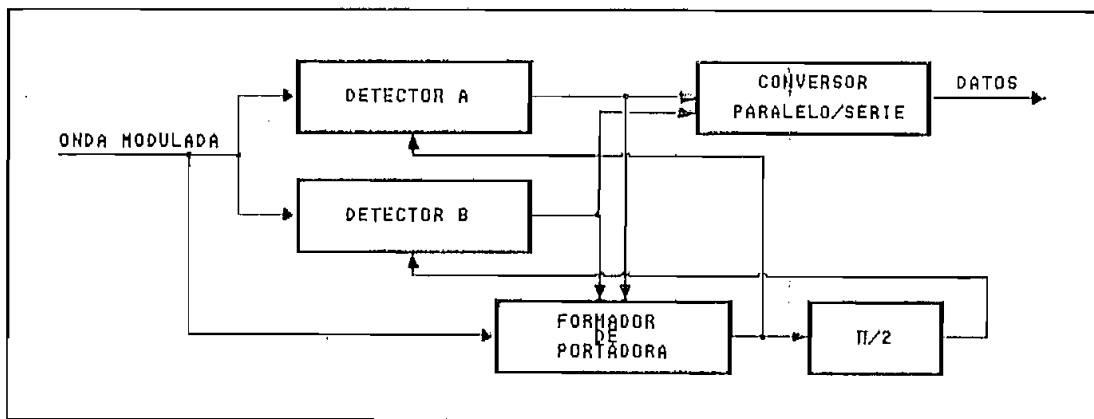


Figura 2.7 : Diagrama simplificado

Un diagrama de bloques del circuito recuperador de portadora únicamente es:

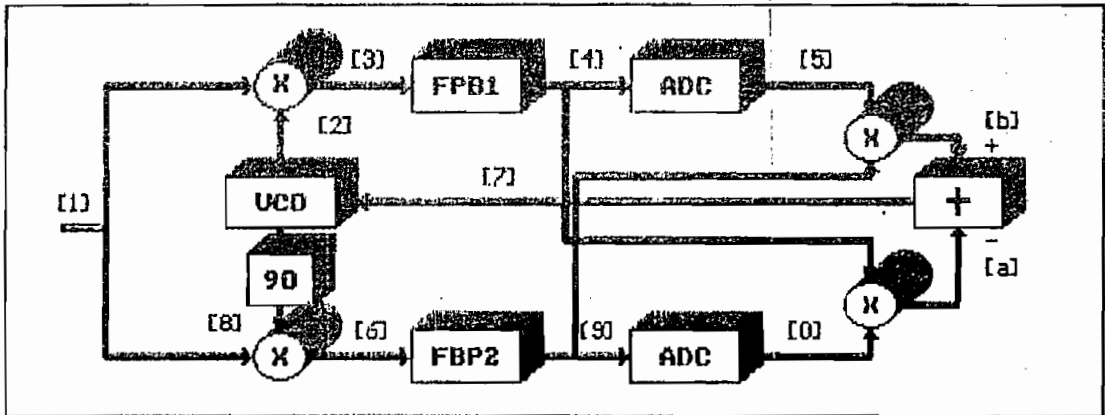


Figura 2.8 : Diagrama del circuito recuperador de sincronismo.

En este bucle los dos brazos en cuadratura son diferentes de cero aún cuando el sistema está sincronizado, lo que no sucedía para BPSK; ya que tanto para  $\phi(t)=\pi$  como para  $\phi(t)=0$ , se tenía  $\sin[\phi(t)]=0$ .

Si la señal modulada es :

$$e_1(t) = A \cos(\omega_p t + \phi(t)) \quad (2.26)$$

Donde

$$\phi(t) = \begin{cases} \pi/4 & \text{para } 01 \\ 3\pi/4 & \text{para } 00 \\ 5\pi/4 & \text{para } 10 \\ 7\pi/4 & \text{para } 11 \end{cases}$$

Las respuestas de los bloques son:

$$e_2(t) = C \cos(\omega_p t + k) \quad (2.27)$$

$$e_4(t) = \frac{A C}{2} \cos(\phi(t) - k) \quad (2.28)$$

$$e_9(t) = -\frac{A C}{2} \operatorname{sen}(\phi(t) - k) \quad (2.29)$$

$$e_b(t) = -\frac{A C}{2} e_5(t) \operatorname{sen}(\phi(t) - k) \quad (2.30)$$

$$e_a(t) = \frac{A C}{2} e_0(t) \cos(\phi(t) - k) \quad (2.31)$$

$$e_7(t) = e_b(t) - e_a(t) \quad (2.32)$$

Cuando el lazo está sincronizado,  $k=0$ .

El funcionamiento del lazo se puede resumir en la siguiente tabla.

Tabla 2.3: Resumen del comportamiento ideal del Bucle

$\phi(t)$	$e_4(t)$	$e_9(t)$	$e_5(t)$	$e_0(t)$	$e_b(t)$	$e_a(t)$	$e_7(t)$
$\pi/4$	$Rf2/2$	$-Rf2/2$	$m$	$-m$	$-Rmf2/2$	$-Rmf2/2$	$0$
$3\pi/4$	$-Rf2/2$	$-Rf2/2$	$-m$	$-m$	$Rmf2/2$	$Rmf2/2$	$0$
$5\pi/4$	$-Rf2/2$	$Rf2/2$	$-m$	$m$	$-Rmf2/2$	$-Rmf2/2$	$0$
$7\pi/4$	$Rf2/2$	$Rf2/2$	$m$	$m$	$Rmf2/2$	$Rmf2/2$	$0$

Se observa que  $e_7(t)=0$  si  $k=0$ , cumpliéndose así los requerimientos de la señal de control del VCO.

La señal binaria original se obtiene de las señales recuperadas  $e_5(t)$  y  $e_0(t)$  mediante un reloj que permite intercalar las dos señales en un intervalo de tiempo que comprende un estado de señal modulada.

# CAPITULO III

## DISCRETIZACION DE LAS RESPUESTAS DE CADA COMPONENTE UTILIZADO PARA LA SIMULACION

TEMA	PAGINA
3.1. Métodos de discretización de señales analógicas	41
3.1.1. Transformación Invariante al Impulso	42
3.1.2. Métodos de la solución numérica de las ecuaciones diferenciales	44
3.1.2.1 Diferencias hacia atrás	45
3.1.2.2 Diferencias hacia adelante	46
3.1.3. Transformación Bilineal	47
3.2. Discretización de Filtros Analógicos	
3.2.1. Filtros de Butterworth y de Tchebyscheff	49
3.2.2. Filtros Elípticos	52
3.3. Discretización del Defasador, Multiplicador de frecuencia, Comparador de fase y Oscilador Controlado por Voltaje	
3.3.1. Defasador de 90 grados	55
3.3.2. Multiplicador de frecuencia	59
3.3.3. Comparador de fase	64
3.3.4. Oscilador Controlado por Voltaje	64

## CAPITULO III

DISCRETIZACION DE LAS RESPUESTAS DE CADA  
COMPONENTE UTILIZADA PARA LA SIMULACION

## 3.1.- METODOS DE DISCRETIZACION DE SEÑALES ANALÓGICAS.

Para la simulación se requiere transformar los diferentes componentes analógicos de los bucles a sistemas discretos.

En general un sistema analógico puede ser representado en frecuencia y en tiempo. En las expresiones siguientes  $Y_a(t)$  es la respuesta a la excitación  $X_a(t)$ , mientras que  $h_a(t)$  es la respuesta a la función impulso del sistema; finalmente  $f_i$  y  $g_i$  son constantes.

## REPRESENTACION EN FRECUENCIA:

Función de transferencia:

$$H_a(s) = \frac{\sum_{i=0}^M f_i S^i}{\sum_{i=0}^N g_i S^i} = \frac{Y_a(s)}{X_a(s)} \quad (3.1)$$

## REPRESENTACION EN EL TIEMPO:

Integral de Convolución:

$$Y_a(t) = \int_{-a}^a X_a(\tau) h_a(t-\tau) d\tau \quad (3.2)$$

Ecuación diferencial

$$\sum_{i=0}^N g_i \frac{d^i}{dt^i} (Y_a(t)) = \sum_{i=0}^M f_i \frac{d^i}{dt^i} (X_a(t)) \quad (3.3)$$

Un sistema discreto, también tiene su representación en



tiempo y en frecuencia. En las expresiones siguientes  $Y(Z)$  y  $X(Z)$  son las transformadas  $Z$  de  $Y_a(t)$  y  $X_a(t)$  respectivamente;  $Y(n)$  es la respuesta a la excitación  $X(n)$ , mientras que  $h(n)$  es la respuesta a la función impulso del sistema, finalmente  $a_i$  y  $b_i$  son constantes.

REPRESENTACION EN FRECUENCIA :

Función de Transferencia.

$$H(Z) = \frac{\sum_{i=0}^M b_i Z^{-i}}{\sum_{i=0}^N a_i Z^{-i}} = \frac{Y(Z)}{X(Z)} \quad (3.4)$$

PRESENTACION EN EL TIEMPO.

Sumatorio de Convolución

$$Y(n) = \sum_{i=-\infty}^{\infty} X(n-i) h(n-i) \quad (3.5)$$

Sistema de ecuaciones de diferencias

$$\sum_{i=0}^M a_i Y(n-i) = \sum_{i=0}^N b_i X(n-i) \quad (3.6)$$

Dentro de los métodos de discretización se tiene varias alternativas:

### 3.1.1.1.- TRANSFORMACION INVARIANTE AL IMPULSO

Este método consiste en elegir la respuesta impulsiva del sistema discreto, tomando muestras a intervalos equiespaciados de la respuesta impulsiva análoga correspondiente, así:  $h(n) = h(nT) = h_a(nT)$ . Donde  $T$  es el intervalo de muestreo, y cumple con el teorema de Nyquist.

La función de transferencia de un sistema analógico recursivo se puede descomponer en elementos de primer orden.

$$H_a(s) = \frac{N(s)}{D(s)} = \sum_{l=1}^N \frac{A_l}{s-s_l} \quad (3.7)$$

Pasando al dominio del tiempo

$$h_a(t) = \alpha^{-1} [H_a(s)] = \sum_{l=1}^N A_l e^{s_l t} u(t) \quad (3.8)$$

Tomando valores de  $h_a(t)$  a intervalos espaciados a  $T$  segundos:

$$h(nT) = \sum_{l=1}^N A_l e^{s_l nT} u(n) \quad (3.9)$$

$$h(nT) = \sum_{l=1}^N A_l (e^{s_l T})^n u(n) \quad (3.10)$$

Pasando al dominio de la frecuencia discreta (Plano Z)

$$Z[h(nT)] = A_l + A_l e^{s_l T} Z^{-1} + A_l e^{s_l T} Z^{-2} + \dots \quad (3.11)$$

Obteniéndose la función del sistema discreto:

$$H(z) = \sum_{l=1}^N \frac{A_l}{1 - e^{s_l T} Z^{-1}} \quad (3.12)$$

Comparando (3.7) y (3.12), se observa que los coeficientes se mantienen mientras que los polos del plano S, se transforman en polos  $e^{s_l T}$  en el plano Z. Esto es:

$$z_l = e^{s_l T}$$

Siendo  $\Omega$  frecuencia análoga [rad/seg] y  $\omega$  frecuencia digital [rad] donde  $\omega = \Omega T$  por el muestreo realizado, se puede

demostrar que:

$$H(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H(j\Omega + j\frac{2\pi k}{T}) \quad (3.13)$$

La respuesta de frecuencia del sistema discreto es periódica, por cada tramo  $2\pi/T$  en el plano S recorreremos una vez el círculo unitario en el plano Z

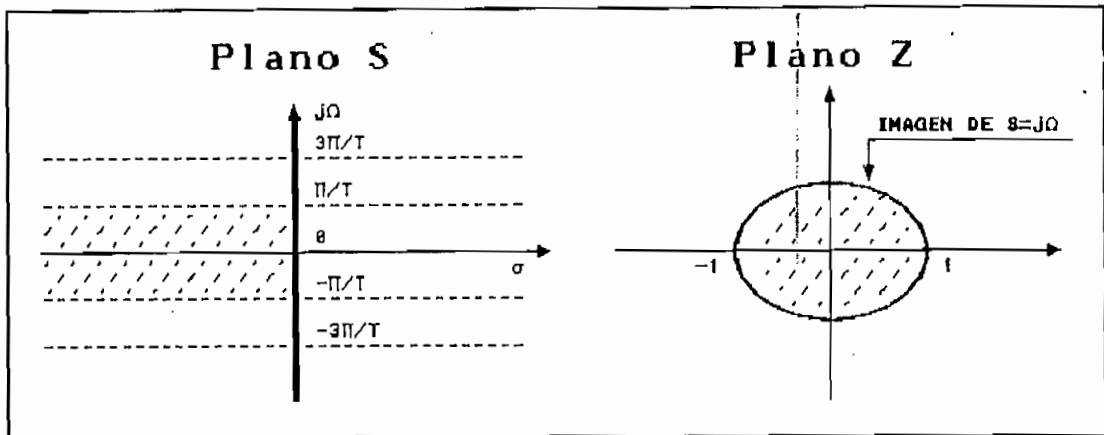


Figura 3.1 : Correspondencia entre los planos S y Z usando la Transformación Invariante al Impulso

Debido a la periodicidad, se puede dar cruce de espectros si no se toma las medidas necesarias para evitarlo. En consecuencia, el método es apropiado para sistemas cuya respuesta  $h(t)$  no tenga componentes de alta frecuencia.

Un aspecto favorable que se nota, es que se mantiene una relación lineal entre las frecuencias análoga y digital.

### 3.1.2.- METODO DE LA SOLUCION NUMERICA DE LAS ECUACIONES DIFERENCIALES

Este método consiste en aproximar las derivadas de (3.3) por diferencias finitas entre valores consecutivos que

toma la señal en los instantes de muestreo. De este modo la ecuación (3.3) toma la forma de la ecuación (3.6).

3.1.2.1. Diferencias hacia atrás.- En esta aproximación se reemplazan las derivadas por la diferencia entre el instante actual y el correspondiente al instante de muestreo previo.

$$\frac{d^n Y(KT)}{dt^n} = \frac{\left[ \frac{d^{n-1} Y(KT)}{dt^{n-1}} - \frac{d^{n-1} Y(KT-T)}{dt^{n-1}} \right]}{T} \quad (3.14)$$

La transformada de Laplace la operación de derivación transforma en el operador  $s$ , asumiendo condiciones iniciales nulas, se tiene.

$$s^n Y(KT) = \frac{s^{n-1} Y(KT) - s^{n-1} Y(KT-T)}{T} \quad (3.15)$$

Pasando al dominio  $Z$ :

$$s^n Y(z) = \frac{s^{n-1} Y(z) - s^{n-1} Z^{-1} Y(z)}{T} \quad (3.16)$$

Resolviendo esta última expresión se obtiene:

$$s = \frac{1 - Z^{-1}}{T} \quad (3.17)$$

Siendo  $s = j\Omega$ , partiendo de (3.17), se llega a la relación:

$$Z = \frac{1}{2} [1 + e^{j2 \arctan \Omega T}] \quad (3.18)$$

Dando valores a esta última ecuación citada, se observa que la transformada no cubre el círculo unitario en el plano  $Z$ , como muestra la figura.

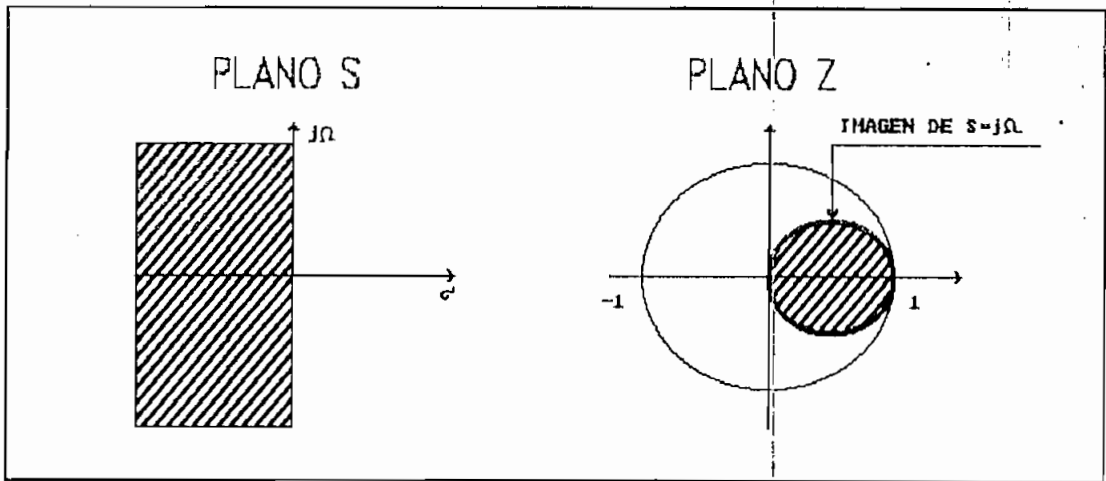


Figura 3.2 : Correspondencia entre los planos S y Z usando el Método de solución numérica de la ecuación diferencial con Diferencias hacia atrás

Para que coincida la transformada con el círculo unitario,  $\Omega T$  debe ser alto. Concluyendo de este modo, que este método presenta desventajas como: requiere de una frecuencia de muestreo de la señal excesivamente alta y produce desplazamiento de frecuencia. Pero al no ser periódica, se ha logrado evitar el cruce de espectros.

3.1.2.2. Diferencias hacia adelante.- En esta aproximación se reemplazan las derivadas por la diferencia entre el muestreo próximo y el correspondiente al instante de muestreo actual.

$$\frac{d^n Y(KT-T)}{dt^n} = \frac{\left[ \frac{d^{n-1} Y(KT)}{dt^{n-1}} - \frac{d^{n-1} Y(KT-T)}{dt^{n-1}} \right]}{T} \quad (3.19)$$

Con un procedimiento similar al anterior, se deduce que esta transformación da la relación:

$$s = \frac{Z - 1}{T} \quad (3.20)$$

### 3.1.3.- METODO DE LA TRANSFORMADA BILINEAL

Este método ha sido denominado también como de Integración Trapezoidal, al igual que los tratados en el punto anterior, se basa en la integración numérica, pero este resulta ser el de mayor precisión. Se basa en la aproximación numérica de una integral. ya que la función o curva original puede ser aproximada sumando las áreas bajo los polígonos de ancho  $T$ .

Para una ecuación diferencial de primer orden, su función de transferencia es:

$$Ha(s) = \frac{d_o}{C_1 s + C_o} \quad (3.21)$$

Se puede escribir  $Y_a(t)$  como una integral de  $Y_a'(t)$  como:

$$Y_a(t) = \int_{t_o}^t Y_a'(t) dt + Y_a(t_o) \quad (3.22)$$

Si  $t = nT$  y  $t_o = (n-1)T$

$$Y_a(nT) = \int_{(n-1)T}^{nT} Y_a'(nT) dnT + Y_a((n-1)T) \quad (3.23)$$

Usando la aproximación de la integral con la regla del trapecio.

$$Y_a(nT) = \frac{T}{2} [Y_a'((n-1)T) + Y_a'(nT)] + Y_a((n-1)T) \quad (3.24)$$

Pasando (3.21) al dominio del tiempo y evaluando  $t$  en  $nT$ .

$$Y_a'(nT) = -\frac{C_o}{C_1} Y_a(nT) + \frac{d_o}{C_1} X(nT) \quad (3.25)$$

Reemplazando (3.25) en (3.24) y haciendo por simplicidad:  
 $Y_a(nT) = Y(n)$  y  $Y_a((n-1)T) = Y(n-1)$  se tiene.

$$Y(n) - Y(n-1) = \frac{T}{2} \left[ -\frac{C_0}{C_1} (Y(n) + Y(n-1)) + \frac{d_0}{C_1} (X(n) + X(n-1)) \right] \quad (3.26)$$

Tomando la transformada Z

$$Y(Z) \left( 1 + \frac{C_0 T}{C_1} \right) - Y(Z) Z^{-1} \left( 1 - \frac{C_0 T}{C_1} \right) = \frac{T d_0}{2 C_1} X(Z) (1 + Z^{-1}) \quad (3.27)$$

Finalmente su función de transferencia es

$$H(Z) = \frac{d_0}{C_1 \frac{2}{T} \frac{1-Z^{-1}}{1+Z^{-1}} + C_0} \quad (3.28)$$

Comparando las funciones de transferencia (3.21) y (3.28) se observa que:

$$H(Z) = H_a(s) \Big|_{s = (2/T)(1-Z^{-1})/(1+Z^{-1})} \quad (3.29)$$

Se puede demostrar que esta transformación válida para una ecuación diferencial de primer orden es también válida para ecuaciones diferenciales de cualquier orden.

Para observar la respuesta de frecuencia reemplazamos Z por  $e^{j\omega}$  en la fórmula:

$$s = \frac{2}{T} \frac{1 - Z^{-1}}{1 + Z^{-1}} \quad (3.30)$$

Obteniéndose

$$s = \frac{2}{T} j \operatorname{tg} \frac{\omega}{2} \quad (3.31)$$

Siendo la relación entre frecuencias analógica y digital.

$$\Omega = \frac{2}{T} \operatorname{tg} \frac{W}{2} \quad (3.32)$$

La transformada bilineal convierte todo el eje imaginario del plano S en el círculo unitario del plano Z (una sola vez).

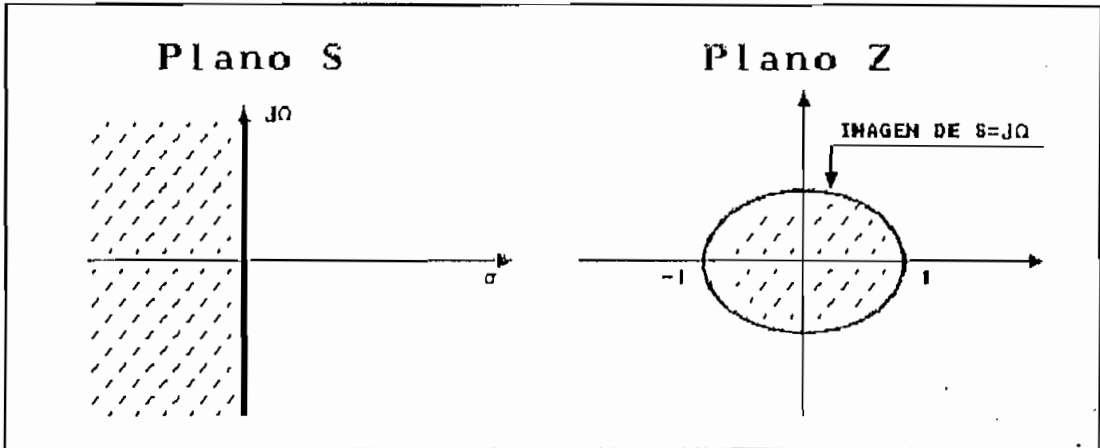


Figura 3.3 : Correspondencia entre los planos S y Z usando la Transformación Bilineal

Se concluye que el método presente la ventaja de evitar el cruce de espectros pero al igual que el caso anterior produce el desplazamiento de frecuencias.

### 3.2.- DISCRETIZACION DE FILTROS DE BUTTERWORTH, TCHEBYSCHIEFF Y ELIPTICOS

#### 3.2.1.- FILTROS DE BUTTERWORTH Y TCHEBYSCHIEFF DIRECTO

Determinada la función de transferencia de los filtros.

$$H(s) = \frac{H_0}{\prod_{l=1}^N s - P_l} \quad (3.33)$$



Cuando  $N$  es impar existen polos conjugados y uno sobre el eje real ( $P_r$ ) en el plano  $S$ . Agrupando polos conjugados.

$$H(s) = \frac{H_0}{D(s)} \prod_{i=1}^r \frac{1}{s^2 + b_i s + c_i} \quad (3.34)$$

Donde

$$D(s) = \begin{cases} 1 & N \text{ par} \\ s+P_r & N \text{ impar} \end{cases}$$

Usando la transformada bilineal, esto es reemplazando  $s$  por  $2/T[(1-Z^{-1})/(1+Z^{-1})]$  en (3.34), ésta toma la forma.

$$H(s) = \frac{H_0}{D(Z)} \prod_{i=1}^r \frac{1+A_i Z^{-1}+D_i Z^{-2}}{1-B_i Z^{-1}-C_i Z^{-2}} \quad (3.35)$$

Donde

$$D(Z) = \begin{cases} 1 & \text{para } N \text{ par} \\ \frac{(\frac{2}{T}+P_r) - (\frac{2}{T}-P_r) Z^{-1}}{1+Z^{-1}} & \text{para } N \text{ impar} \end{cases}$$

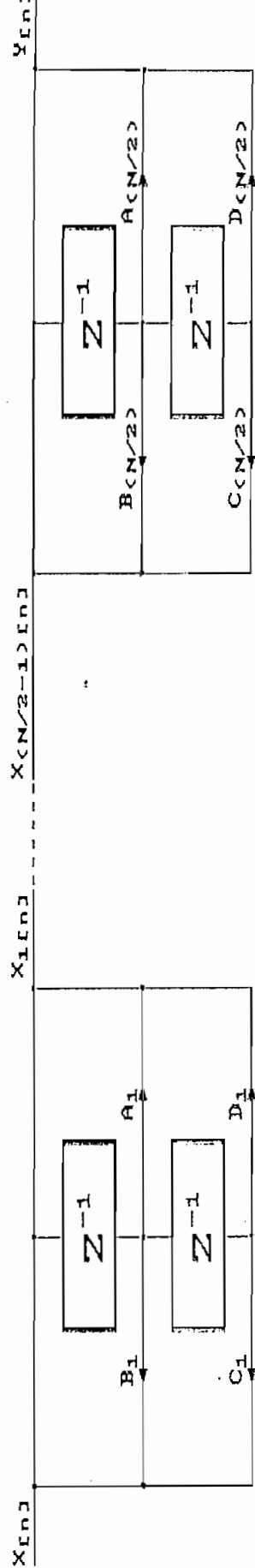
Cada multiplicando, pasando al dominio del tiempo.

$$Y_1[n] = B_1 Y[n-1] + C_1 Y[n-2] + X_1[n] + A_1 X_1[n-1] + D_1 X_1[n-2] \quad (3.36)$$

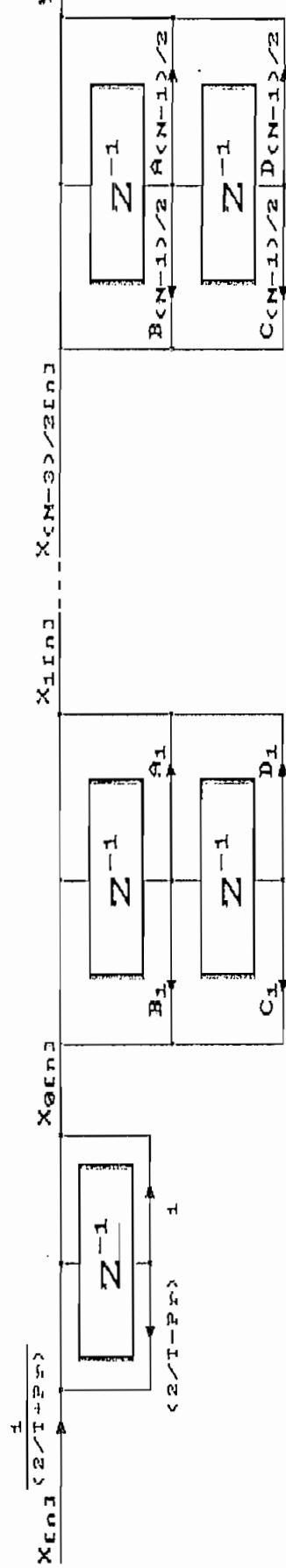
Mientras que para  $D(z)$ , toma la forma:

$$Y[n] = \frac{1}{(\frac{2}{T}+P_r)} \left[ (\frac{2}{T}-P_r) Y[n-1] + X[n] + X[n-1] \right] \quad (3.37)$$

Para orden del filtro ( N ) par:



Para orden del filtro ( M ) impar:



## 3.2.2.-DISCRETIZACION DE FILTROS ELIPTICOS:

Determinados los coeficientes, se tiene la siguiente función de transferencia:

$$H_N(s) = \frac{H_0}{D(s)} \prod_{i=1}^r \frac{s^2 + A_i}{s^2 + B_i s + C_i} \quad (3.38)$$

Donde:

$$r = \begin{cases} \frac{N-1}{2} & N \text{ impar} \\ \frac{N}{2} & N \text{ par} \end{cases}$$

$$D(s) = \begin{cases} S + \sigma & N \text{ impar} \\ 1 & N \text{ par} \end{cases}$$

Usando la transformada bilineal, para discretizar el filtro analógico, se tiene la función de transferencia en el dominio Z.

$$H_N(z) = \frac{H_0}{D(z)} \prod_{i=1}^r \frac{g_2 z^{-2} + g_1 z^{-1} + g_0}{G_2 z^{-2} + G_1 z^{-1} + G_0} \quad (3.39)$$

Con

$$\begin{aligned} g_2 &= e + A_i \\ g_1 &= 2 (A_i - e) \\ g_0 &= g_2 \\ G_2 &= e - B_i m + C_i \\ G_1 &= 2 (C_i - e) \\ G_0 &= e + B_i m + C_i \end{aligned}$$

$$e = (2/T)^2 \quad y \quad m = 2/T$$

Donde

$$D(Z) = \begin{cases} \frac{(m+\sigma) - (m-\sigma)z^{-1}}{1+z^{-1}} & N \text{ impar} \\ 1 & N \text{ par} \end{cases}$$

Cada multiplicando, pasando al dominio del tiempo queda:

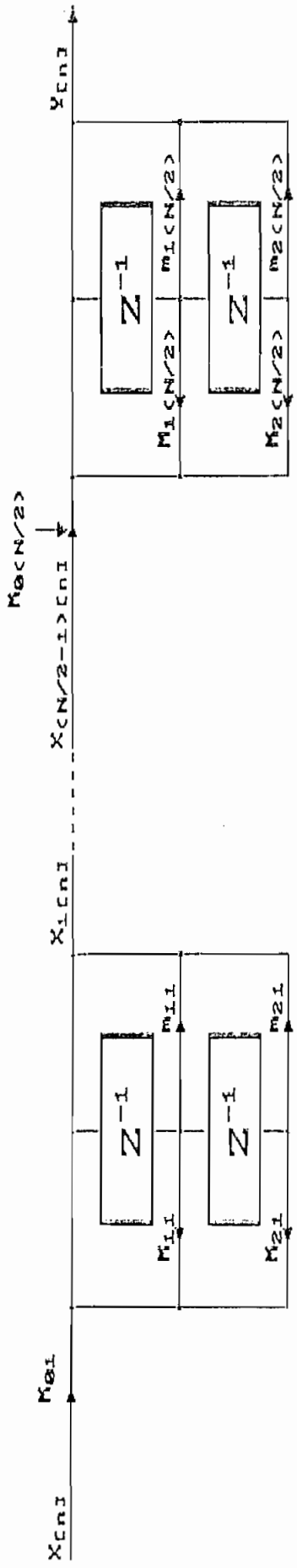
$$Y_1[n] = \frac{1}{G_0} (G_1 Y[n-1] + G_2 Y[n-2] + g_0 X_1[n] + g_1 X_1[n-1] + g_2 X_1[n-2]) \quad (3.40)$$

Mientras que para  $D(z)$ , toma la forma:

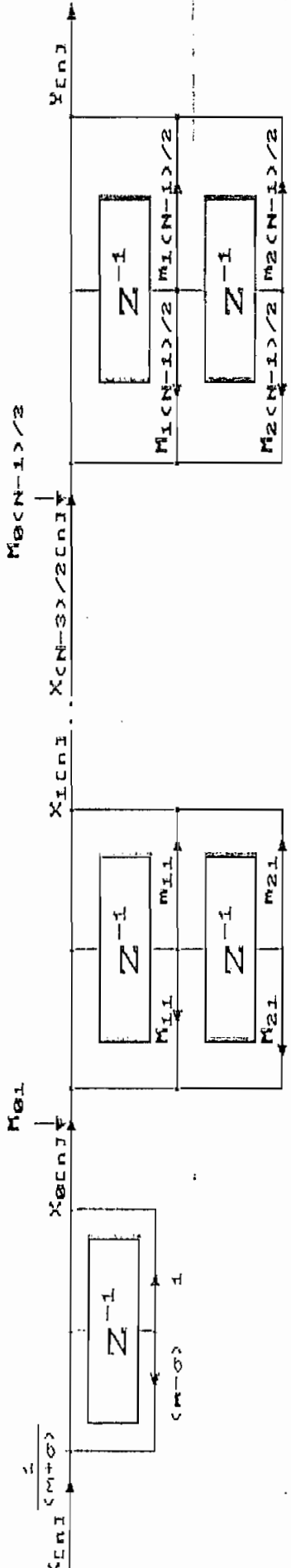
$$Y[n] = \frac{1}{(m+\sigma)} [(m-\sigma) Y[n-1] + X[n] + X[n-1]] \quad (3.41)$$

Su diagrama de flujo tiene componentes en cascada cuyo número depende del orden del filtro  $N$ . Se tiene dos componentes básicas una de primer orden semejantes al dado en los filtros anteriores, y uno de segundo orden que se diferencia de los filtros mencionados por tener un coeficiente de multiplicación para toda la etapa y otro para la componente no retardada de la entrada a la etapa en cuestión.

Para orden del filtro ( N ) par:



Para orden del filtro ( N ) impar:



EQUIVALENCIAS:

- $M_{ei} = G_{ei} / g_{ei}$
- $M_{1i} = G_{1i} / g_{ei}$
- $M_{2i} = G_{2i} / g_{ei}$
- $m_{1i} = g_{1i} / g_{ei}$
- $m_{2i} = g_{2i} / g_{ei}$

### 3.3 DISCRETIZACION DEL DEFASADOR DE 90 GRADOS, MULTIPLICADOR DE FRECUENCIA, COMPARADOR DE FASE Y OSCILADOR CONTROLADO POR VOLTAJE

#### 3.3.1. DISCRETIZACION DEL DEFASADOR DE 90°

Se analiza un sistema que produce una diferencia de fase de 90° con un error inferior a  $\epsilon$ , a todas las componentes de la señal de entrada dentro de un rango de frecuencias ( $f_1, f_2$ ) dadas.

Un defasador puro, no debe alternar la señal dentro del rango de frecuencias dado, es decir  $|H(e^{j\omega})|=1$ . Lo expuesto, se representa en el plano Z como muestra la figura:

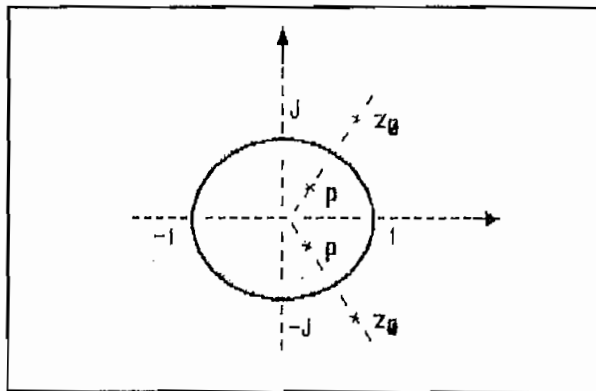


Figura 3.4 : Representación de polos y ceros de un Defasador puro en el plano Z

Del gráfico se deduce que los polinomios numerador y denominador de la función de transferencia son imagen, es decir, tienen los mismos coeficientes pero de orden inverso.

$$H(Z) = \frac{C + BZ^{-1} + Z^{-2}}{1 + BZ^{-1} + CZ^{-2}} \quad (3.42)$$

$$H(z) = \frac{(P-Z^{-1})(\bar{P}-Z^{-1})}{(1-PZ^{-1})(1-\bar{P}Z^{-1})} \quad (3.43)$$

La técnica de cálculo es la misma que para los filtros elípticos:

1.- Determinación del orden del circuito:

$$N = \frac{K(k_1) K(\sqrt{1-k^2})}{K(k) K(\sqrt{1-k_1^2})} \quad (3.44)$$

Donde  $k$  es la selectividad ( $k=\Omega_1/\Omega_2$ ). Debido a que se usa la transformada bilineal se hace el predesplazamiento de frecuencia.

$$k = \frac{\operatorname{tg}\left(\frac{\Omega_1 T}{2}\right)}{\operatorname{tg}\left(\frac{\Omega_2 T}{2}\right)} \quad (3.45)$$

$$k_1 = \left[ \frac{1 - \operatorname{tg}(\epsilon/2)}{1 + \operatorname{tg}(\epsilon/2)} \right]^2 \quad (3.46)$$

2.- Determinación de los cero  $Z_1$  de la función de transferencia en  $Z$ .

$$A = \operatorname{sn} \left[ \frac{(4i+1)K(\sqrt{1-k^2})}{2N}, \sqrt{1-k^2} \right] \quad (3.47)$$

$$B = \operatorname{cn} \left[ \frac{(4i+1)K(\sqrt{1-k^2})}{2N}, \sqrt{1-k^2} \right] \quad (3.48)$$

$$P_i = -\operatorname{tg} \left( \frac{\Omega_i}{2} \right) \frac{A}{B} \quad (3.49)$$

$$Z_i = \begin{cases} \frac{1+P_i}{1-P_i} & \text{para } 0 \leq i \leq N-1 \text{ si } P_i < 0 \\ \frac{1-P_i}{1+P_i} & \text{para } 0 \leq i \leq N-1 \text{ si } P_i > 0 \end{cases} \quad (3.50)$$

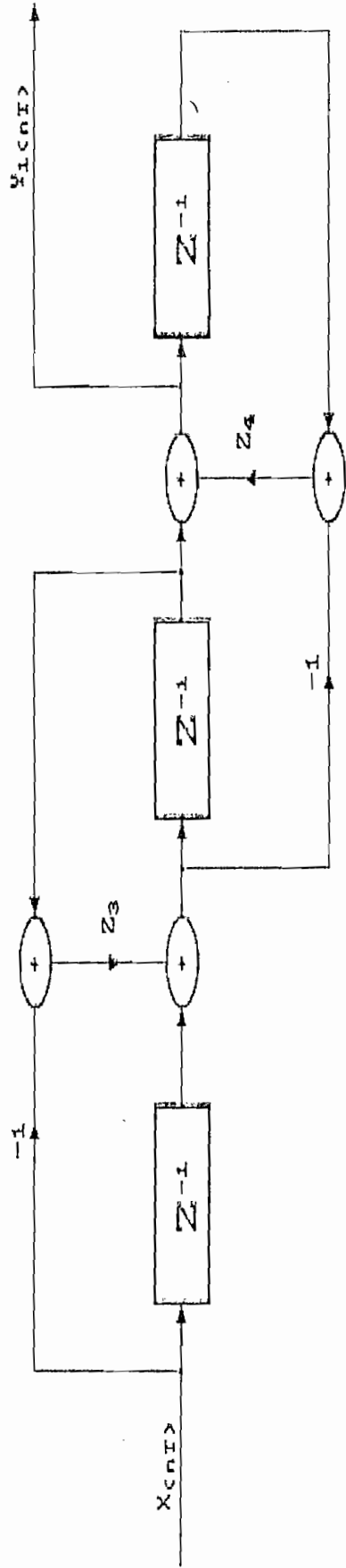
Conocidos los ceros ( $Z_i$ ), la función de transferencia del defasador queda determinada como:

$$H(Z) = \prod_{i=1}^N \frac{(Z_i - Z^{-1})}{(1 - Z_i Z^{-1})} \quad (3.51)$$

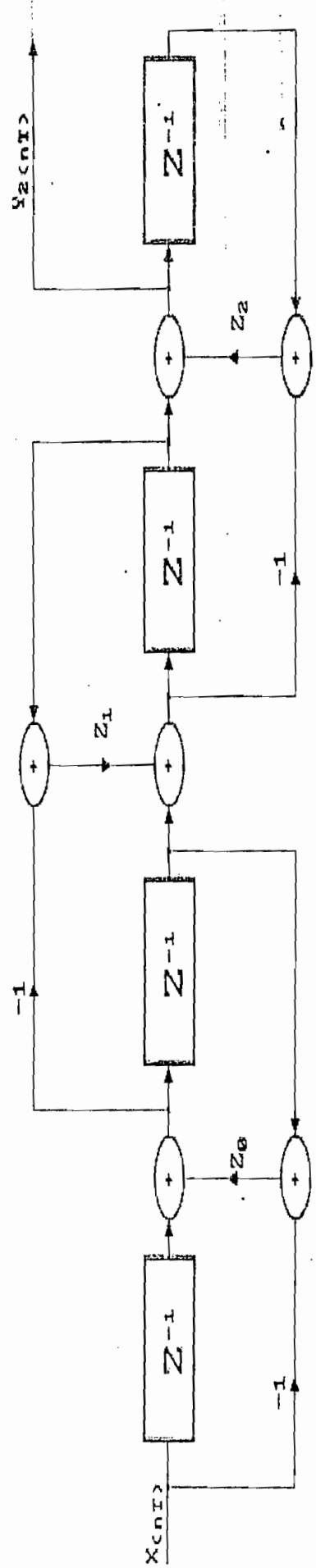
Diagramas de flujo de sistemas pasabanda con características de defasador de 90 grados en un rango de frecuencias dado se presentan a continuación.



Filtro de tercer orden ( N = 2 )



Filtro de tercer orden ( N = 3 )



### 3.3.2.- DISCRETIZACION DEL MULTIPLICADOR DE FRECUENCIA

Un multiplicador de frecuencia puede ser representado por el siguiente diagrama de bloque:

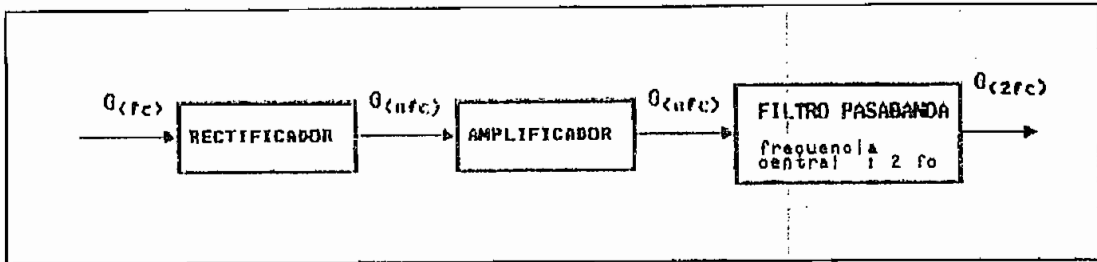


Figura 3.5 : Diagrama de bloques de un Multiplicador de Frecuencia.

En los bucles estudiados intervienen duplicadores de frecuencia, así, si  $f_c$  es la frecuencia principal de la señal, el filtro pasabanda debe estar centrado a  $2f_c$ .

El filtro pasabanda puede obtenerse a partir de la estructura de un filtro pasabajos, usando transformación de frecuencia, reemplazando en la función de transferencia del filtro pasabajos  $(s^2 + \Omega_0^2)/(sW)$  por  $s$ . Siendo  $W = \Omega_p^+ - \Omega_p^-$  y  $\Omega_0$  la frecuencia central del filtro deseado.

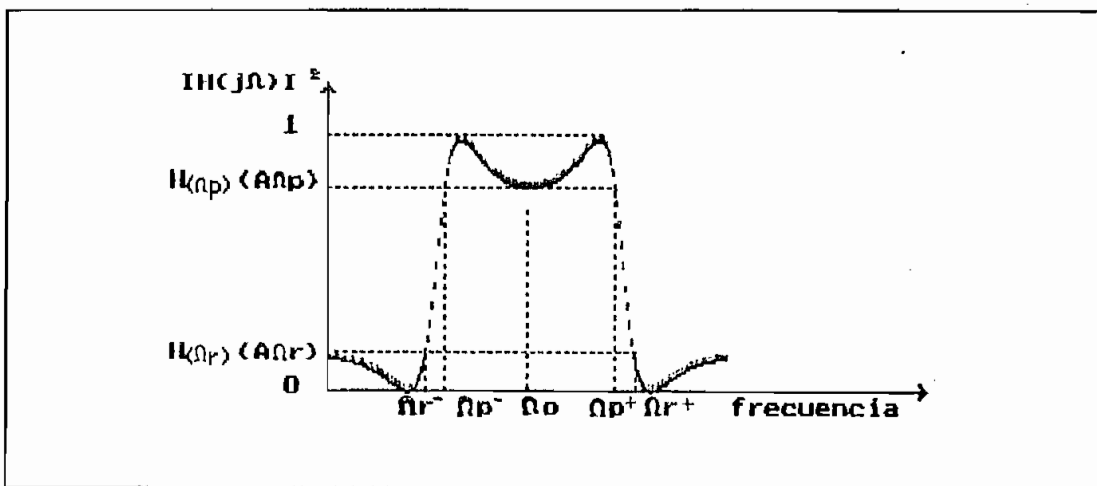


Figura 3.6 : Respuesta de frecuencia de un filtro Pasabanda de segundo orden.

Luego de la transformación de frecuencia:

$$H_B(s) = \frac{H_o}{D_B(s)} \prod_{i=1}^r \frac{s^4 + (2\Omega_o^2 + A_i W^2) s^2 + \Omega_o^4}{s^4 + B_i W s^3 + (2\Omega_o^2 + C_i W^2) s^2 + B_i W s + \Omega_o^4} \quad (3.52)$$

Con

$$D_B(s) = \begin{cases} \frac{Ws}{s^2 + Ws + \Omega_o^2} & \text{para } N \text{ impar} \\ 1 & \text{para } N \text{ par} \end{cases}$$

El filtro Pasabanda analógico puede ser digitalizado mediante tres métodos .

1.- Partiendo de la función de transferencia discreta del filtro pasabajos y usando la transformación a pasabanda:

$$Z^{-1} \quad \text{por} \quad \frac{Z^{-2} - \frac{2\alpha k Z^{-1}}{k+1} + \frac{k-1}{k+1}}{\frac{k-1}{k+1} Z^{-2} - \frac{2\alpha k}{k+1} Z^{-1} + 1} \quad (3.53)$$

Con

$$\alpha = \frac{\cos\left(\frac{\Omega_p^- + \Omega_p^+}{2}\right)}{\cos\left(\frac{\Omega_p^+ - \Omega_p^-}{2}\right)}$$

$$K = \operatorname{ctg}\left(\frac{\Omega_p^+ - \Omega_p^-}{2}\right) \operatorname{tg} \frac{\Omega_p^-}{2}$$

$\Omega_p^+$  y  $\Omega_p^-$  son las frecuencias de corte superior e inferior deseadas respectivamente.

Correspondiendo  $H(Z)$  al filtro pasabajos y  $H_B(Z)$  al filtro pasabanda, con la transformación se tiene:

$$H(Z) = \frac{H_o}{D(Z)} \prod_{i=1}^r \frac{a_i + b_i Z^{-1} + d_i Z^{-2}}{A_i + B_i Z^{-1} + D_i Z^{-2}} \quad (3.54)$$

Con

$$D(Z) = \begin{cases} \frac{E+F Z^{-1}}{e+f Z^{-1}} & \text{para } N \text{ impar} \\ 1 & \text{para } N \text{ par} \end{cases}$$

La función de transferencia del filtro pasabanda es:

$$H_B(Z) = \frac{H_o}{D_B(Z)} \prod_{i=1}^r \frac{h_{4i}Z^{-4}+h_{3i}Z^{-3}+h_{2i}Z^{-2}+h_{1i}Z^{-1}+h_{0i}}{H_{4i}Z^{-4}+H_{3i}Z^{-3}+H_{2i}Z^{-2}+H_{1i}Z^{-1}+H_{0i}} \quad (3.55)$$

Con

$$\begin{aligned} h_{4i} &= a_i m^2 + b_i m + d_i \\ h_{3i} &= -[2a_i m l + b_i l(m+1) + 2l] \\ h_{2i} &= a_i(l^2 + 2m) + b_i(m^2 + l^2 + 1) + d_i(l^2 + 2m) \\ h_{1i} &= -[2d_i l + b_i l(m+1) + 2d_i m l] \\ h_{0i} &= a_i + b_i m + d_i m^2 \\ H_{4i} &= A_i m^2 + B_i m + D_i \\ H_{3i} &= -[2A_i m l + B_i l(m+1) + 2l] \\ H_{2i} &= A_i(l^2 + 2m) + B_i(m^2 + l^2 + 1) + D_i(l^2 + 2m) \\ H_{1i} &= -[2D_i l + B_i l(m+1) + 2D_i m l] \\ H_{0i} &= A_i + B_i m + D_i m^2 \end{aligned}$$

Donde:

$$D_B(Z) = \begin{cases} \frac{(Em+F)Z^{-2} - (El+Fl)Z^{-1} + (E+Fm)}{(em+f)Z^{-2} - (el+fl)Z^{-1} + (e+fm)} & \text{para } N \text{ impar} \\ 1 & \text{para } N \text{ par} \end{cases}$$

Con:

$$l = \frac{2\alpha K}{K+1} \text{ y } m = \frac{K-1}{K+1}$$

2.- Partiendo de la función de transferencia del filtro pasabajos análogo (dominio S), se obtiene la correspondiente al filtro pasabanda análogo, y esta aplicando alguna aproximación hacerla digital.

Usando la transformada bilineal y siendo  $m=2/T$ , se obtiene  $H_B(Z)$

$$H_B(Z) = \frac{H_o}{D_B(Z)} \prod_{i=1}^r \frac{h_{4i}Z^{-4} + h_{3i}Z^{-3} + h_{2i}Z^{-2} + h_{1i}Z^{-1} + h_{0i}}{H_{4i}Z^{-4} + H_{3i}Z^{-3} + H_{2i}Z^{-2} + H_{1i}Z^{-1} + H_{0i}} \quad (3.56)$$

Donde

$$D_B(Z) = \begin{cases} \frac{Wm(1-Z^{-2})}{(m^2+Wm\sigma+\Omega_o^2) + 2(\Omega_o^2-m^2)Z^{-1} + (\Omega_o^2+m^2-Wm\sigma)Z^{-2}} & \text{para } N \text{ impar} \\ 1 & \text{para } N \text{ par} \end{cases}$$

Con:

$$\begin{aligned} h_{4i} &= h_{0i} = m^4 + m^2(2\Omega_o^2 + A_i W^2) + \Omega_o^4 \\ h_{3i} &= h_{1i} = 4(\Omega_o^4 - m^4) \\ h_{2i} &= 6m^4 + 6\Omega_o^4 - 2m^2(2\Omega_o^2 + A_i W^2) \\ H_{4i} &= m^4 + m^2(2\Omega_o^2 + C_i W^2) + \Omega_o^4 - B_i Wm^3 - \Omega_o^2 B_i Wm \\ H_{3i} &= 2B_i Wm^3 - 2\Omega_o^2 B_i Wm + 4\Omega_o^4 - 4m^4 \\ H_{2i} &= 6m^4 + 6\Omega_o^4 - 2m^2(2\Omega_o^2 + C_i W^2) \\ H_{1i} &= 2B_i Wm\Omega_o^2 - 2B_i Wm^3 + 4\Omega_o^4 - 4m^4 \\ H_{0i} &= m^4 + m^2(2\Omega_o^2 + C_i W^2) + \Omega_o^4 + B_i Wm^3 + \Omega_o^2 B_i Wm \end{aligned}$$

3.- Un método de discretización directo para un filtro pasabanda da la transformación.

$$S \rightarrow \frac{Z^2 - 2Z \cos \Omega_o T + 1}{Z^2 - 1} \quad (3.57)$$

Para esta aproximación el eje imaginario del plano S pasa al arco superior e inferior del plano Z y el origen pasa a  $e^{\pm j\Omega_o T}$ .

Al igual que la transformada bilineal se necesita hacer una adaptación de a las frecuencias críticas para evitar corrimientos de frecuencia.

$$\Omega'_A = \frac{\cos \Omega_o T - \cos \Omega_A T}{\text{sen } \Omega_A T} \quad (3.58)$$

donde  $\Omega_A$  representa a  $\Omega_r^-, \Omega_r^+, \Omega_p^-, \Omega_p^+$

$$\cos \Omega_o T = \frac{\cos \frac{1}{2} \left[ \frac{\Omega_p^+}{fs} + \frac{\Omega_p^-}{fs} \right]}{\cos \frac{1}{2} \left[ \frac{\Omega_p^+}{fs} - \frac{\Omega_p^-}{fs} \right]}$$

El problema se reduce a calcular un filtro pasabajos dado las frecuencias y las atenuaciones en  $\Omega_p^-$  y  $\Omega_r^-$ .

Discretizando y siendo  $m=2\cos(\Omega_o T)$ , se obtiene .

$$H_B(Z) = \frac{H_o}{D(Z)_B} \prod_{i=1}^r \frac{h_{4i}Z^{-4} + h_{3i}Z^{-3} + h_{2i}Z^{-2} + h_{1i}Z^{-1} + h_{oi}}{H_{4i}Z^{-4} + H_{3i}Z^{-3} + H_{2i}Z^{-2} + H_{1i}Z^{-1} + H_{oi}} \quad (3.59)$$

Con:

$$\begin{aligned} h_{4i} &= 1 + A_i \\ h_{3i} &= -2m \\ h_{2i} &= 2 + m^2 - 2A_i \\ h_{1i} &= h_{3i} = -2m \\ h_{oi} &= 1 + A_i \\ H_{4i} &= 1 + C_i + B_i \\ H_{3i} &= m(B_i - 2) \\ H_{2i} &= (2 + m^2 - 2C_i) \\ H_{1i} &= -m(2 + B_i) \\ H_{oi} &= 1 + B_i + C_i \end{aligned}$$

Donde:

$$D_B(Z) = \begin{cases} \frac{(1+\sigma)Z^{-2} - mZ^{-1} + (1-\sigma)}{1-Z^{-2}} & \text{para } N \text{ impar} \\ 1 & \text{para } N \text{ par} \end{cases}$$

Con cualquier método de discretización, la ecuación en diferencias se obtiene del mismo modo que para los filtros pasabajos.

$$Y_I[n] = \frac{1}{H_0} (H_1 Y[n-1] + H_2 Y[n-2] + H_3 Y[n-3] + H_4 Y[n-4] + h_0 X_I[n] + h_1 X_I[n-1] + h_2 X_I[n-2] + h_3 X_I[n-3] + h_4 X_I[n-4]) \quad (3.60)$$

Mientras que para  $D(z)$ , tomando el caso dos de discretización, se tiene:

$$Y[n] = \frac{1}{(m^2 + Wm\sigma + \Omega_o^2)} [-2(\Omega_o^2 - m^2) Y[n-1] - (\Omega_o^2 + m^2 - Wm\sigma) Y[n-2] + WmX[n] + WmX[n-2]] \quad (3.61)$$

### 3.3.2.-DISCRETIZACION DEL COMPARADOR DE FASE

Se analiza el comparador de Fase mencionado en el capítulo I, donde se encuentra también su análisis matemático, el que puede estar representado por el diagrama de bloques:

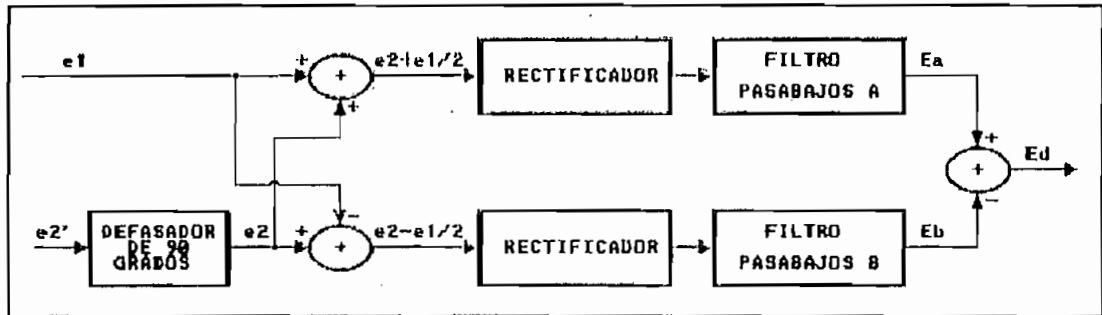


Figura 3.7: Diagrama de bloques de un Comparador de fase

El filtro RC se construye en base a un filtro pasabajos de primer orden .

### 3.3.3.- DISCRETIZACION DEL OSCILADOR CONTROLADO POR VOLTAJE

Un VCO en forma general puede ser representado por el

diagrama de bloques.

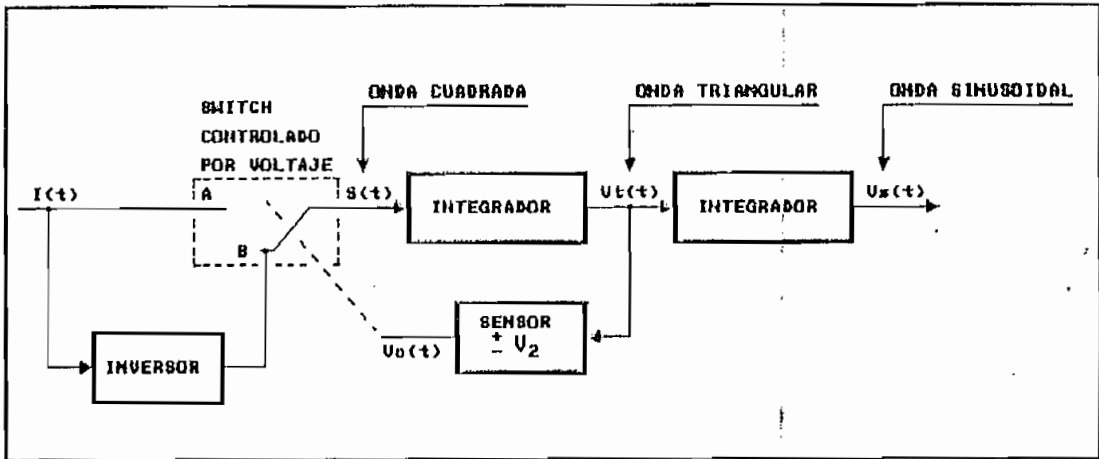
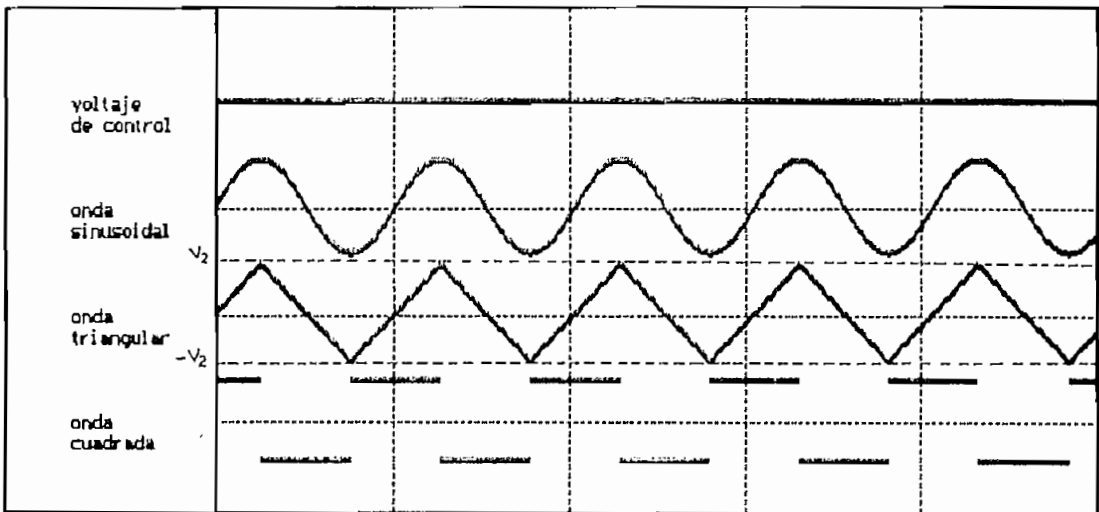


Figura 3.8: Diagrama de bloques de un Oscilador Controlado por Voltaje (V.C.O.)

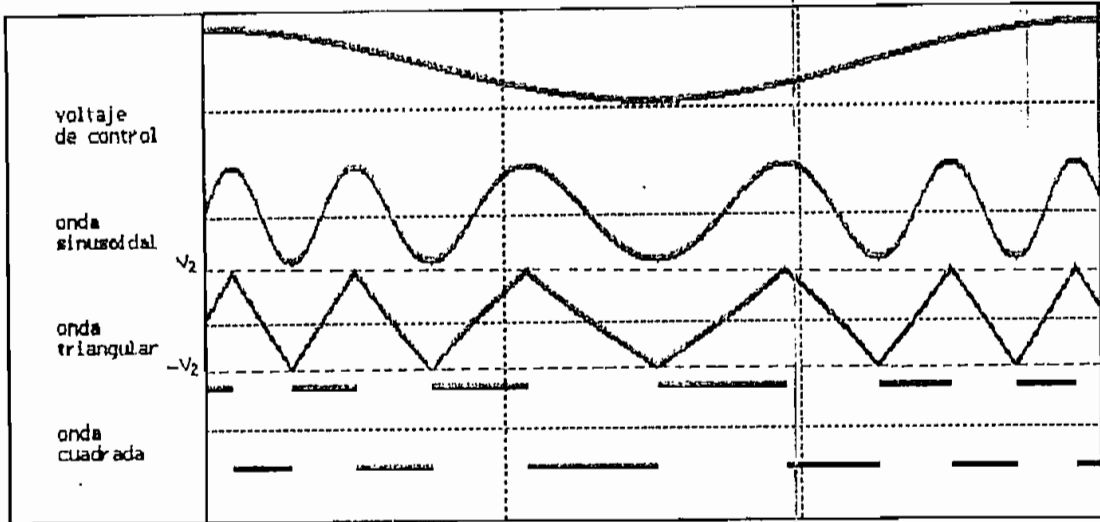
Con  $I(t)$  como señal de control del VCO, de acuerdo a su valor el bloque puede responder:

$I(t)$  constante



Cuando  $I(t)$  no es constante el VCO oscila a una mayor frecuencia mientras mayor sea el valor de  $I(t)$ , ya que al incrementar esta señal, la integración es mas rápida de esta forma,  $V_x(t)$  iguala mas pronto a  $\pm V_2$ .





Mientras la señal cuadrada  $s(t)$  se mantiene en el nivel alto, la onda triangular  $v_t(t)$  crece monótonamente.

$$v_t(t) = -V_2 + K_1 \int I(t) dt \quad (3.62)$$

Cuando  $s(t)$  se mantiene en el nivel bajo,  $v_t(t)$  decrece monótonamente y la función que la rige es:

$$v_t(t) = V_2 - K_1 \int I(t) dt \quad (3.63)$$

Pasando  $v_t(t)$  por un dispositivo no lineal, se obtiene la onda sinusoidal  $v_s(t)$ .

$$v_s(t) = K \operatorname{sen}\left(\frac{\pi v_t(t)}{2 V_2}\right) \quad (3.64)$$

Siendo  $f_o$  la frecuencia central de oscilación del VCO, el período de las ondas es:

$$T = \frac{1}{f_o} = \frac{4 V_2}{K_1}$$

## CAPITULO IV

### SIMULACION DE LOS METODOS DE RECUPERACION DE SINCRONISMO Y LAZO PLL

TEMA	PAGINA
4.1 Introducción	67
4.2 Algoritmos	
4.2.1. Filtros	68
4.2.2. Defasador de 90 grados	70
4.2.3. Comparador de fase, Duplicador de frecuencia y Oscilador controlado por voltaje	71

## CAPITULO IV

SIMULACIÓN DE LOS METODOS DE RECUPERACIÓN  
DE SINCRONISMO Y LAZO PLL

## 4.1 INTRODUCCION

Tanto los bucles de recuperación de sincronismo como el lazo PLL, han sido analizados como sistema. Se ha conseguido modelos matemáticos de cada componente que conforman estos sistemas, de tal forma que, ubicando estos modelos como indican los diagramas de bloques de los bucles dados en el capítulo II, se puede simular el comportamiento de cada uno de los bucles de recuperación de sincronismo. Igualmente se procede con el PLL cuyo diagrama de bloques se da en el capítulo I.

Los modelos matemáticos cumplen con la teoría y los diagramas de bloques de los elementos reales analizados en los capítulos I y III. En este capítulo se describen los procesos que se han seguido para obtener y simular los modelos mencionados y se presentan diagramas de flujo que resumen dichos procesos.

El código fuente de los programas por facilidad para el lector se presenta en el ANEXO C, pero se mencionan aquí ciertas características de los programas, que están vinculadas con la estructuración y simulación de los modelos matemáticos de los elementos básicos de los bucles en estudio.

## 4.2 ALGORITMOS

### 4.2.1. FILTROS

ESTRUCTURACION.- Para obtener los modelos matemáticos de los filtros, se ha procedido de la siguiente manera:

1.- Se analizan las características de los filtros pasabajos y pasabanda reales, que son componentes de los diferentes bucles en los capítulos I y III respectivamente.

2.- Se obtienen las funciones matemáticas que representan estos elementos. (1.10) para filtro Pasabajos de Butterworth, (1.26) para filtro Pasabajos Elíptico y (3.52) para filtro pasabanda Elíptico.

3.- Como en un computador el procesamiento es discreto, se hace necesario discretizar las funciones matemáticas obtenidas, para esto, se utiliza la Transformación Bilineal debido a sus características que son mejores que las otras transformaciones estudiadas (Capítulo III), ya que, evita el cruce de espectros lo que es un problema de la Transformación Invariante al Impulso, y la aproximación Trapezoidal usada en la Transformación Bilineal es mejor que la aproximación por diferencias usada en la Solución numérica de las ecuaciones diferenciales. (3.35) para filtro Pasabajos de Butterworth, (3.39) para filtro Pasabajos Elíptico y (3.56) para filtro pasabanda Elíptico..

4.- Una vez obtenida la función discreta (Dominio Z), el siguiente paso es determinar las ecuaciones en diferencias, que es lo que se simula. (3.36) y (3.37) para filtro Pasabajos de Butterworth, (3.40) y (3.41) para filtro Pasabajos Elíptico y finalmente (3.60) y (3.61) para filtro

pasabanda Elíptico.

En el segundo paso, para determinar las constantes de las funciones de transferencia citadas, se realiza en forma secuencial el cálculo de las relaciones matemáticas, dadas en los capítulos I y III para estos fines. En forma resumida el procedimiento seguido, se representa en los diagramas de flujo de ESTRUCTURACION de los filtros, no se presenta el diagrama de flujo del filtro pasabanda ya que es el mismo que el filtro pasabajos Elíptico mas la transformación de frecuencia dada en el capítulo III. En los programas, las constantes determinadas se guardan en arreglos unidimensionales de tipo global.

**SIMULACION.**- En la Simulación de los filtros se establece un sistema en cascada respetando las ecuaciones en Diferencias de cada célula elemental, donde el número de componentes en cascada depende del orden del filtro que se adopta para cumplir con las características de frecuencia y atenuación impuestas al mismo. El procedimiento para la simulación se da en los diagramas de flujo SIMULACION de los filtros.

Cuando el filtro se compone de varias etapas, la respuesta de una etapa es la entrada a la siguiente, para optimizar el código del programa, se utiliza arreglos bidimensionales  $[n+2][m]$ , donde  $n$  es el número de elementos en cascada que tiene el filtro y  $m$  es el número de componentes que tiene la respuesta de cada etapa,  $m=3$  para filtro pasabajos y  $m=5$  para filtro pasabanda. Con estos arreglos, no se repite el código para los cálculos de cada etapa, lo que sucedería si se toma como base los diagramas de

flujo de los filtros digitales dado en el capítulo III, sino que, se forma un solo lazo de cálculos para las  $n$  etapas.

**RESPUESTA DE FRECUENCIA** .- La respuesta de los Filtros a simular se obtiene de la función de transferencia del mismo en el Dominio de la frecuencia Discreta (Dominio  $Z$ ), reemplazando  $z$  por  $e^{-j\omega}$ . Luego de una serie de operaciones matemáticas, cada factor ya sea que proviene de uno de primero, segundo o cuarto orden, siempre se lo puede representar como la suma de una componente real y una componente imaginaria.

PARA  $N$  IMPAR

$$H(e^{j\omega}) = H_0 (R_0 + j I_0) (R_1 + j I_1) \dots (R_{\frac{(N-1)}{2}} + j I_{\frac{(N-1)}{2}})$$

PARA  $N$  PAR

$$H(e^{j\omega}) = H_0 (R_1 + j I_1) \dots (R_{\frac{N}{2}} + j I_{\frac{N}{2}})$$

Para determinar  $|H(e^{j\omega})|^2$  se debe llegar a obtener las componentes equivalentes real e imaginaria.

$$H(e^{j\omega}) = H_0 (R_{\theta\omega} + j I_{\theta\omega})$$

La determinación de  $I_{\theta\omega}$  y  $R_{\theta\omega}$  para filtro pasabajos se describen en el diagrama de flujo (RESPUESTA DE FRECUENCIA DE LOS FILTROS).

#### 4.2.2. DEFASADOR DE 90°

El Defasador se puede considerar como un filtro pasabanda que tiene la propiedad adicional de desplazar  $90^\circ$  a todas las componentes de frecuencia de una señal que estén dentro del rango de frecuencia del filtro. De este modo, el cálculo de las constantes de la función matemática que lo representa (3.51), es similar a la de un filtro elíptico, este proceso matemático se describe en el diagrama de flujo ESTRUCTURACION DE DEFASADOR DE  $90^\circ$ . En los programas, igual que las constantes de los filtros, sus constantes son almacenadas en arreglos unidimensionales de tipo global.

Para la simulación igual que los filtros se llega a la ecuación en diferencias a partir de (3.51). Como se ha limitado el orden del elemento a 5, el arreglo bidimensional hecho en los filtros no es necesario, por lo que, la simulación se hace de acuerdo a los diagramas de flujo del defasador dados en el capítulo III, de acuerdo al orden del mismo.

#### 4.2.3 COMPARADOR DE FASE, DUPLICADOR DE FRECUENCIA Y OSCILADOR CONTROLADO POR VOLTAJE

Se obtiene el modelo matemático del comparador de fase a partir del diagrama de bloques dado en el capítulo III, el que representa al circuito dado en el capítulo I, cuyas relaciones matemáticas se dan en el mismo capítulo.

Una representación del proceso de cálculo en la simulación se da en el diagrama de flujo SIMULACION del comparador de fase.

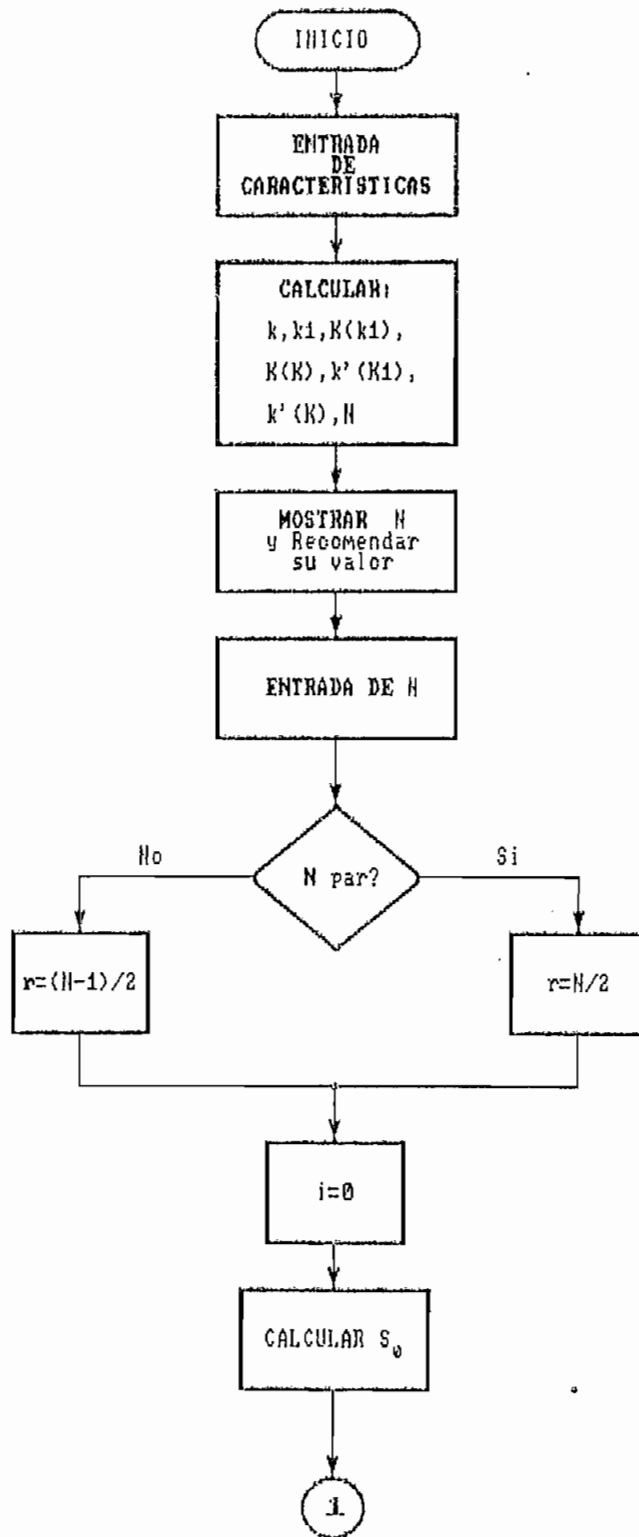
El Duplicador de frecuencia se forma de acuerdo al diagrama de bloques dado en el capítulo III, donde el elemento principal es el filtro pasabanda, que ha sido analizado en FILTROS.

El modelo del oscilador controlado por voltaje se ha conseguido en base al diagrama de bloques y las relaciones matemáticas dado en el capítulo III, que cumplen con la teoría del capítulo I.

Una representación del proceso de cálculo en la simulación se da en el diagrama de flujo SIMULACION del oscilador controlado por voltaje.



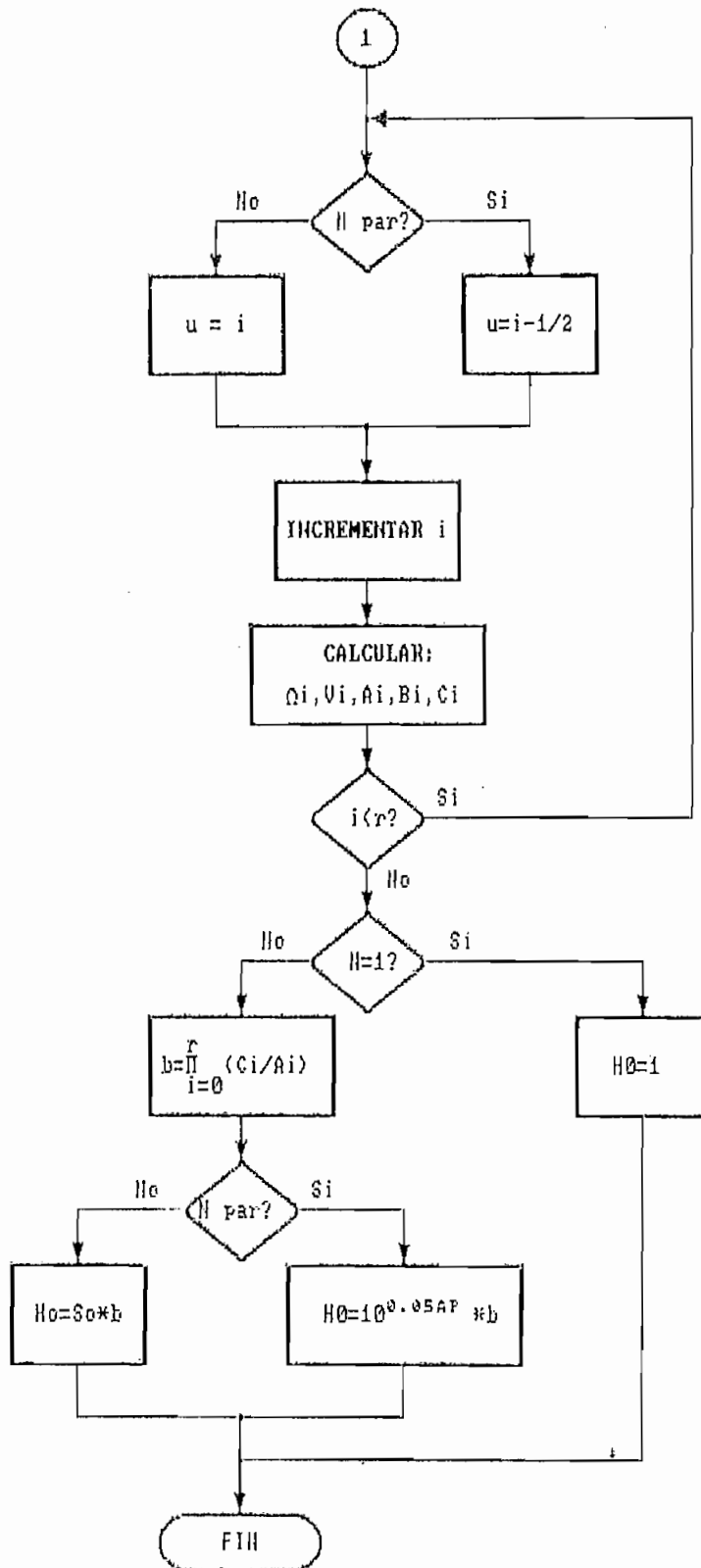
## DIAGRAMA DE FLUJO DE LA ESTRUCTURACION DEL FILTRO PASABANDAS ELIPTICO



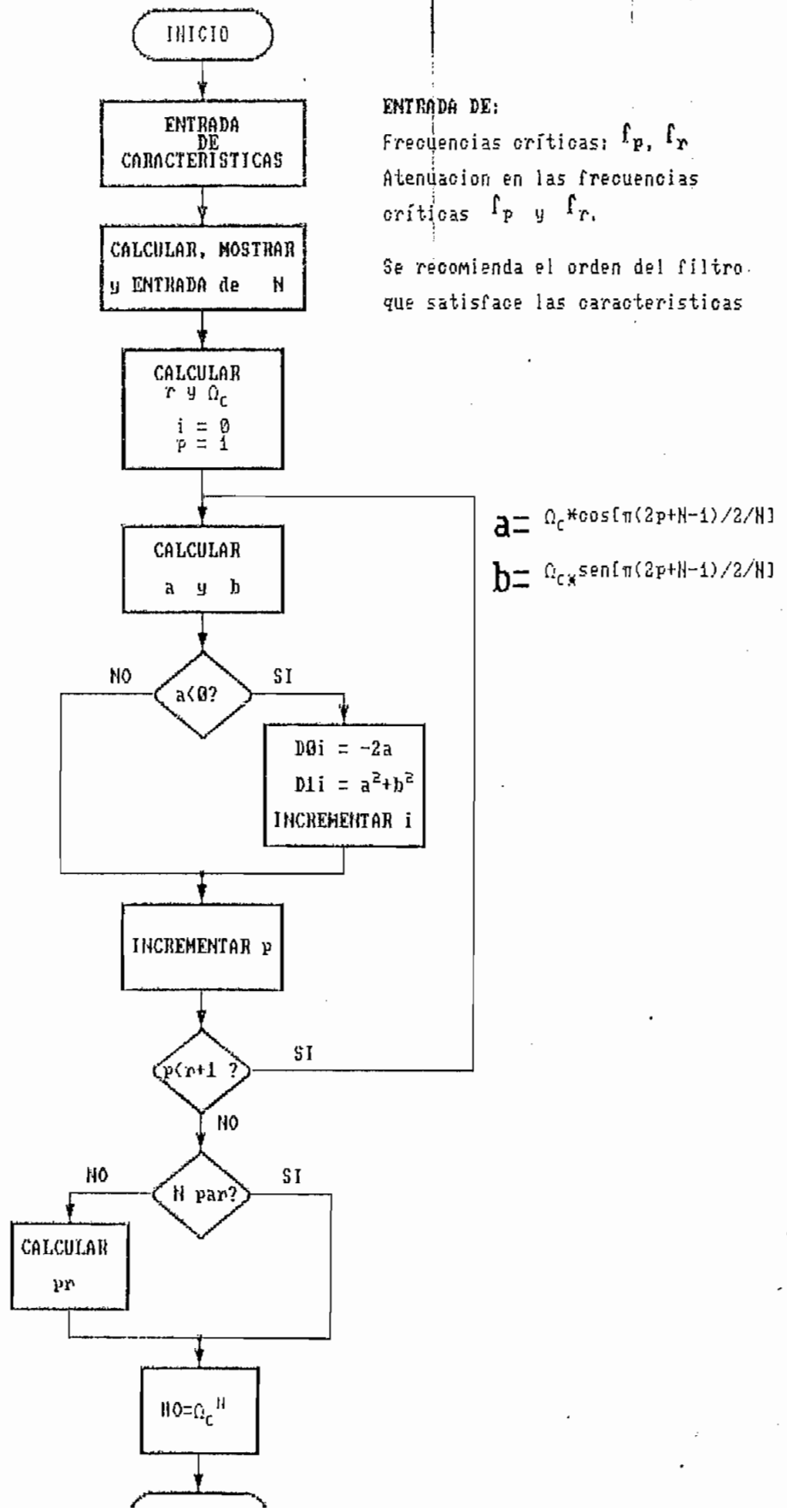
ENTRADA DE:

Frecuencias críticas:  $f_p$ ,  $f_r$ Atenuación en las frecuencias  
críticas  $f_p$  y  $f_r$ .

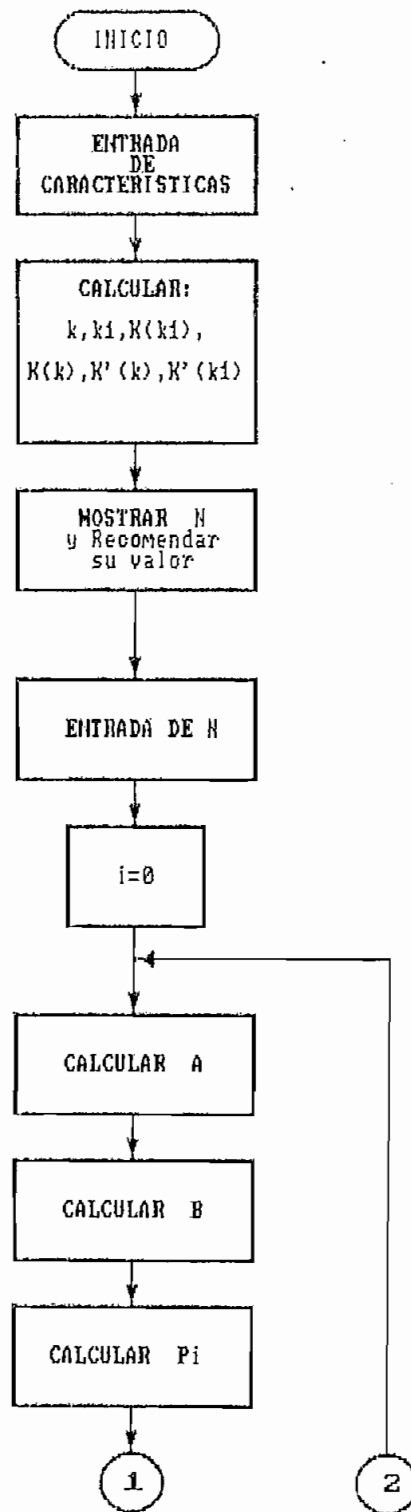
DIAGRAMA DE FLUJO DE LA ESTRUCTURACION DEL FILTRO PASABAJOS ELIPTICO  
(CONTINUACION)



## DIAGRAMA DE FLUJO DE LA ESTRUCTURACION DEL FILTRO PASABAJOS DE BUTTERWORTH



## DIAGRAMA DE FLUJO DE LA ESTRUCTURACION DEL DEPASADOR



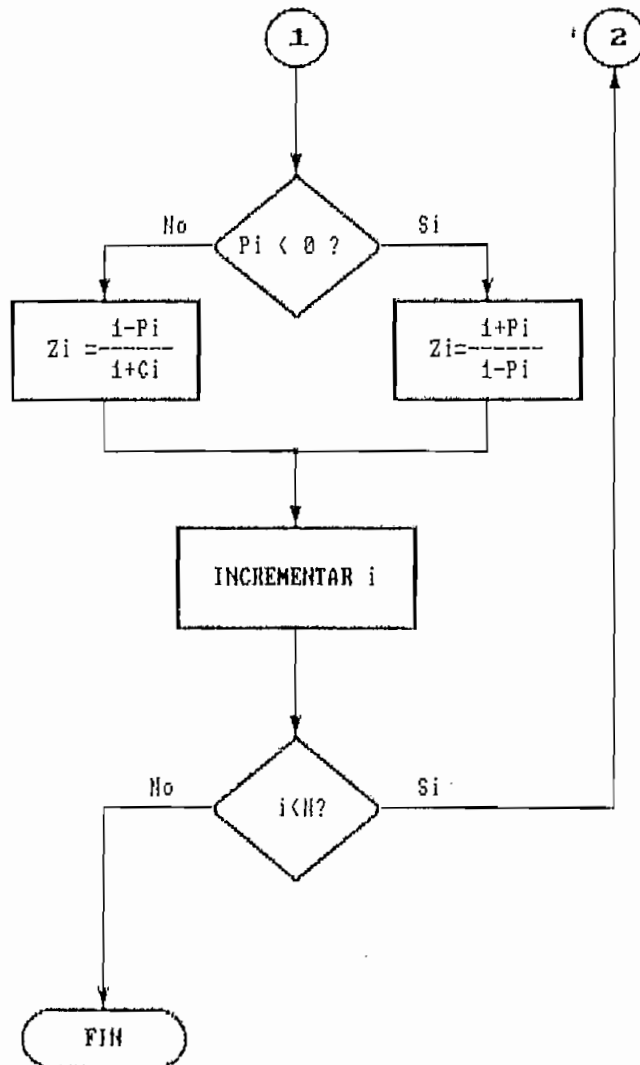
ENTRADA DE:

Frecuencias críticas:  $f_1$  y  $f_2$ Atenuación en las frecuencias  
críticas  $f_1$  y  $f_2$ .Factor de error  $\epsilon$ 

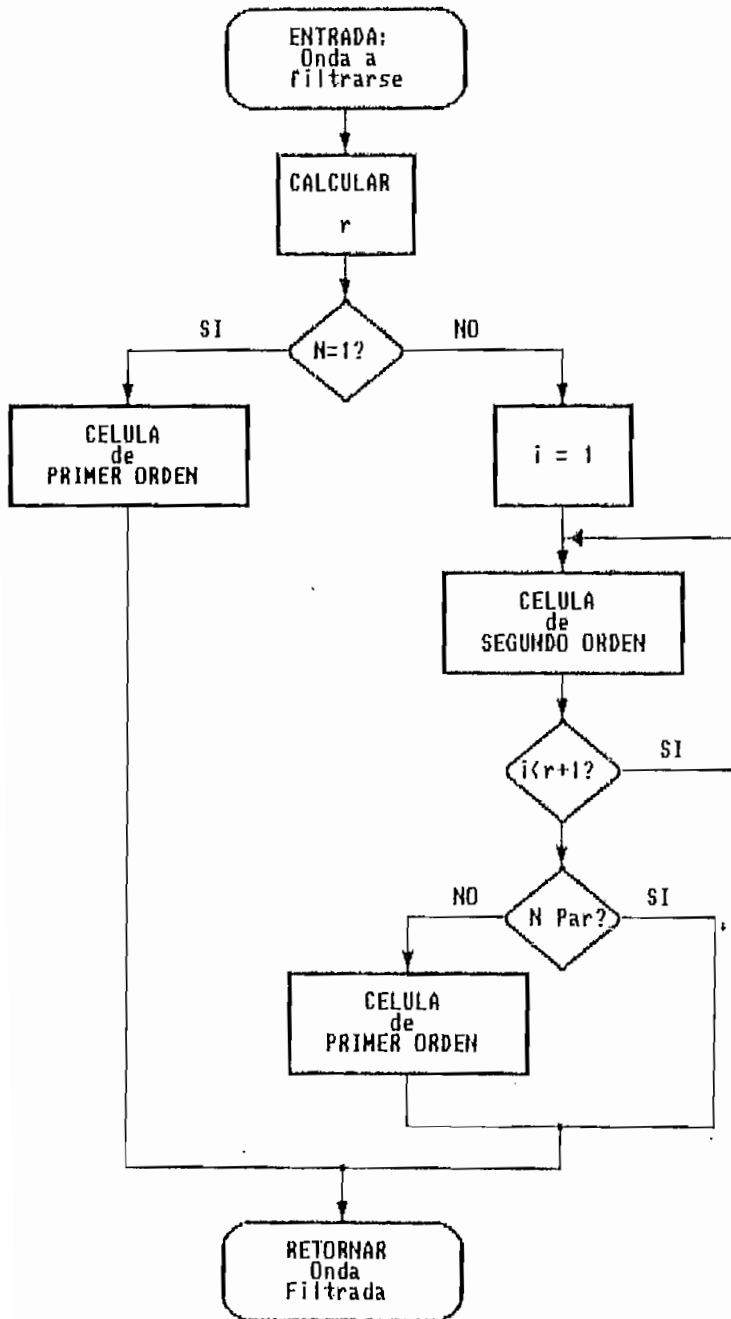
$$A = \operatorname{sn} \left( \frac{(4i+1)K'(k)}{2N}, (1-k^2)^{1/2} \right)$$

$$B = \operatorname{cn} \left( \frac{(4i+1)K'(k)}{2N}, (1-k^2)^{1/2} \right)$$

$$P_i = -\operatorname{tang}(\Omega_1 T/2) \frac{A}{B}$$

DIAGRAMA DE FLUJO DE LA ESTRUCTURACION DEL DEFASADOR  
(CONTINUACION)

## DIAGRAMA DE FLUJO DE SIMULACION DEL FILTRO PASABAJOS



## DIAGRAMA DE FLUJO DE LA SIMULACION DE FILTROS PASABANDA

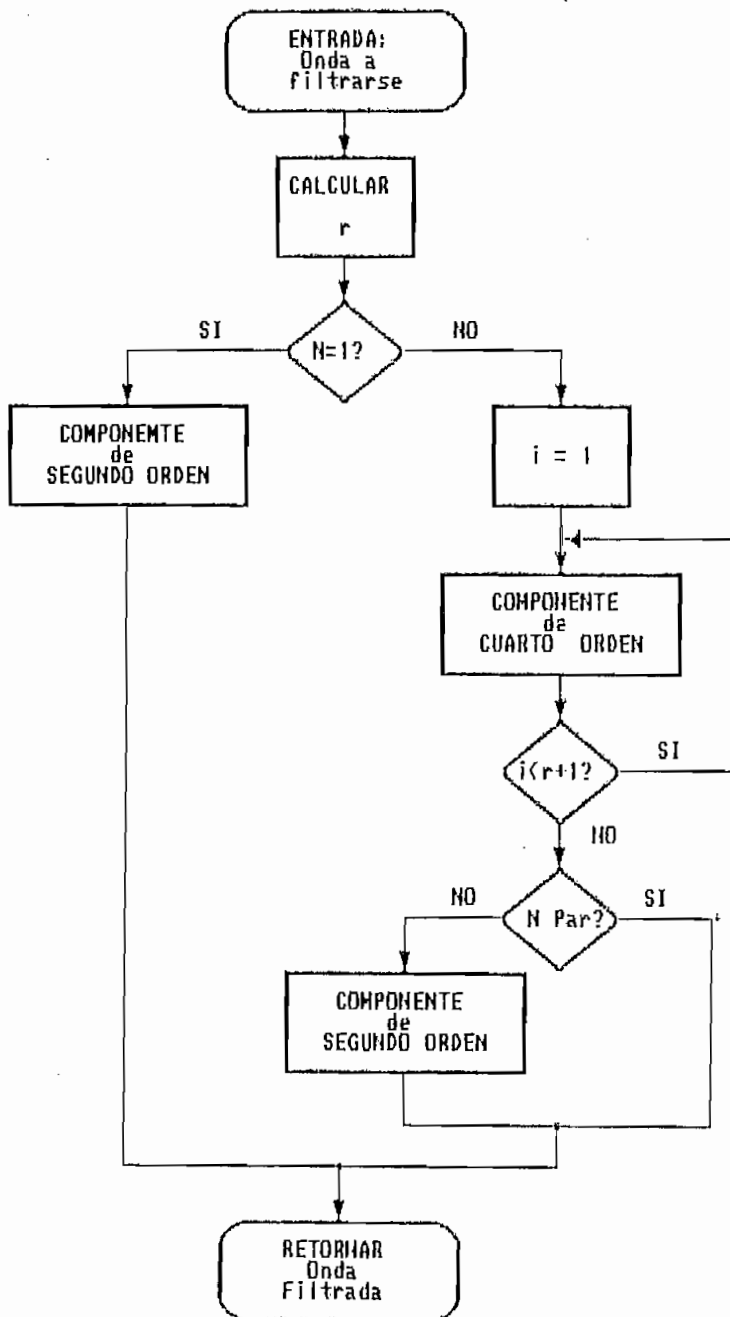
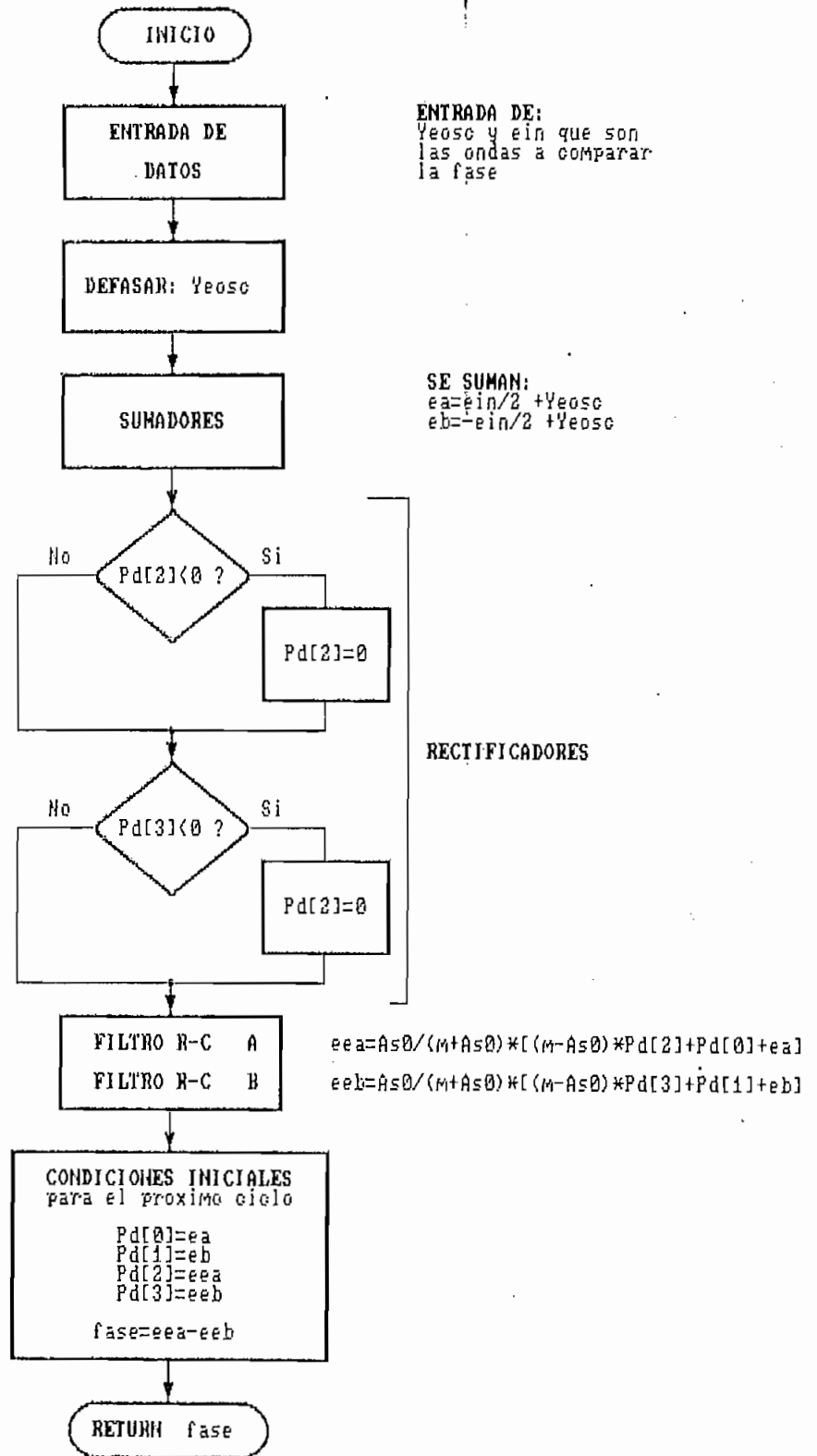
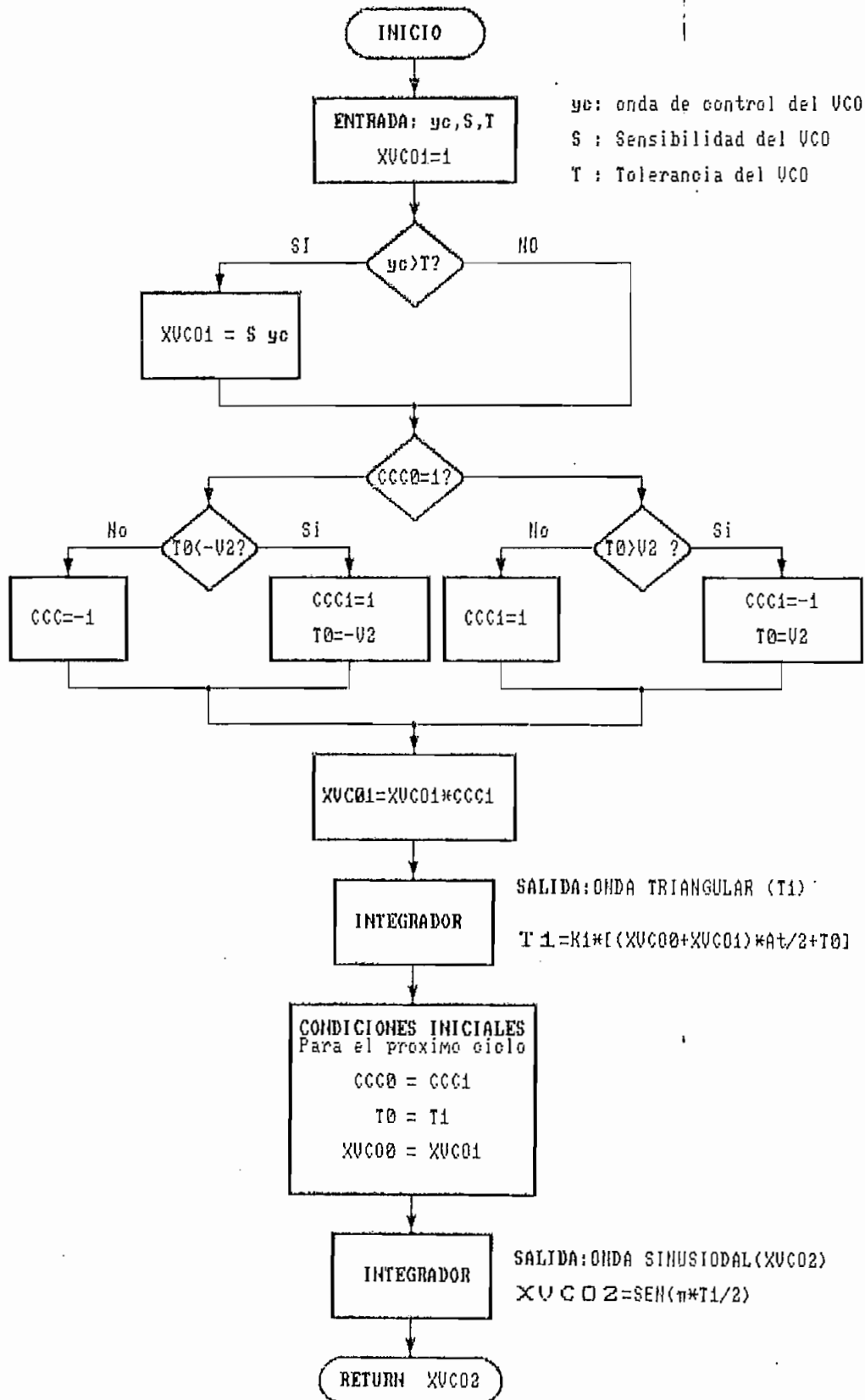


DIAGRAMA DE FLUJO DE SIMULACION DEL COMPARADOR DE FASE

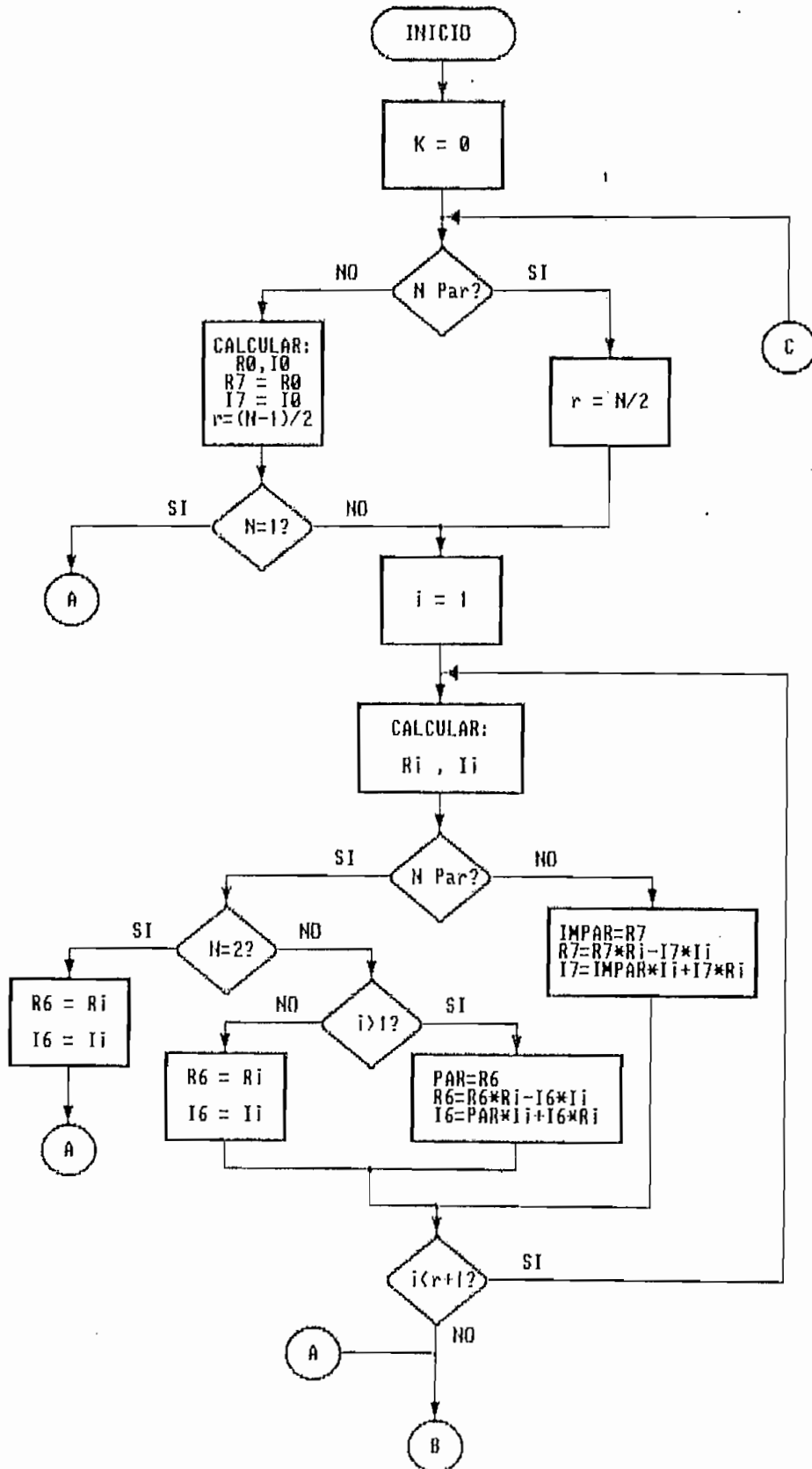




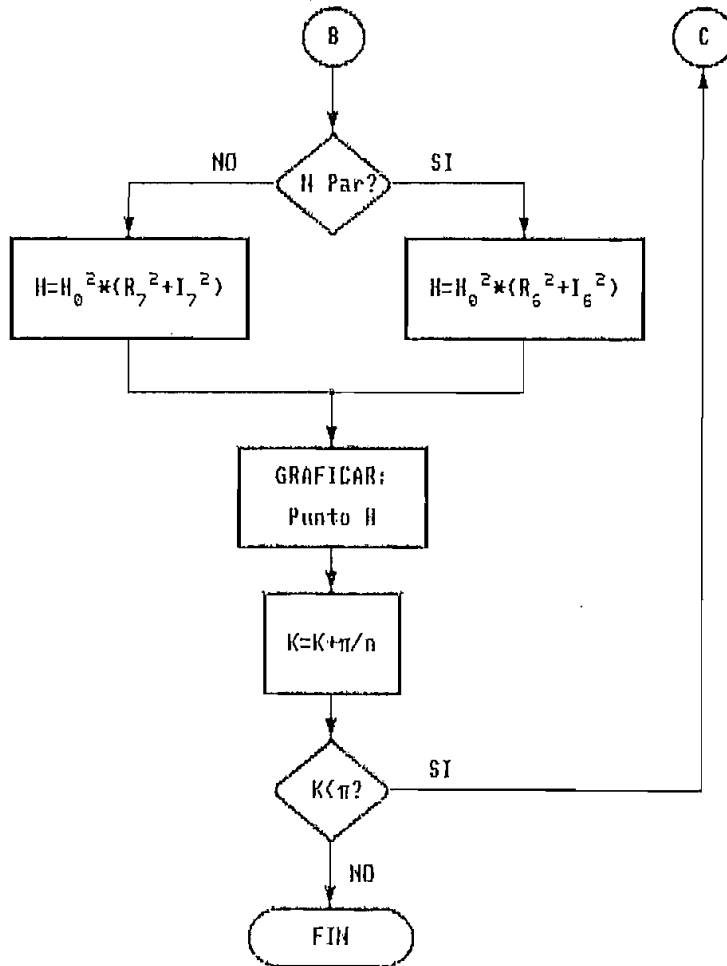
## DIAGRAMA DE FLUJO DE SIMULACION DEL OSCILADOR CONTROLADO POR VOLTAJE (V.C.O.)



## RESPUESTA DE FRECUENCIA DE LOS FILTROS PASAJOS O PASABANDA



RESPUESTA DE FRECUENCIA DE LOS FILTROS PASABAJOS O PASABANDA  
(CONTINUACION)



## CAPITULO V

### PRESENTACION DE RESULTADOS

TEMA	PAGINA
5.1. Bucle con multiplicación	88
5.2. Bucle de Costas General	89
5.3. Bucle con Remodulación por el Oscilador Local	91
5.4. Bucle con remodulación por la señal recibida	92
5.5. Bucle de Costas con detección dentro de un brazo	94
5.6. Bucle de Costas Numérico	94
5.7. Bucle de Costas con Remodulación para QPSK	96

## CAPITULO V

## PRESENTACION DE RESULTADOS

Para observar características, ventajas y desventajas de los distintos bucles se hace un estudio de respuestas de los mismos, a iguales condiciones de entrada e igual estructuración de los componentes comunes a diferentes bucles como VCO's, filtros, etc.

Condicionantes principales para una buena respuesta de los bucles son: Sensibilidad del VCO (frecuencias inferior y superior de oscilación), Rapidez de Integración de los filtros y Precisión del Defasador de  $90^\circ$  en los bucles que éste interviene.

## CONDICIONES DE PRUEBA:

## CARACTERISTICAS DE LA ONDA DE ENTRADA AL BUCLE

Tren de bits :	1 1 0 1 1
Ritmo del tren de bits :	1200 [bps]
Frecuencia de portadora :	2400 [Hertz]
Angulo de error inicial entre la onda recibida y la creada por VCO:	(A)0 (B)60 (C)120 (D)240 (E)300.

## ESTRUCTURA DEL FILTRO PASABAJOS

## Filtro Elíptico (Ea):

Frecuencia de corte (fp):	500 [Hertz]
Frecuencia de supresión (fr):	1500 [Hertz]
Atenuación en fp:	1 [dB]
Atenuación en fr:	8 [dB]
Orden del Filtro:	1

## Filtro Elíptico (Eb):

Frecuencia de corte (fp):	1000 [Hertz]
Frecuencia de supresión (fr):	1500 [Hertz]
Atenuación en fp:	1 [dB]
Atenuación en fr:	5 [dB]
Orden del Filtro:	2

## Filtro Elíptico (Ec):

Frecuencia de corte (fp):	1000 [Hertz]
Frecuencia de supresión (fr):	1500 [Hertz]

Atenuación en fp:	1	[dB]
Atenuación en fr:	45	[dB]
Orden del Filtro:	4	

**Filtro Elíptico (Ed):**

Frecuencia de corte (fp):	2500	[Hertz]
Frecuencia de supresión (fr):	5000	[Hertz]
Atenuación en fp:	1	[dB]
Atenuación en fr:	5	[dB]
Orden del Filtro:	1	

**Filtro Elíptico (Ee):**

Frecuencia de corte (fp):	400	[Hertz]
Frecuencia de supresión (fr):	1000	[Hertz]
Atenuación en fp:	1	[dB]
Atenuación en fr:	5	[dB]
Orden del Filtro:	1	

**Filtro de Butterworth (Ba):**

Frecuencia de corte (fp):	500	[Hertz]
Frecuencia de supresión (fr):	1500	[Hertz]
Atenuación en fp:	1	[dB]
Atenuación en fr:	8	[dB]
Orden del Filtro:	2	

**Filtro de Butterworth (Bb):**

Frecuencia de corte (fp):	1000	[Hertz]
Frecuencia de supresión (fr):	1500	[Hertz]
Atenuación en fp:	1	[dB]
Atenuación en fr:	5	[dB]
Orden del Filtro:	3	

**Filtro de Butterworth (Be):**

Frecuencia de corte (fp):	400	[Hertz]
Frecuencia de supresión (fr):	1000	[Hertz]
Atenuación en fp:	1	[dB]
Atenuación en fr:	5	[dB]
Orden del Filtro:	1	

**ESTRUCTURA DEL OSCILADOR CONTROLADO POR VOLTAJE****Condición 1 (X1):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	20	[Hz/grado]
Tolerancia :	10	[%]

**Condición 2 (X2):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	30	[Hz/grado]
Tolerancia :	10	[%]

**Condición 3 (X3):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	20	[Hz/grado]
Tolerancia :	40	[%]

**Condición 4 (X4):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	20	[Hz/grado]
Tolerancia :	40	[%]

**Condición 5 (X5):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	10	[Hz/grado]
Tolerancia :	40	[%]

**Condición 6 (X6):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	5	[Hz/grado]
Tolerancia :	40	[%]

**Condición 7 (X7):**

Frecuencia central de oscilación :	2400	[Hertz]
Sensibilidad :	5	[Hz/grado]
Tolerancia :	20	[%]

**ESTRUCTURA DEL DEFASADOR DE 90 GRADOS****Condición 1 (D1):**

Frecuencia inferior :	1000	[Hertz]
Frecuencia superior :	4000	[Hertz]
Error de fase máximo:	1	[grado]
Orden del filtro :	5	

**ESTRUCTURA DEL DUPLICADOR DE FRECUENCIA****Condición 1 (B1):**

Frecuencia central( $f_0$ ):	4800	[Hertz]
Frecuencia de corte ( $f_p$ ):	4000	[Hertz]
Frecuencia de supresión ( $f_r$ ):	3500	[Hertz]
Atenuación en $f_p$ :	1	[dB]
Atenuación en $f_r$ :	30	[dB]
Orden del Filtro:	3	

**CODIGOS PARA CONDICIONES DE PRUEBA EN LOS GRAFICOS**

Bajo cada gráfico de respuestas de los bucles, dados en este capítulo se muestran códigos que representan las condiciones de prueba. EJEMPLO: FIGURA B1.1:  $E_c$   $E_c$  X1 (A)

$E_c$  Representa un filtro elíptico ( $E_a$ ) [VER página 84:ESTRUCTURA DEL FILTRO PASABAJOS]

X1 Representa la condición 1 del VCO [VER página

85:ESTRUCTURA DEL OSCILADOR CONTROLADO POR VOLTAJE]

(A) Representa el error inicial de fase [VER página

## 84: CARACTERISTICAS DE LA ONDA DE ENTRADA AL BUCLE]

## RECONOCIMIENTO DE ONDAS SOBREPUESTAS EN LOS GRAFICOS

ONDAS 1 y 2: La onda recibida(1) en todos los gráficos excepto para el numeral 5.7 (Bucle de costas para QPSK), tiene un ángulo inicial de cero grados; al ser una función cosenoidal, tomando voltajes normalizados empieza siempre en 1 o -1. Además como las secuencias binarias de prueba tienen cambios de niveles lógicos, la onda(1) es la que tiene cambios bruscos de fase en el inicio de cada intervalo que corresponde a un cambio de nivel lógico de la señal de banda base. De este modo se puede reconocer la onda (1) para QPSK.

ONDAS 3,4 y 5: La onda (5) es la salida del ecualizador, por lo tanto esta formada por líneas rectas.

La onda (3) es la salida del multiplicador analógico del demodulador, por tanto tiene la componente de doble frecuencia bien definida [mejor que la onda (4)]; es decir, se nota claramente que la onda (3) es de menor amplitud y de doble frecuencia que la onda (1). Mientras que la onda (4) es la salida del filtro pasabajos, que tiene como entrada la onda (3); consecuentemente en la onda (4) se nota mas claramente que en la onda (3) la componente continua, además es mas atenuada que la onda mencionada.

ONDAS 6,9 y 0: En los bucles de costas, estas ondas pueden ser reconocidas mediante el mismo criterio que para las ondas 3, 4 y 5; tratando las ondas 6, 9 y 0 semejantes a 3, 4 y 5 respectivamente



### 5.1. BUCLE CON MULTIPLICACION DE FRECUENCIA

1.- Considerando los gráficos B1.1, B1.2 y B1.3, en los que la única diferencia está en los filtros pasabajos, se nota que en los dos primeros al ser las imperfecciones del filtrado despreciable, no se dan errores en las respuestas, mientras que en B1.3 existen picos en el filtrado que provocan errores en las respuestas del bucle. Por lo que se preferirá un buen filtrado aunque esto involucre un retardo mayor de la señal binaria recuperada con respecto a la modulada recibida.

2.- Observando los gráficos B1.4, B1.5 y B1.6, en los que la diferencia se da en la Sensibilidad del VCO. En B1.4, el cambio de la portadora recuperada es tan brusco que produce variaciones muy grandes en el salida del duplicador de frecuencia de la señal del VCO, como es un sistema realimentado, el bucle tiende a ser inestable; esto es, en intervalos se recupera el sincronismo y en otros se pierde, lo que no debe suceder ya que una vez recuperado el sincronismo por primera vez, éste debe mantenerse mientras exista la transmisión. Mientras se disminuye la Sensibilidad del VCO, el bucle gana en estabilidad, es así que en B1.6 las respuestas del Bucle son aceptables.

3.- Considerando las curvas de B1.6, B1.7, B1.8 y B1.9, que son obtenidas del mismo bucle con diferentes errores iniciales de fase, se nota que la posibilidad de que se presente un pulso en la señal de control del VCO, que tienda a recuperar el sincronismo como en B1.6 es pobre. Se debe a que la Tolerancia del VCO es alta (40%).

4.- Se considera un bucle con las características del VCO, necesarias para que la tendencia a recuperar el sincronismo se de para cualquier ángulo de error de fase inicial; para esto, con respecto al VCO anterior (X6) se ha disminuido la tolerancia del mismo. En las curvas de B1.10 y B1.12 donde el error de fase inicial es menor que  $90^\circ$  ya sea en retraso o adelanto, la recuperación de sincronismo es el deseado, mientras que en B1.11, al ser el error de fase inicial mayor que  $90^\circ$  se recupera el sincronismo con un error del 100% en la señal binaria detectada.

## 5.2.BUCLE DE COSTAS GENERAL

1.- Considerando los gráficos B2.1 y B2.2 en los que la única diferencia está en los filtros pasabajos, se nota que el retraso existente entre la señal de entrada y la binaria recuperada depende del retardo de bucle producido por el filtrado, así como el orden de los filtros de la figura B2.1 es mayor, el retardo es también mayor.

2.- Observando los gráficos B2.3 y B2.4, se nota que en la figura B2.4 la recuperación de sincronismo es mas rápida ya que la sensibilidad del VCO es mayor. (La condición X2 cambia 30 [Hz] por grado error de fase mientras que la condición X1 cambia solo 20 [Hz] por grado de error de fase).

3.- Tanto en las curvas de B2.5 como de B2.6, la señal creada por el VCO se sincroniza con la modulada de entrada, pero se produce un error del 100% en la onda binaria recuperada.

4.- Figura B2.7. Como la diferencia de fase entre la portadora modulada y la creada por el VCO es menor que  $90^\circ$ , y los elementos componentes del bucle son aceptables, la sincronización se da sin ninguna clase de error.

5.- Figura B2.8. Los filtros Pasabajos han sido estructurados de tal manera que su filtrado no es el deseado, ya que su banda de paso permite pasar componentes de frecuencia que deben ser filtrados para que las respuestas del bucle no contengan errores. En consecuencia, la señal binaria tiene imperfecciones ya que el margen de discriminación para el ecualizador es demasiadamente pequeño.

6.- Considerando las curvas de B2.3 y B2.9, se observa claramente que el retardo de la señal binaria con respecto a la modulada recibida en el bucle, es menor en B2.9 puesto que en este bucle intervienen filtros de Butterworth; los que tienen una respuesta transitoria mejor que los elípticos, consecuentemente su respuesta es mas rápida al cambio de fase de la señal modulada recibida.

Debido a la estructura de las bandas de frecuencia del filtro de Butterworth (La selectividad del filtro de Butterworth es pobre comparada con la del filtro Elíptico), el filtro del bloque demodulador presenta picos que produce un margen de discriminación pequeño para el ecualizador.

7.- Figura B2.10. Las frecuencias críticas de los filtros pasabajos son mejor elegidas que en el bucle que da las curvas B2.9, por lo que se evita el problema decisión del ecualizador, analizado en el punto anterior.

8.- Figura B2.11 y B2.12. Considerando las curvas de estas dos condiciones, se puede deducir que es deseable que la frecuencia de portadora sea igual o múltiplo entero del ritmo de transmisión; de esta manera, una vez realizada la modulación por cada bit se tiene ciclos completos de portadora. Requiriéndose que la recuperación del sincronismo se de solo en el primer bit o los primeros bits de acuerdo a las características del bucle, mas no como se observa en estas curvas en cada cambio de fase de la señal recibida en el bucle.

### 5.3.BUCLE CON REMODULACION POR EL OSCILADOR LOCAL

1.- Figura B3.1. La señal modulada en este bucle  $e_7(t)$ , es igual a la modulada recibida  $e_1(t)$  excepto en el intervalo desde el cambio de fase de  $e_1(t)$ , el demodulador cambia el nivel de onda binaria  $e_8(t)$  y ésta varía la modulación de  $e_7(t)$ . Con estas condiciones del VCO, su señal de control contiene picos que producen deformaciones de la señal creadas a su salida.

2.- Figura B3.2. Persiste el ruido del caso anterior, ya que las condiciones del VCO son las mismas, pero si comparamos con B2.3, se deduce que la recuperación de sincronismo es mas rápida que en el bucle de costas general.

3.- Considerando B3.2 y B3.3, la recuperación de sincronismo es mas rápida en B3.3 ya que la sensibilidad del VCO es también mayor, pero el ruido del bucle mencionado en los casos anteriores aumenta debido a la misma sensibilidad.

4.- Figura B3.4. El VCO se ha estructurado de tal manera

que se reduce el ruido que molestaba en los casos anteriores, esto se ha logrado a aumentando la tolerancia del VCO.

5.- Tanto en B3.5 como en B3.6 se da un error del 100% en la recuperación de sincronismo, ya que la diferencia de fase original entre la creada por el VCO y la recibida es mayor que  $90^\circ$ .

6.- Figura B3.7. Al ser los elementos del bucle aceptable y el error de fase inicial menor que  $90^\circ$  se recupera correctamente el sincronismo del bucle.

7.- Figura B3.8. Las condiciones del bucle son iguales a las del bucle que da las curvas de B3.7, excepto por el filtro pasabajos, que en B3.8 ha sido mal seleccionado ya que no filtra los armónicos para cuya función se lo ha puesto; en consecuencia se presentan errores en la onda binaria demodulada.

8.- Figura B3.9. Al igual que en el bucle de costas general, se nota que el retardo de la onda binaria respecto a la recibida es despreciable, debido a las características del filtro de Butterworth en régimen transitorio.

#### 5.4. BUCLE CON REMODULACION POR LA SEÑAL RECIBIDA

1.- Figura B4.1. La señal modulada en este bucle  $e_7(t)$ , es igual a la creada por el VCO  $e_2(t)$  excepto en el intervalo desde el cambio de fase de  $e_1(t)$ , el demodulador cambia el nivel de onda binaria  $e_5(t)$  y esta varía la modulación de  $e_7(t)$ . Con las características del VCO, al igual que en B3.1 producen deformaciones de la señal creadas a su salida.

2.- Figura B4.2. Persiste el ruido del caso anterior, ya que las condiciones del VCO son las mismas, pero si comparamos con B2.3, se deduce que la recuperación de sincronismo es mas rápida que en el bucle de costas general (B2.3).

3.- Considerando B4.2 y B4.3, la recuperación de sincronismo es mas rápida en B4.3 ya que la sensibilidad del VCO es también mayor; pero el ruido del bucle mencionado en los casos anteriores aumenta, debido a la misma sensibilidad.

4.- Figura B4.4. Al aumentar la tolerancia del VCO, los picos a la salida del comparador de fase tiene influencia en lapsos menores de tiempo, por lo que se reduce el ruido que molestaba en los casos anteriores.

5.- Tanto en B4.5 como en B4.6 se da un error del 100% en la recuperación de sincronismo, ya que la diferencia de fase original entre la creada por el VCO y la recibida es mayor que  $90^\circ$ .

6.- Figura B4.7. Al ser los elementos del bucle aceptable y el error de fase inicial menor que  $90^\circ$  se recupera correctamente el sincronismo del bucle.

7.- Figura B4.8. Las condiciones del bucle son iguales a las del bucle que da las curvas de B4.7, excepto por el filtro pasabajos, que en B4.8 ha sido mal seleccionado ya que no filtra los armónicos para cuya función se lo ha puesto, luego se da errores en la onda binaria demodulada.

8.- Figura B4.9. Al igual que en el bucle de costas general, se nota que el retardo de la onda binaria respecto

a la recibida es despreciable, debido a las características del filtro de Butterworth en régimen transitorio.

### 5.5. BUCLE DE COSTAS CON DETECCION DENTRO DE UN BRAZO

1.- Figura B5.1. Con los componentes del bucle estructurados correctamente y el error de fase inicial menor que  $90^\circ$ , se observa una sincronización sin errores del bucle.

2.- Tanto en las curvas de B5.2 como de B5.3, la señal creada por el VCO se sincroniza con la recibida, pero se produce un error del 100% en la onda binaria recuperada, ya que el error de fase inicial es mayor que  $90^\circ$ .

3.- Figura B5.4. Se ha cambiado el tren de bits : 00100. A la onda creada en el VCO se le ha impuesto un adelanto de  $60^\circ$  [La respuesta inicial del VCO es  $\cos(W_p t + 60^\circ)$ ]; al ser el primer bit 0 lógico, la onda recibida es  $-\cos(W_p t)$ . Aún cuando el desfase inicial es de  $240^\circ$ , se logra la recuperación correcta del sincronismo.

4.- Observando B5.1, B5.4 y B5.5, se deduce que el error del 100% en la señal binaria se da si el error de fase inicial es mayor de  $90^\circ$  con respecto a la portadora original [ $\cos(W_p t)$ ], mas no de la diferencia de fase relativa que existe inicialmente entre la señal del VCO y la recibida.

### 5.6. BUCLE DE COSTAS NUMERICO

1.- Figura B6.1. Al ser el error de fase inicial igual a cero, la salida de la compuerta AND es siempre cero por lo

que el VCO se mantiene en sincronismo con la señal recibida, recuperándose la onda binaria sin errores a no ser por el retardo de filtrado.

2.- Figura B6.2. Se tiene el pulso necesario para establecer el sincronismo del bucle.

3.- Figura B6.3. A diferencia del resto de bucles analizados, la señal de control del VCO no es proporcional a la diferencia de fase entre las ondas recibida y creada por el VCO, sino que es un pulso que bruscamente cambia de fase de la onda a la salida del VCO, por lo que el diseño se hace inseguro pues hay que probar su funcionamiento cambiando el nivel de la señal de control del VCO y los parámetros del mismo para obtener buenos resultados.

4.- B6.4. Se da pulsos en la señal de control del VCO tendientes a sincronizar el bucle únicamente cuando la onda binaria recuperada toma el valor lógico 1, como en este caso no se presenta coincidencia en 1 lógico entre  $e_s(t)$  y  $e_o(t)$  no se sincroniza. Como el error de fase inicial está dentro del rango de error del 100%, es lo que se da.

5.- Figura B6.5. El error de fase inicial se ha cambiado a  $110^\circ$  con respecto al caso anterior; se produce recuperación de sincronismo aunque se debe a las imperfecciones del filtrado en el filtro del demodulador, el rango de decisión para el ecualizador es muy pequeño por lo que se da errores en la onda binaria.

6.- Figura B6.6. Se mantiene el error de fase inicial del caso anterior pero, se ha cambiado los filtros pasabajos por otros de orden mayor es decir más selectivos. Considerando



B6.5 y B6.6, se deduce que en la recuperación de sincronismo de este bucle tiene gran importancia el orden del filtro, ya que de él depende su filtrado. En B6.5 son las imperfecciones a la salida del filtro del demodulador las que permiten recuperar el sincronismo, no así en B6.6.

7.- Figura B6.7. Al ser el error de fase inicial mayor que  $90^\circ$  y los elementos del bucle aceptables se da el error de 100% en la señal binaria recuperada.

8.- Figura B6.8. La señal de control del VCO no tiene el pulso para recuperar el sincronismo pero al ser el error de fase inicial menor que  $90^\circ$ , la onda binaria es la deseada.

#### 5.7. BUCLE DE COSTAS CON REMODULACION PARA QPSK

Secuencia del tren de bits : 11 00 10 01 11

Como se mencionó en el capítulo II, transformando  $e_o(t)$  y  $e_s(t)$  de paralelo a serie, donde el primer bit está dado por  $e_o(t)$ ; si  $e_o(t)=0$  y  $e_s(t)=1$  los bits recuperados son 01.

1.- Figura B7.1. Es un caso ideal donde el error de fase inicial es cero. Esto es la portadora dada por el VCO es  $\cos(W_p t + 0^\circ)$ , mientras que la onda recibida llega con su ángulo de fase dado en la modulación de acuerdo a la información  $\cos(W_p t + \phi^\circ)$ .

Respaldando la teoría se nota que las señales  $e_a(t)$  y  $e_b(t)$  tienden a ser iguales, excepto en pequeños intervalos en los que la señal recibida por el bucle cambia su fase. Al ser iguales  $e_a(t)$  y  $e_b(t)$ , la señal de control del VCO  $e_7(t) = e_b(t) - e_a(t)$  es cero, en consecuencia la respuesta del

VCO no sufre ningún cambio.

Se nota la recuperación del tren de bits originales a no ser por el retardo entre la señal recibida y la onda binaria, producido por el filtrado y un error de discriminación instantánea en el ecualizador.

2.- Figura B7.2 Debido a las imperfecciones del filtrado se dan errores en la señal ecualizada del brazo en cuadratura.

3.- Figura B7.3. Analizando con respecto al eje imaginario, al ser el primer par de bits 11, existe un retardo de  $45^\circ$  de la onda recibida, mientras que a la señal del VCO se le ha impuesto un adelanto de  $60^\circ$ , de este modo realmente existe un retardo de  $105^\circ$  de la onda modulada con respecto a la creada en el bucle. La sincronización tiende hacia el estado mas próximo este es el que tiene un retardo de  $135^\circ$ , razón por la que no se recupere como primeros bits 11 ( $-45^\circ$ ) sino que se obtiene 10 ( $-135^\circ$ ). De este modo recordando el diagrama de constelaciones dado en el capítulo 1 (Figural.1), los pares binarios se recuperan con un error de  $90^\circ$  en sentido antihorario en el diagrama de constelaciones. Así se recupera 10 por 11, 00 por 10, 01 por 00 y 11 por 01.

4.- Figura B7.4. Siendo el primer par de bits 11, existe un retardo de  $45^\circ$  de la onda recibida, mientras que a la señal del VCO se le ha impuesto un adelanto de  $120^\circ$ , por consiguiente existe un retardo de  $165^\circ$  de la onda modulada con respecto a la creada en el bucle. La sincronización tiende hacia el estado mas próximo en el sentido antihorario este es el que tiene un retardo de  $225^\circ$ , razón por la que no se recupere como primeros bits 11 ( $-45^\circ$ ) sino que se obtiene

00 ( $-225^\circ$ ). De tal forma que, los pares binarios se recuperan con un error de  $180^\circ$  en el diagrama de constelaciones. Esto es, se recupera 00 por 11, 01 por 10, 11 por 00 y 10 por 01. Obteniéndose un error del 100 % en la onda binaria demodulada.

5.- Figura B7.5. Igual que el caso anterior, al ser el primer par de bits 11, existe un retardo de  $45^\circ$  de la onda recibida, mientras que a la señal del VCO se le ha impuesto un adelanto de  $240^\circ$ , existiendo de este modo realmente un retardo de  $285^\circ$  de la onda modulada con respecto a la creada en el bucle. La sincronización tiende hacia el estado mas próximo en el sentido antihorario, este es el que tiene un retardo de  $315^\circ$ , razón por la que no se recupere como primeros bits 11 ( $-45^\circ$ ) sino que se obtiene 01 ( $-315^\circ$ ). De este modo, los pares binarios se recuperan con un error de  $270^\circ$  en sentido antihorario en el diagrama de constelaciones. Esto es, se recupera 01 por 11, 11 por 10, 10 por 00 y 00 por 01.

6.- Figura B7.6. Al ser el primer par de bits 11, existe un retardo de  $45^\circ$  de la onda recibida, mientras que a la señal del VCO se le ha impuesto un adelanto de  $300^\circ$ , por consiguiente existe un retardo de  $345^\circ$  de la onda modulada con respecto a la creada en el bucle. La sincronización tiende hacia el estado mas próximo en el sentido antihorario este es el que tiene un retardo de  $45^\circ$ , razón por la que se recupere como primeros bits 11 ( $-45^\circ$ ) que es el par originalmente fue modulado. De tal forma que la recuperación del sincronismo esta libre de errores.

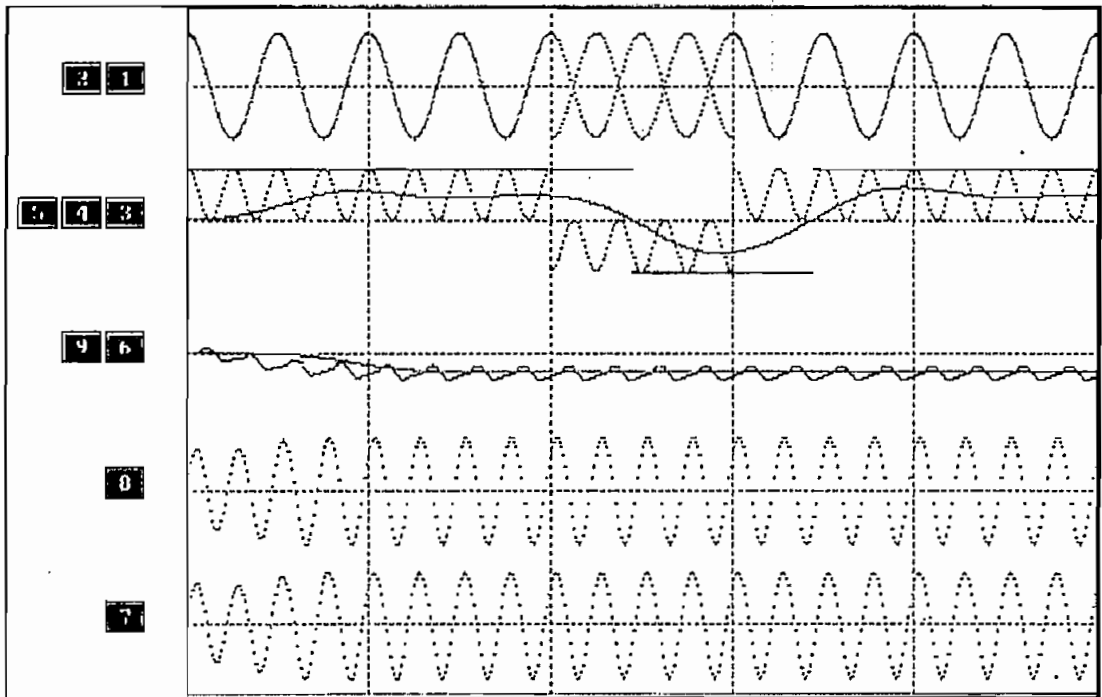


FIGURA B1.1 : Ec Ec X4 (A)

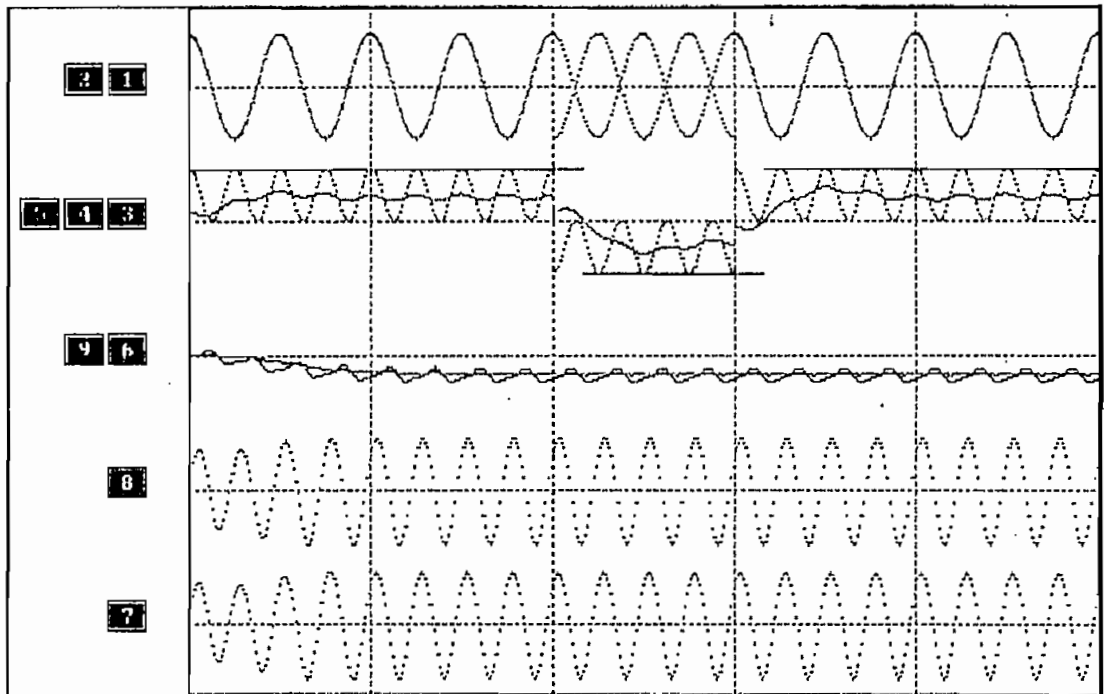


FIGURA B1.2 : Eb Eb X4 (A)

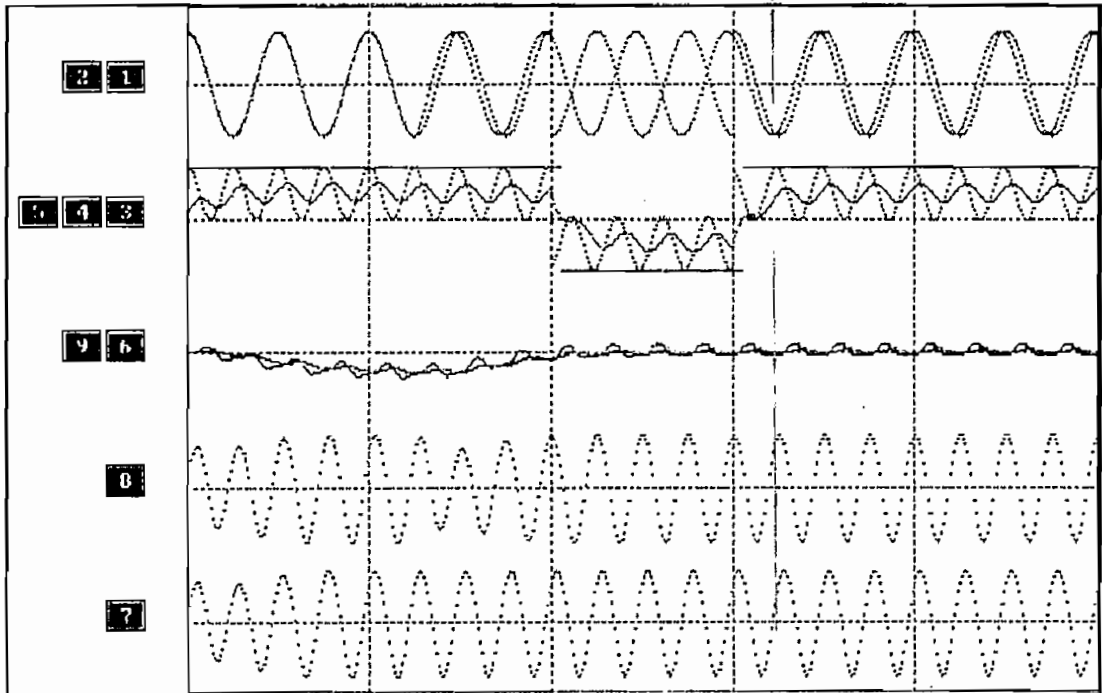


FIGURA B1.3 : Ea Ea X4 (A)

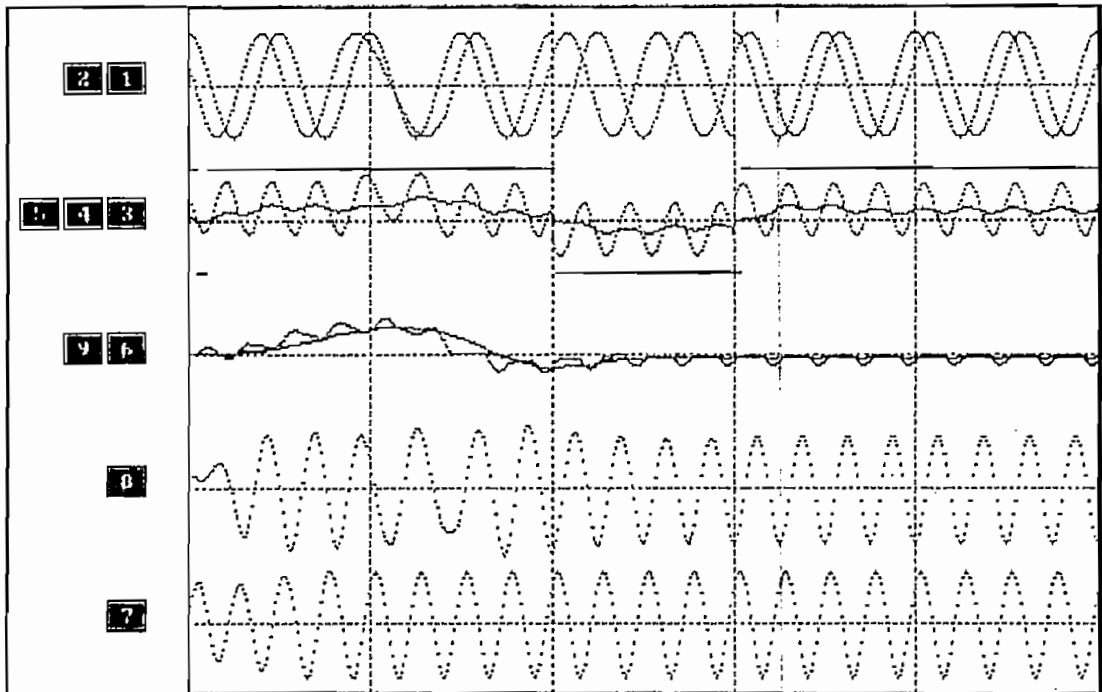


FIGURA B1.4 : Eb Eb X4 (B)

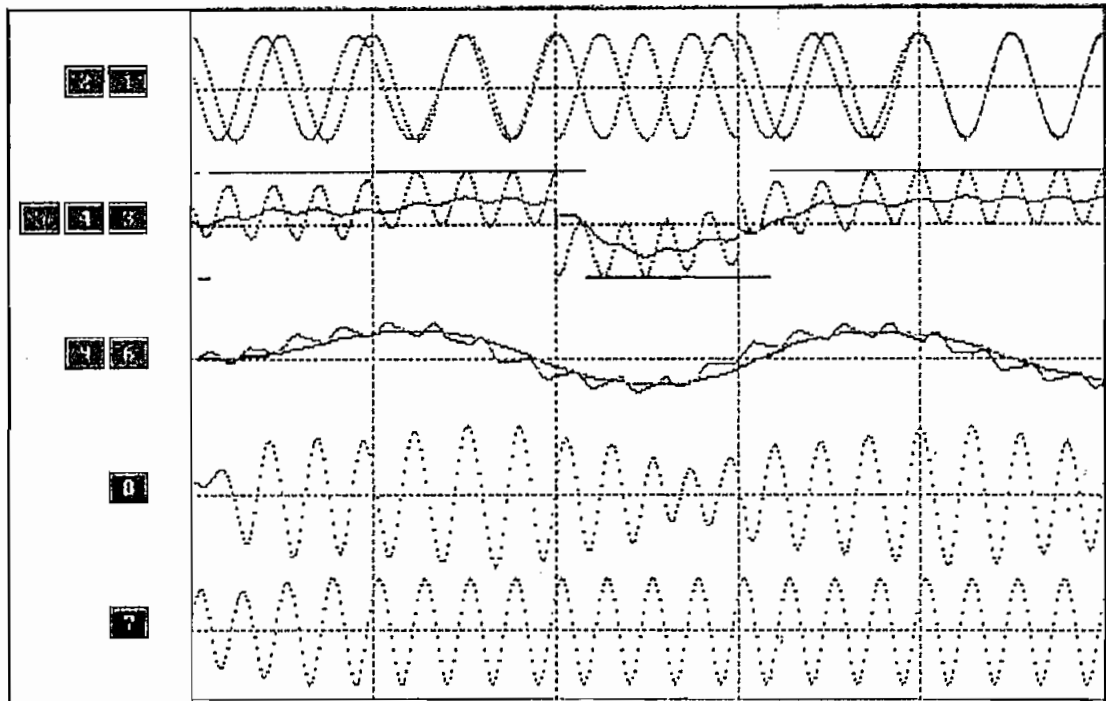


FIGURA B1.5 : Eb Eb X5 (B)

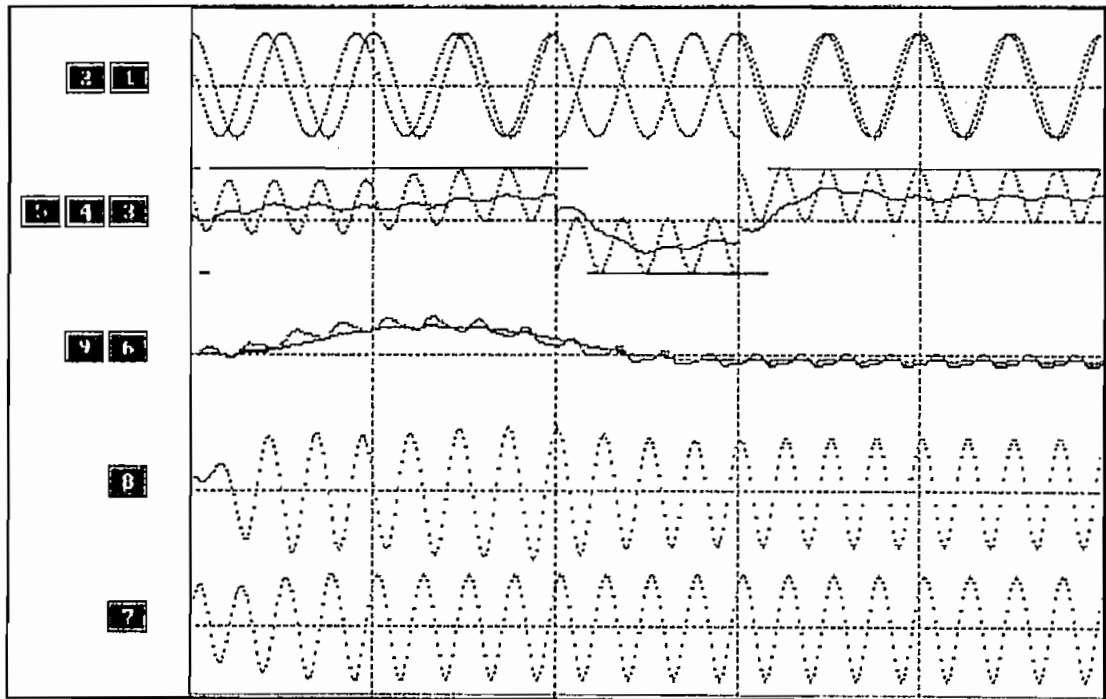


FIGURA B1.6 : Eb Eb X6 (B)

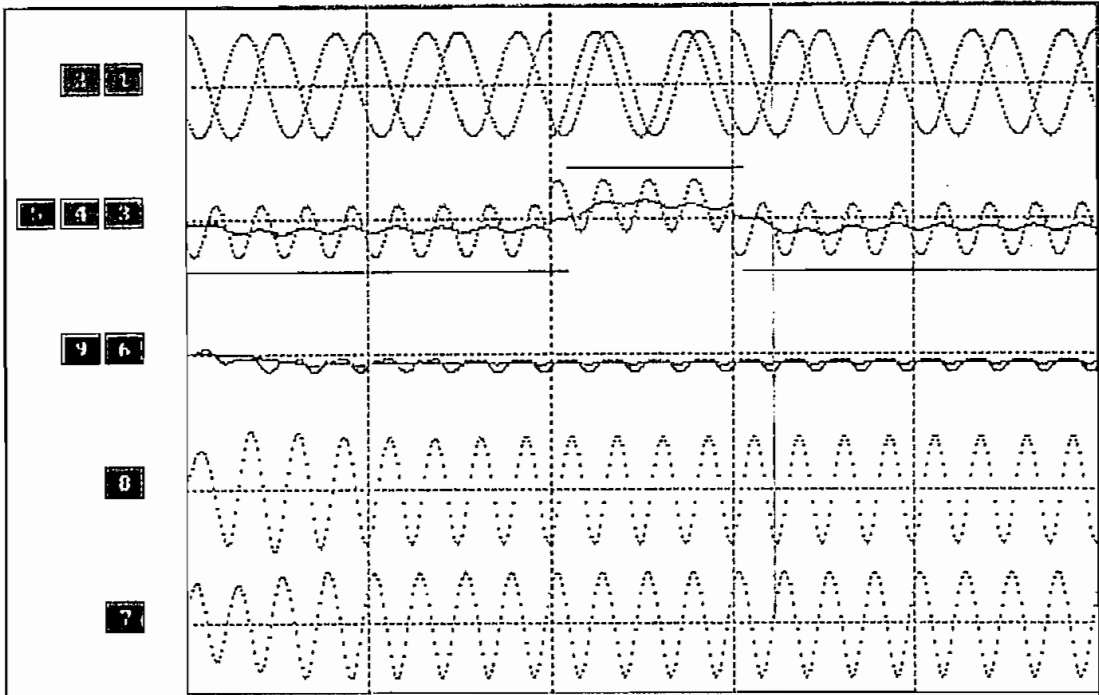


FIGURA B1.7 : Eb Eb X6 (C)

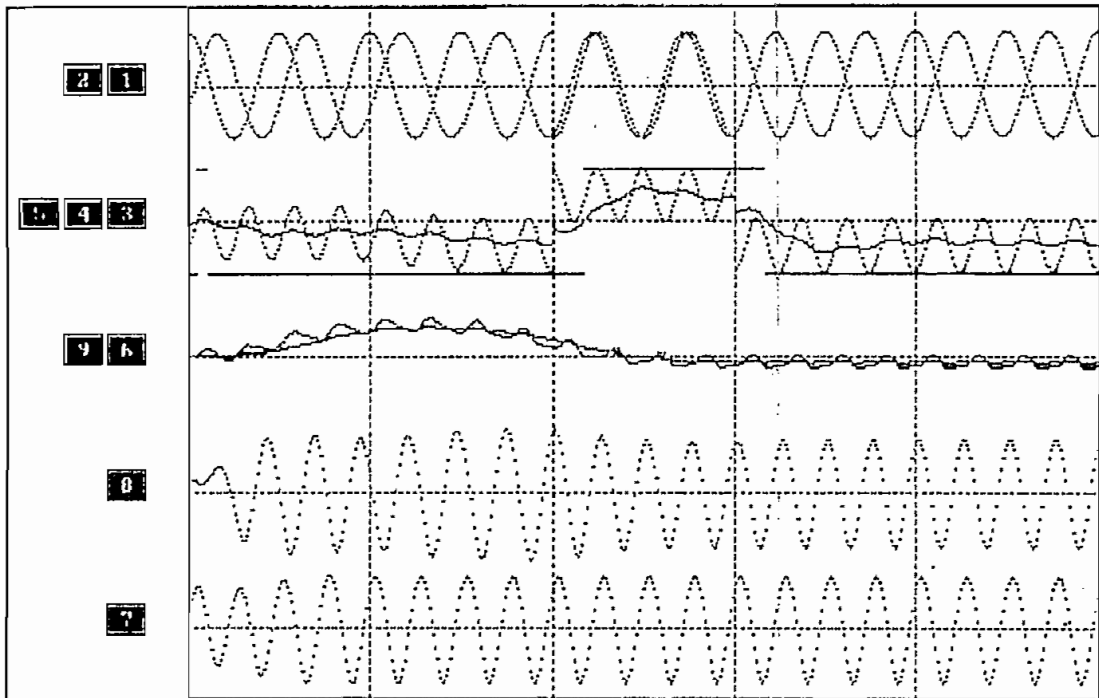


FIGURA B1.8 : Eb Eb X6 (D)

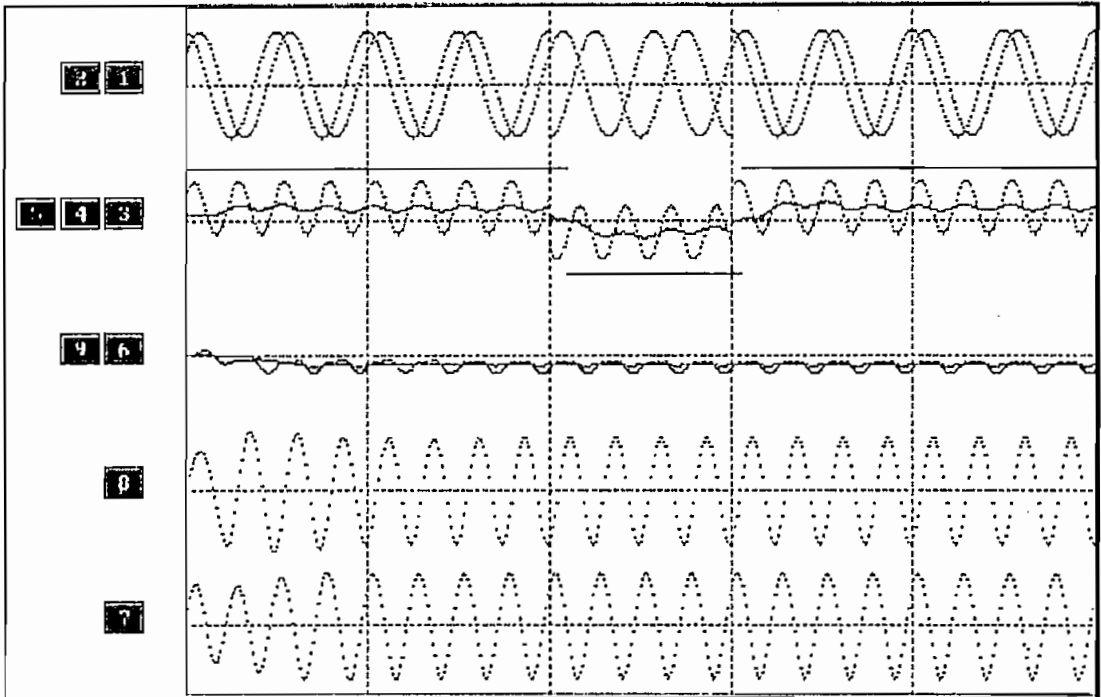


FIGURA B1.9 : Eb Eb X6 (E)

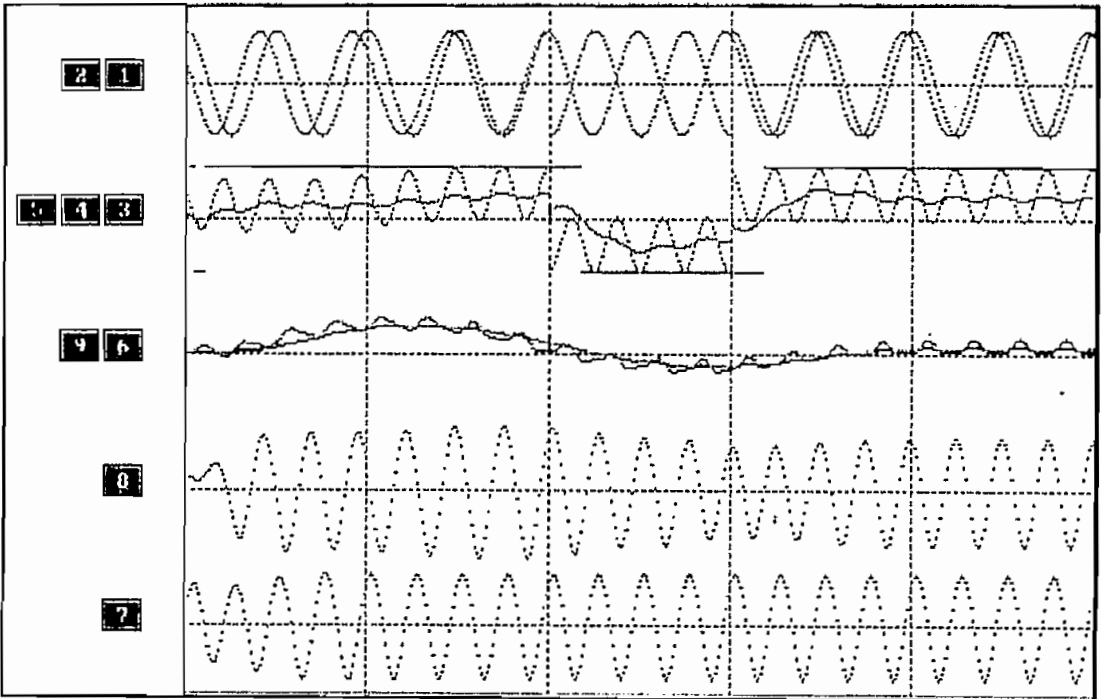


FIGURA B1.10 : Eb Eb X7 (B)



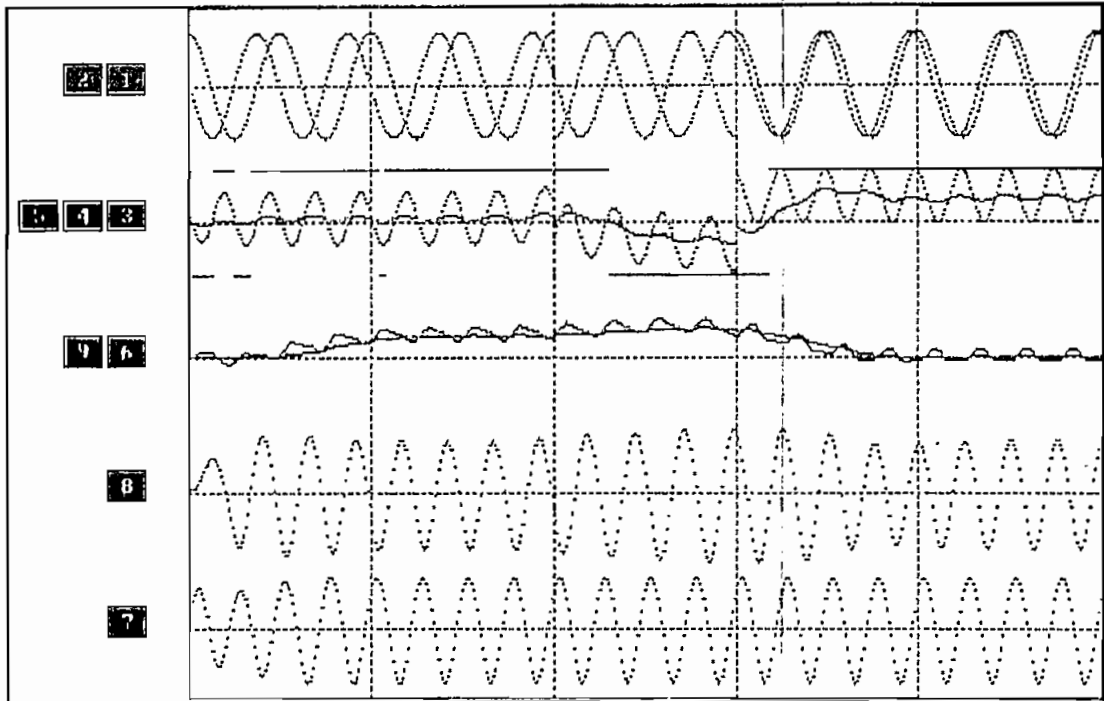


FIGURA B1.11 : Eb Eb X7 (C)

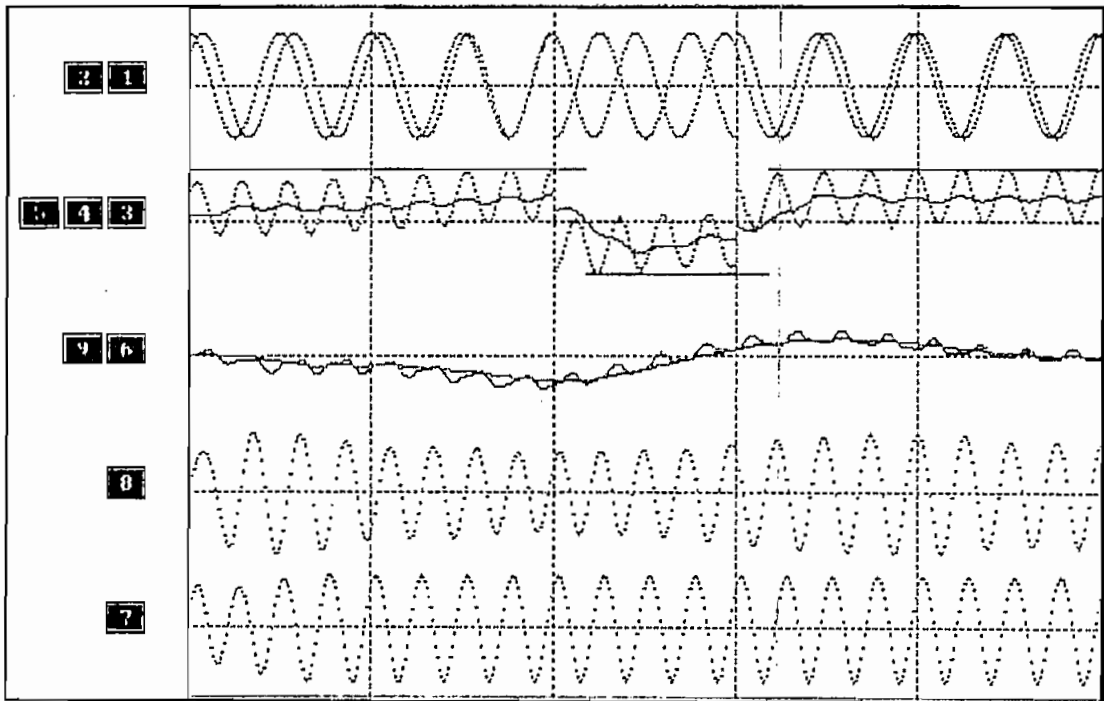


FIGURA B1.12 : Eb Eb X7 (E)

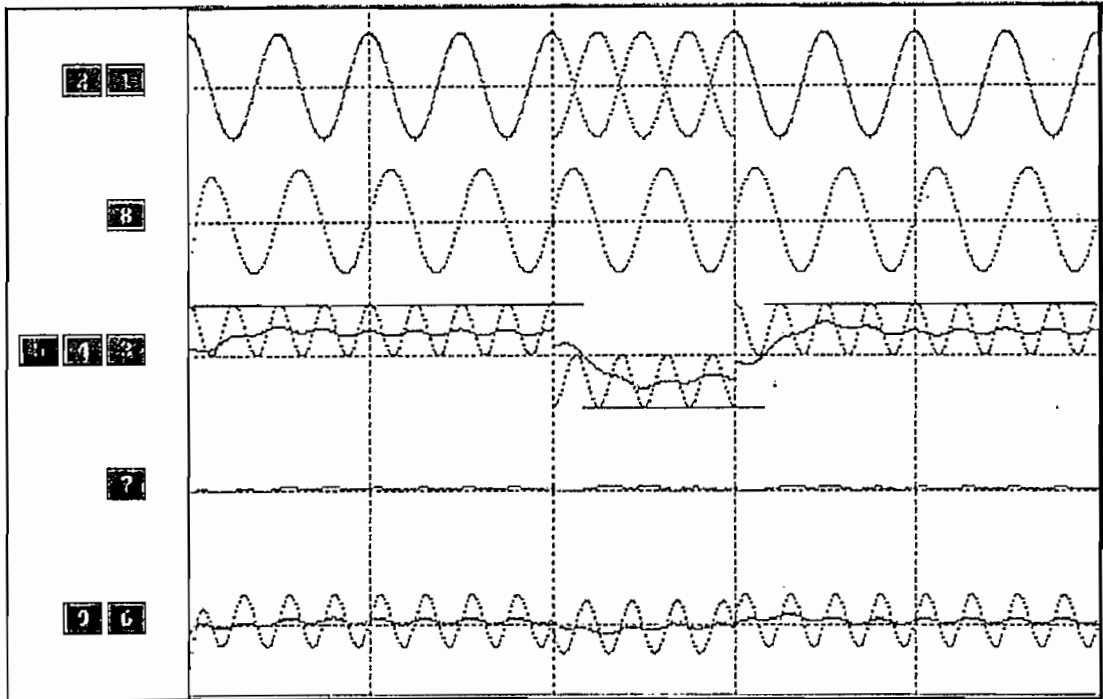


FIGURA B2.1 : Eb Eb X1 (A)

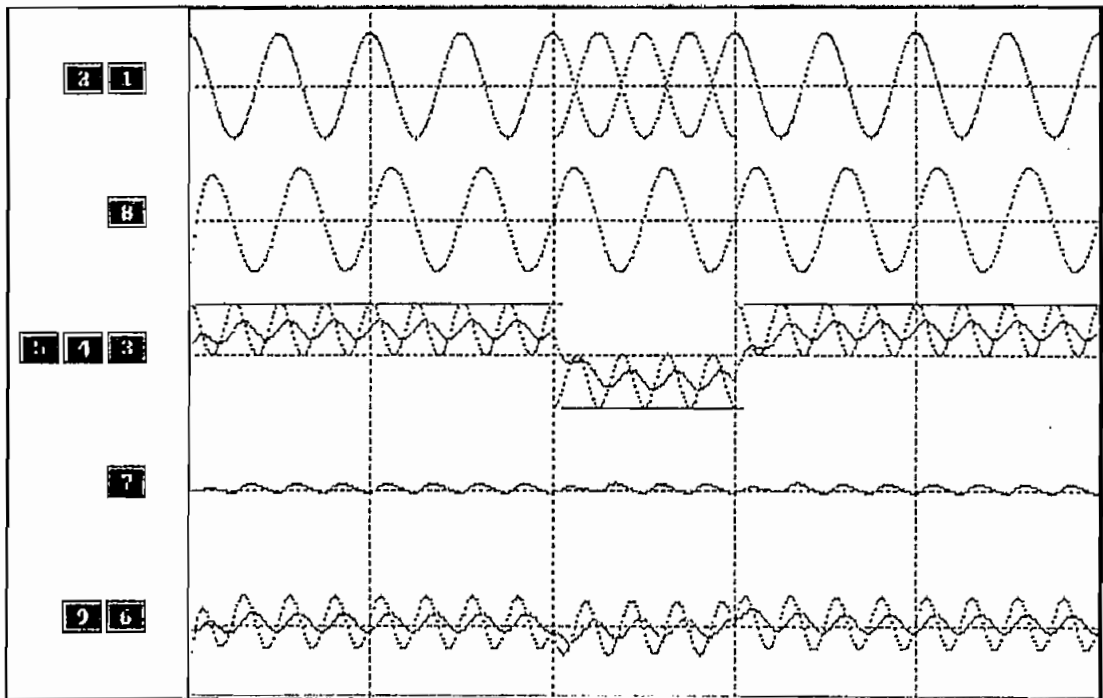


FIGURA B2.2 : Ea Ea X1 (A)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

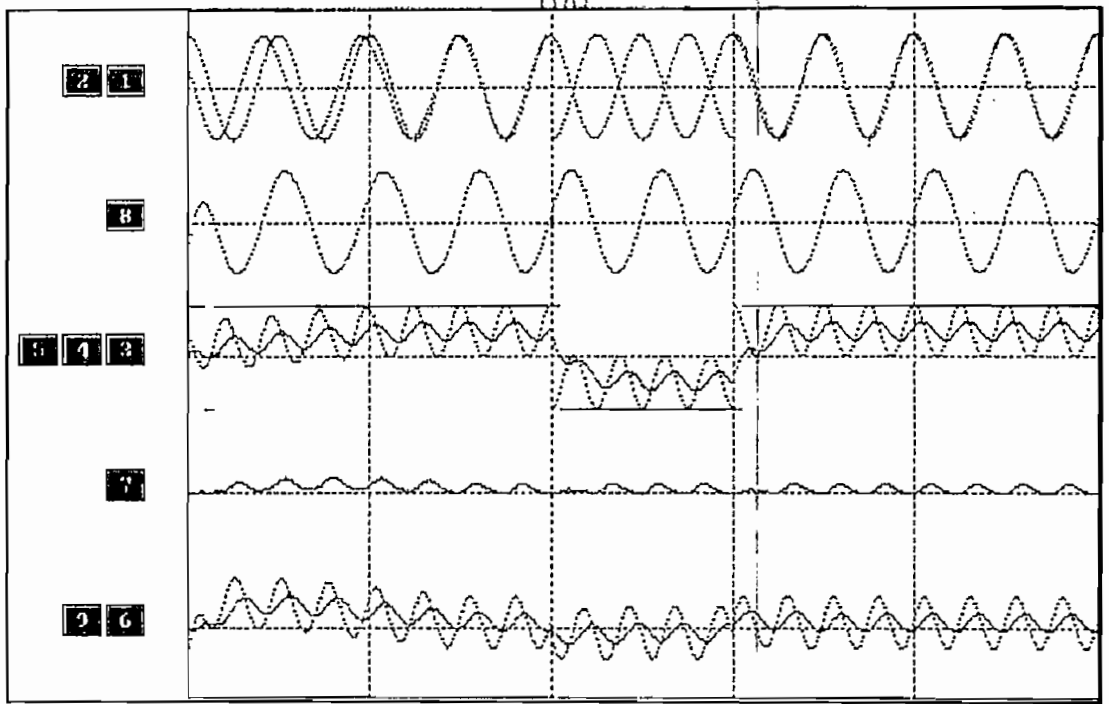


FIGURA B2.3 : Ea Ea X1 (B)

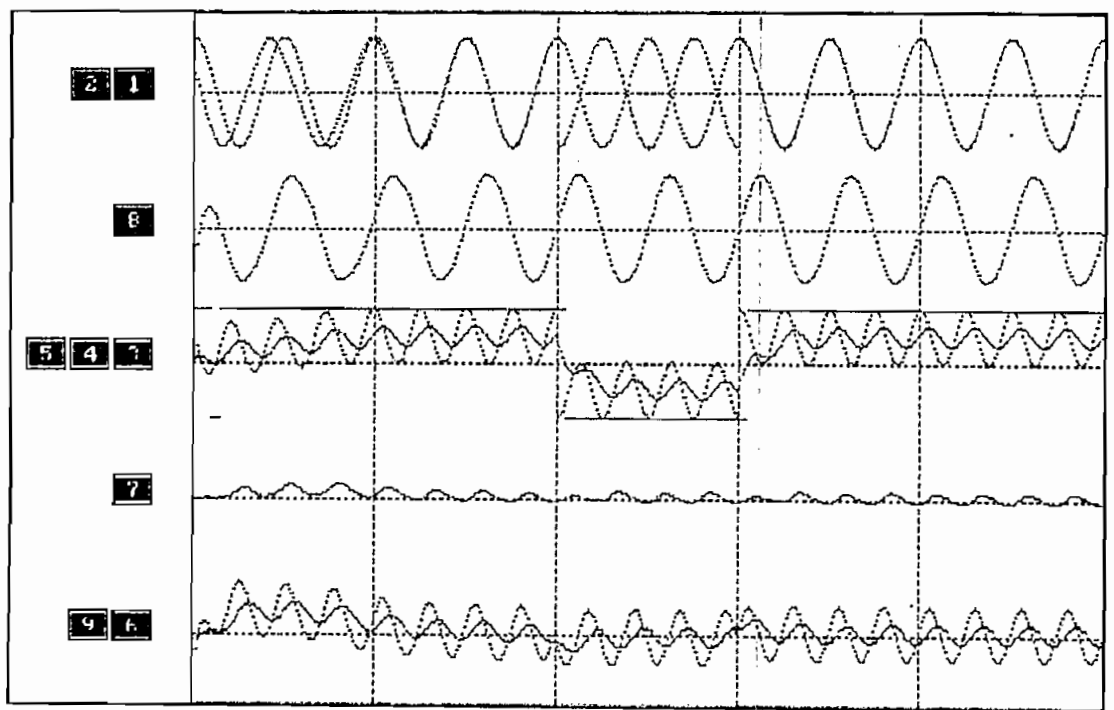


FIGURA B2.4 : Ea Ea X2 (B)

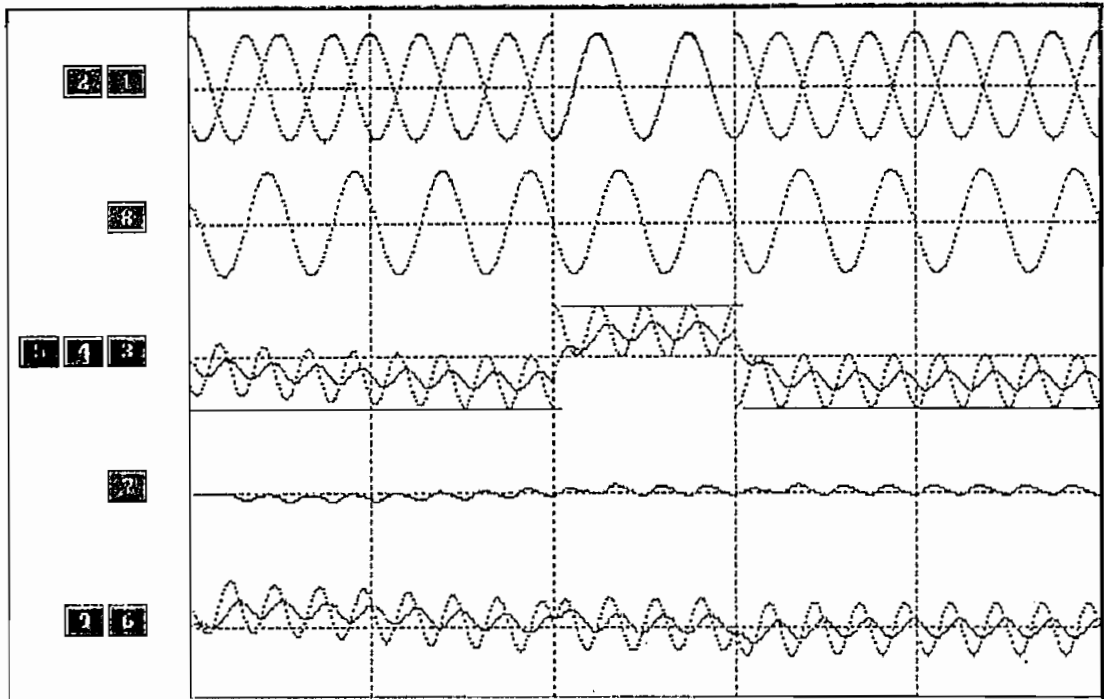


FIGURA B2.5 : Ea Ea X1 (C)

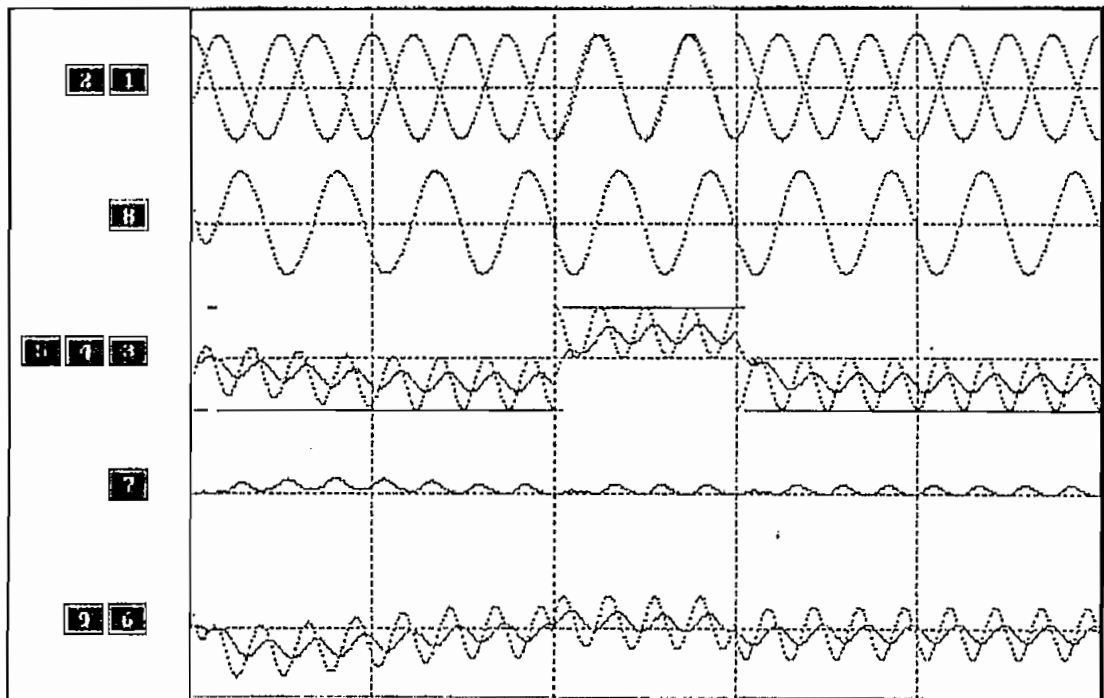


FIGURA B2.6 : Ea Ea X1 (D)

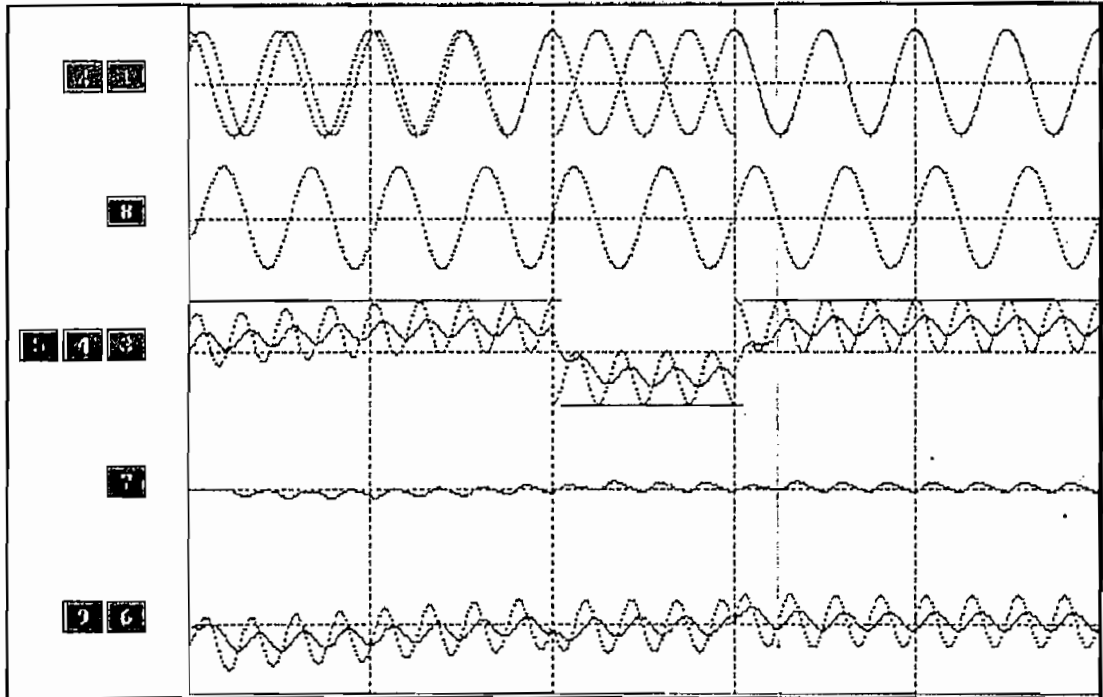


GRAFICO B2.7 : Ea Ea X1 (E)

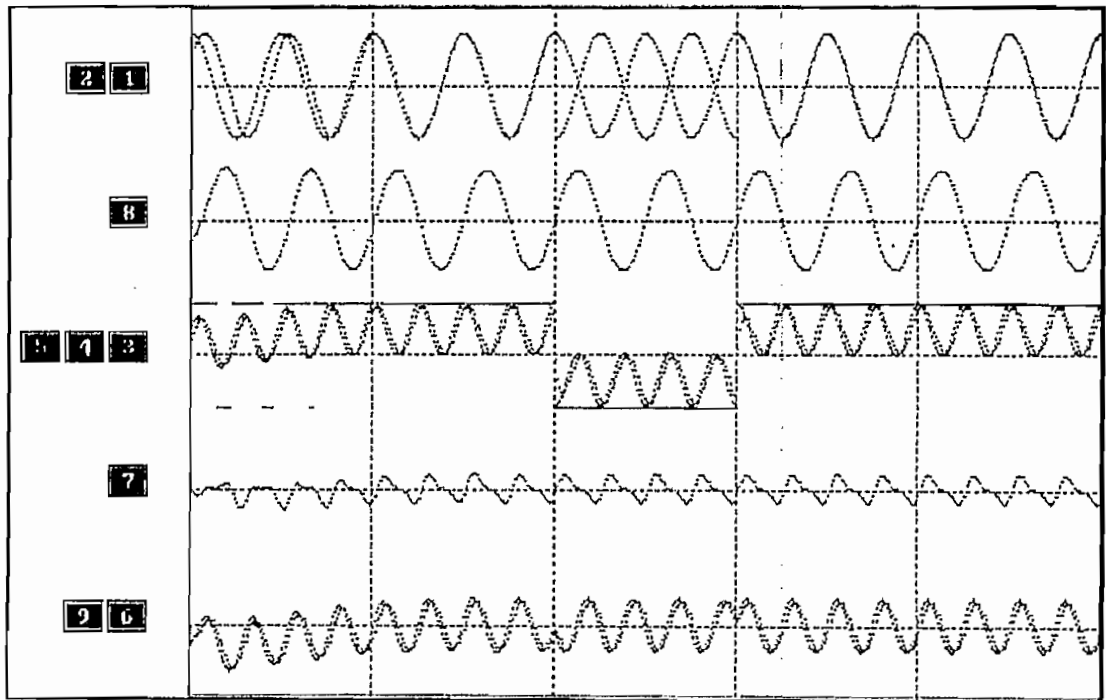


FIGURA B2.8 : Ed Ed X1 (E)

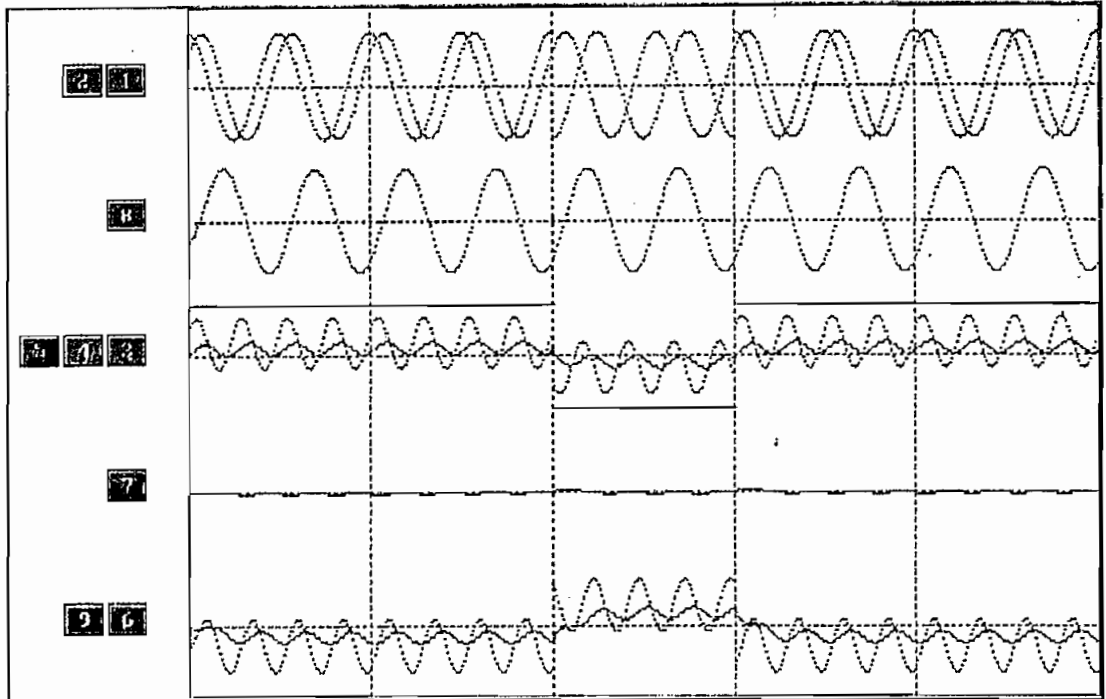


FIGURA B2.9 : Bb Bb X1 (E)

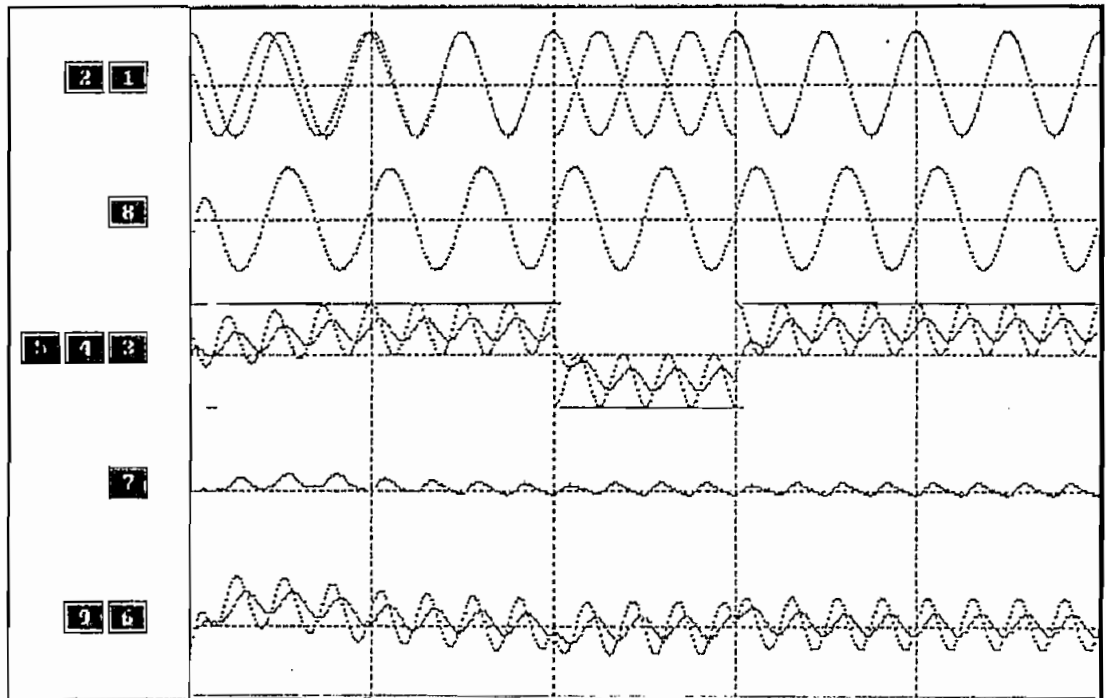


FIGURA B2.10 : Be Be X1 (B)

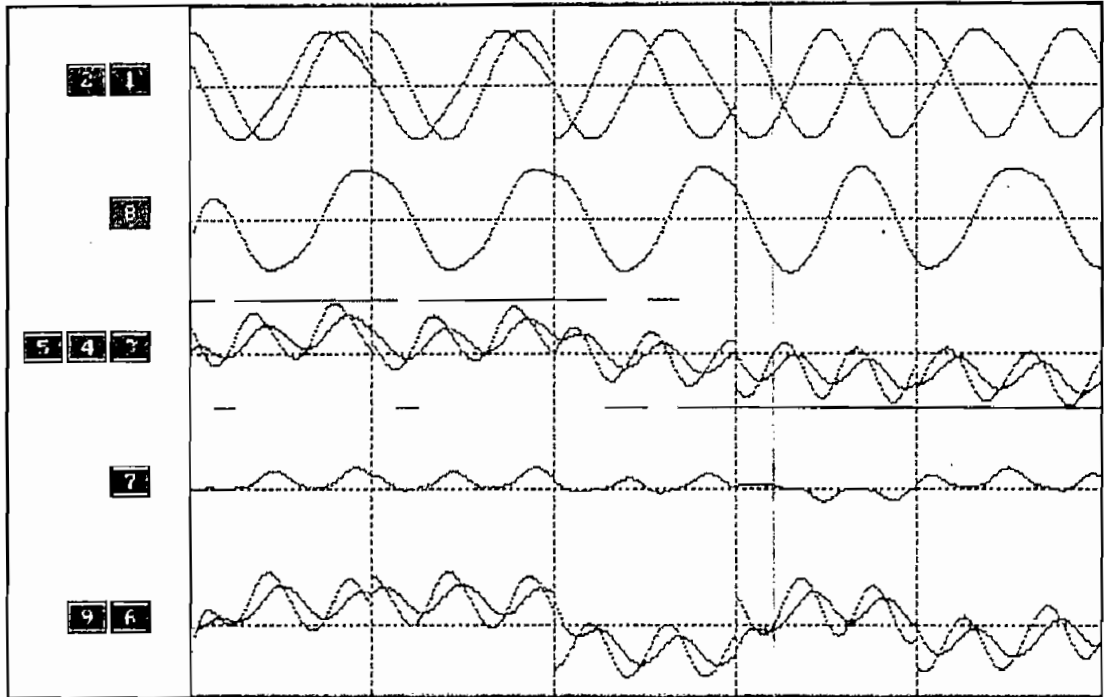


FIGURA B2.11: Ea Ea X2 (B) Ritmo de transmisión=2000

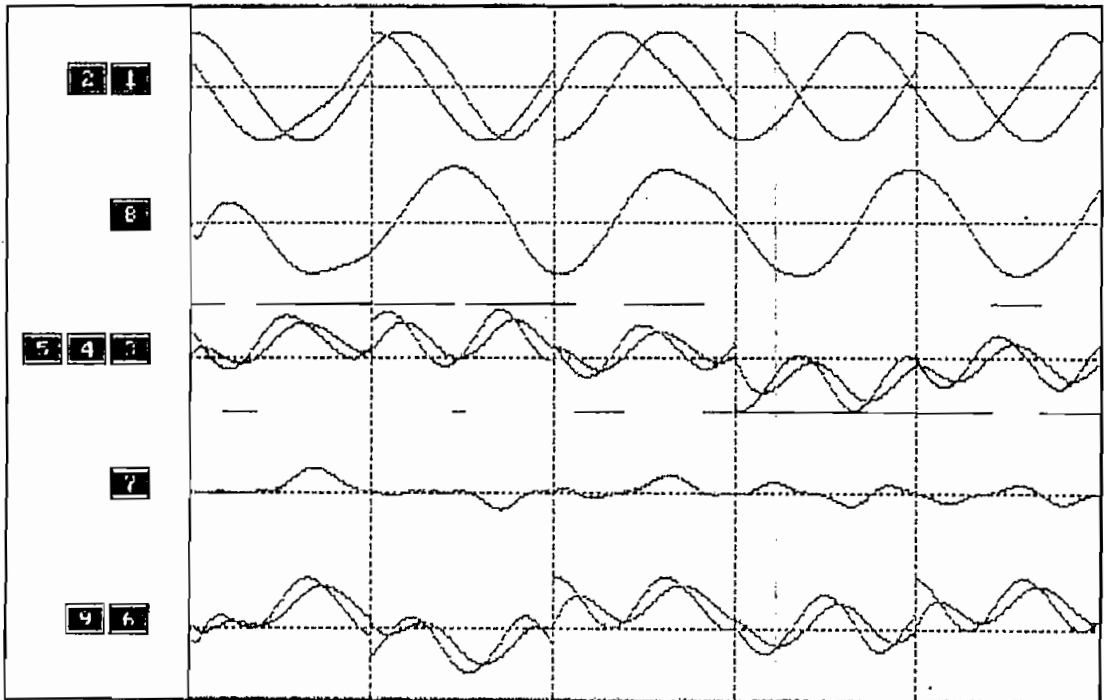


FIGURA B2.12: Ea Ea X2 (B) Ritmo de transmisión(R)=3000

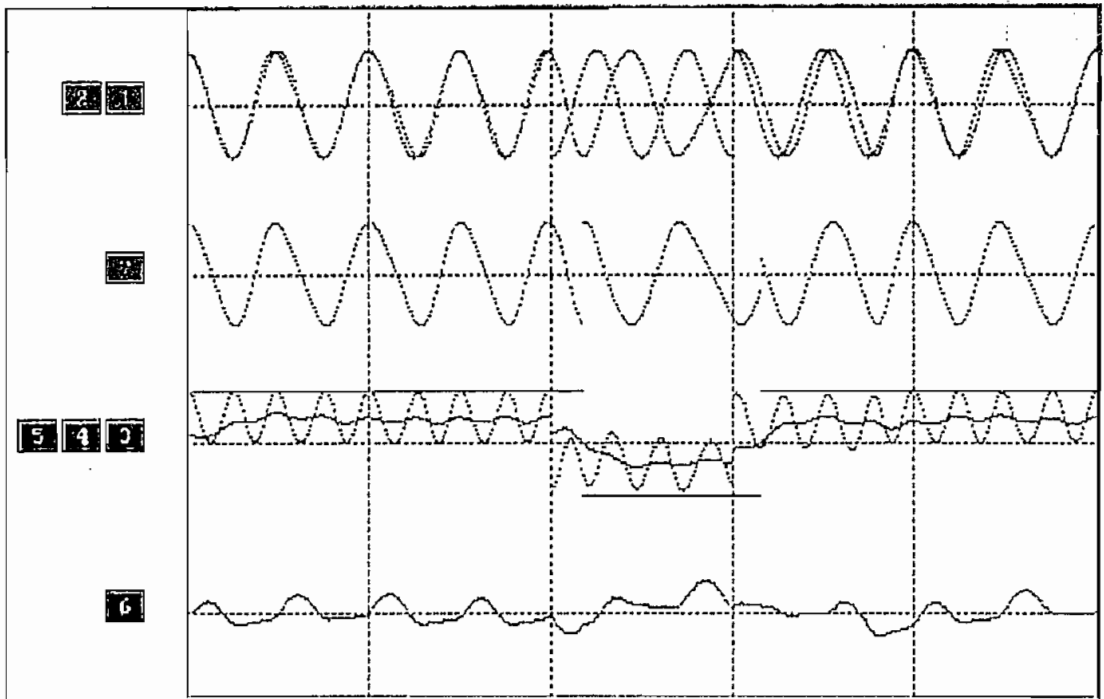


FIGURA B3.1 : Eb X1 (A)

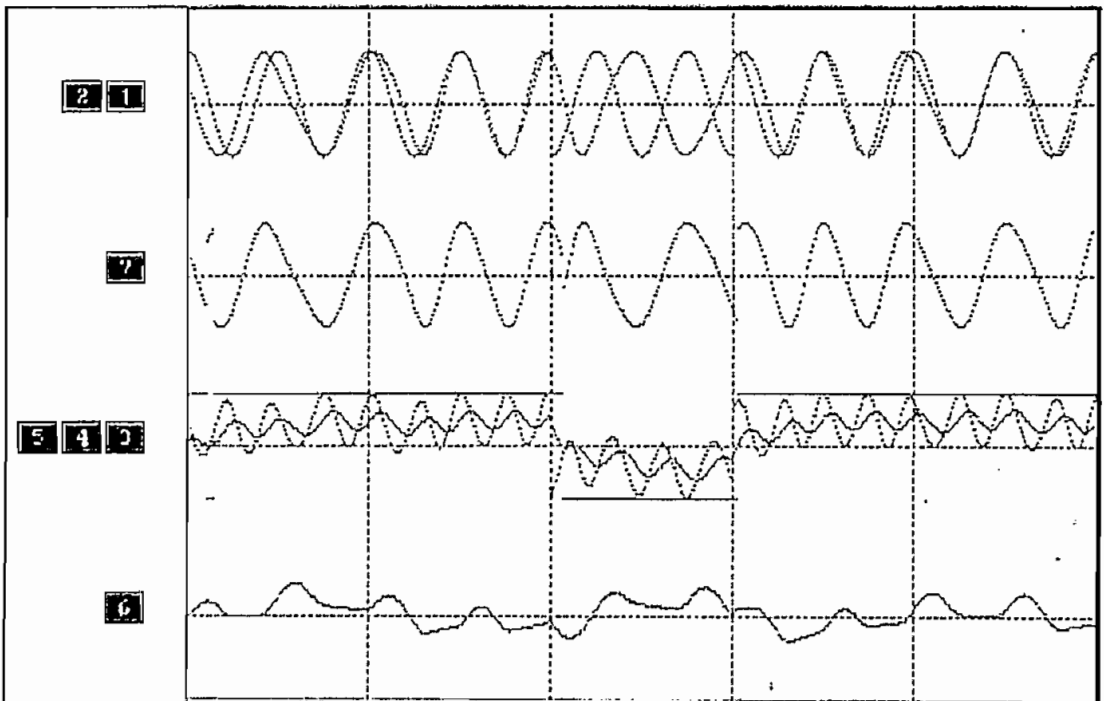


FIGURA B3.2 : Ea X1 (B)



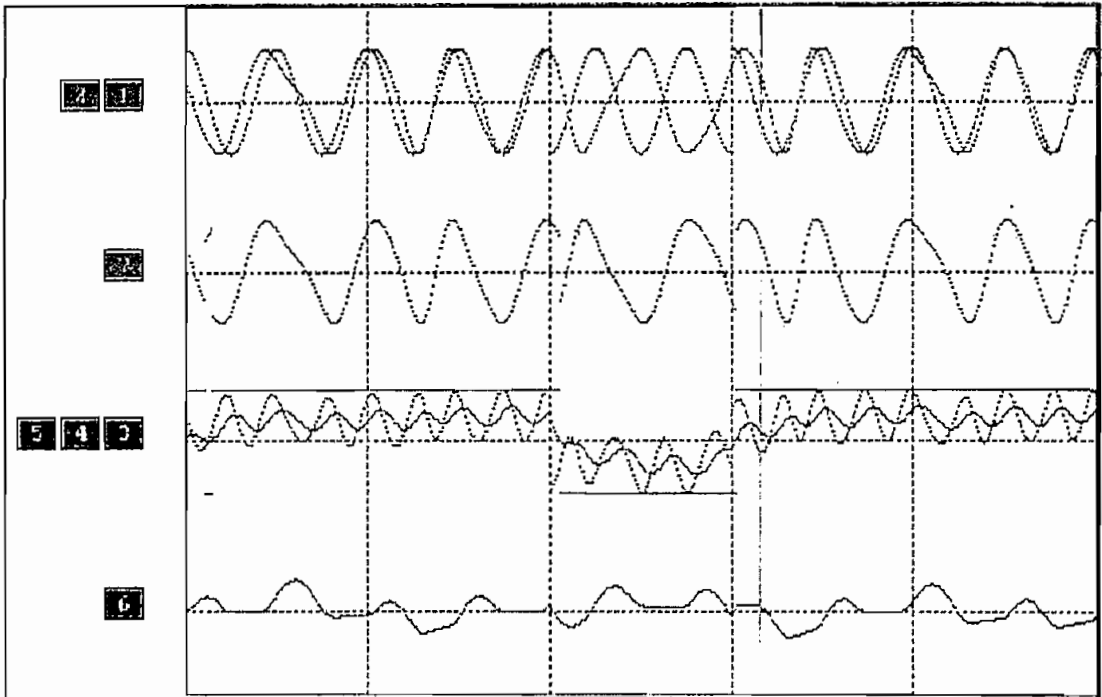


FIGURA B3.3 : Ea X2 (B)

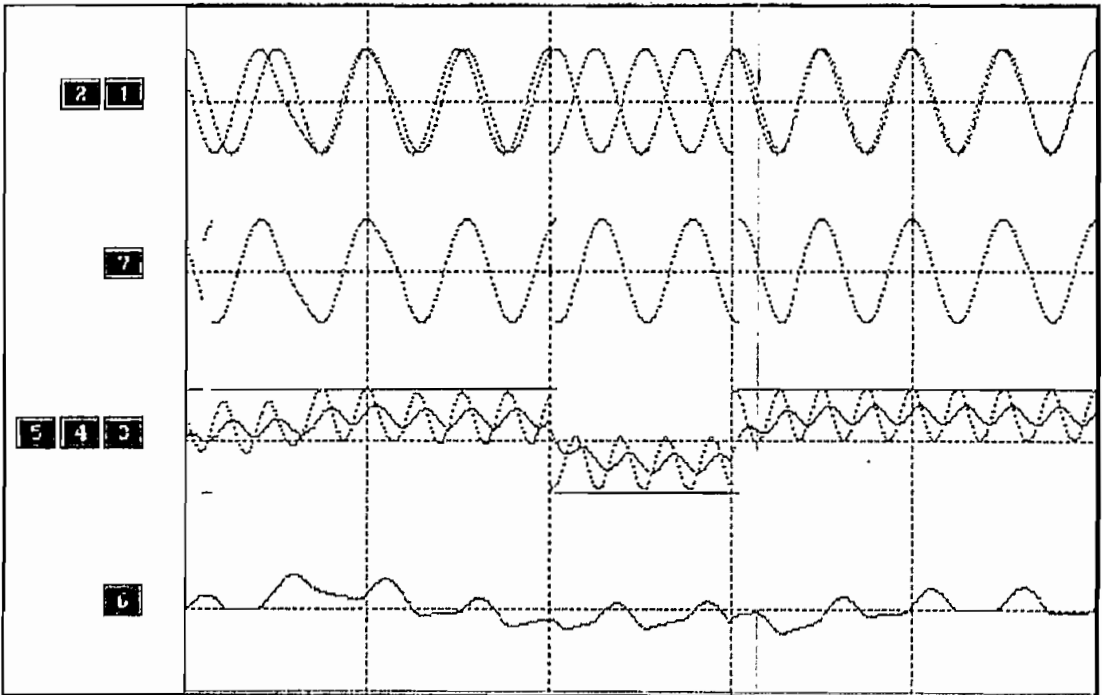


FIGURA B3.4 : Ea X3 (B)

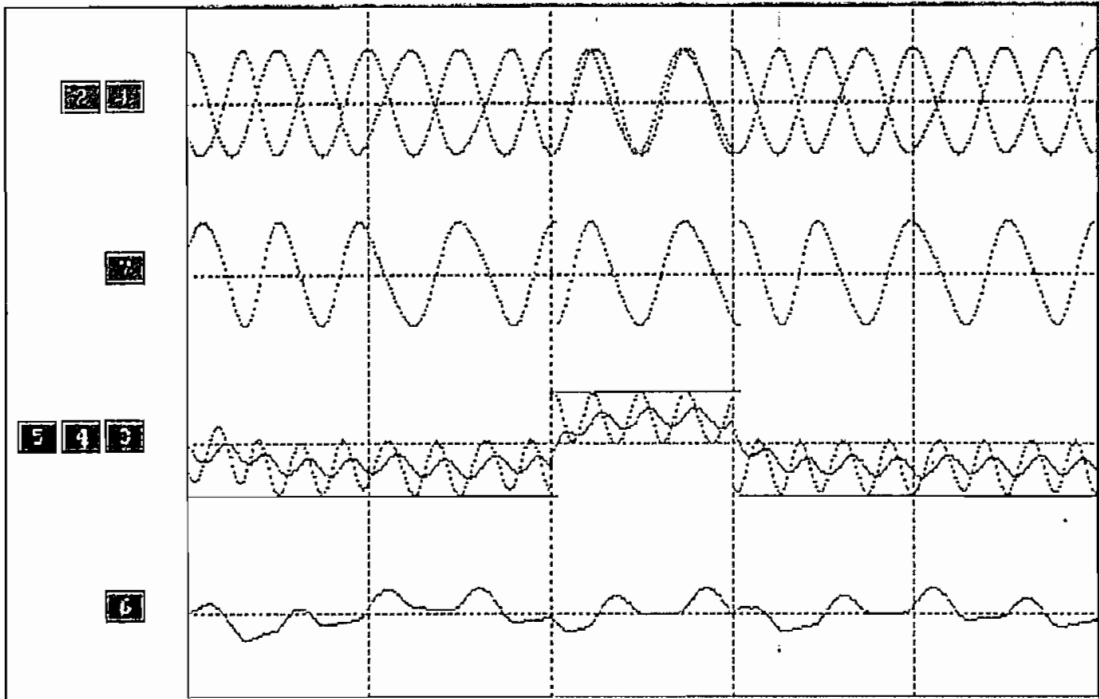


FIGURA B3.5 : Ea X1 (C)

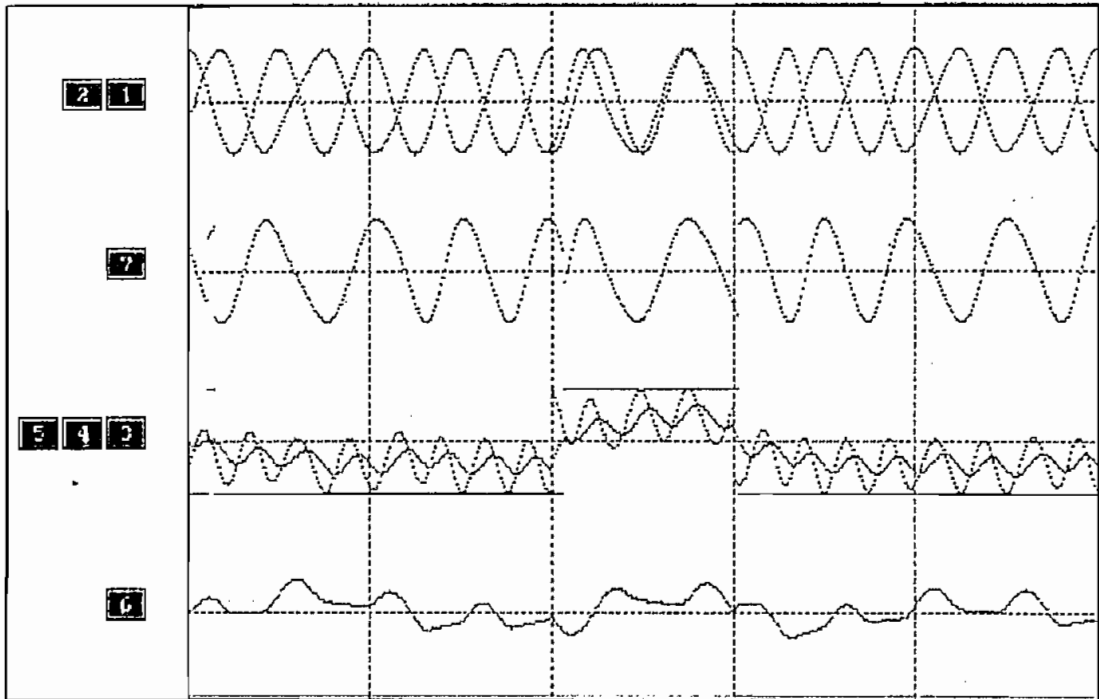


FIGURA B3.6 : Ea X1 (D)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

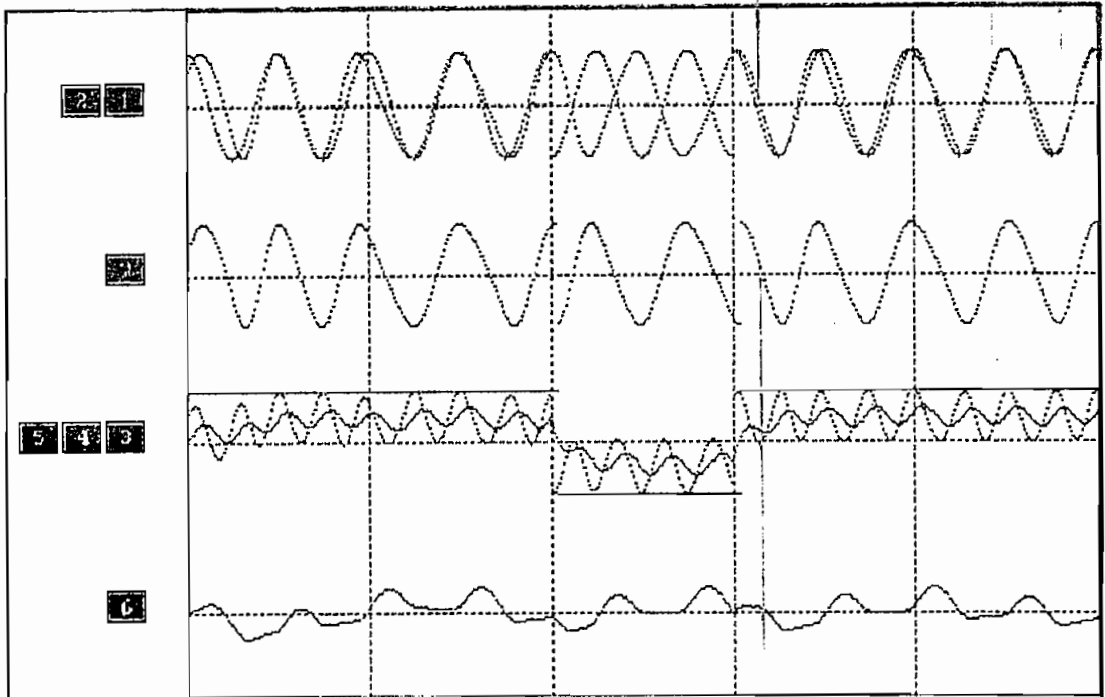


FIGURA B3.7 : Ea X1 (E)

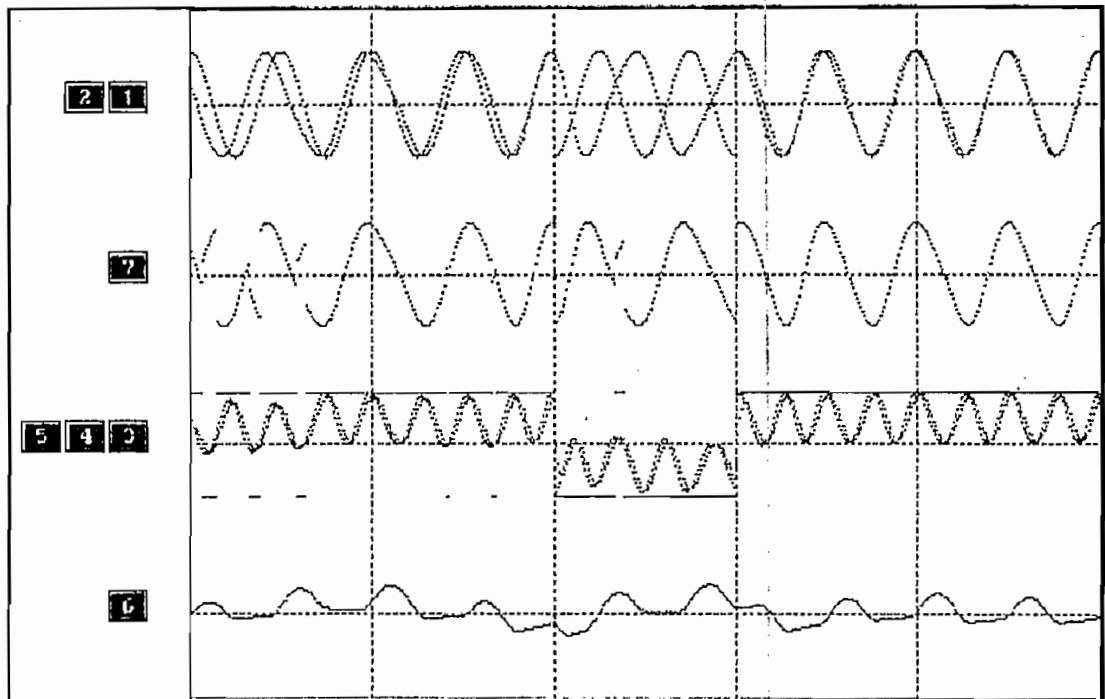


FIGURA B3.8 : Ed X1 (B)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

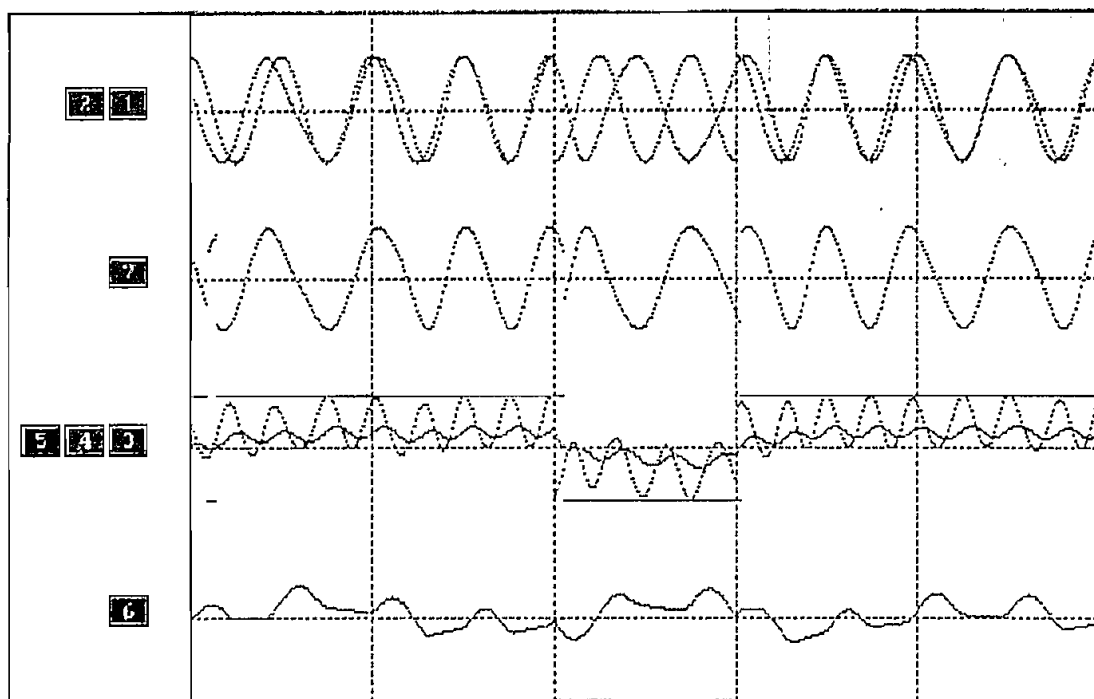


FIGURA B3.9 : Eb X1 (B)

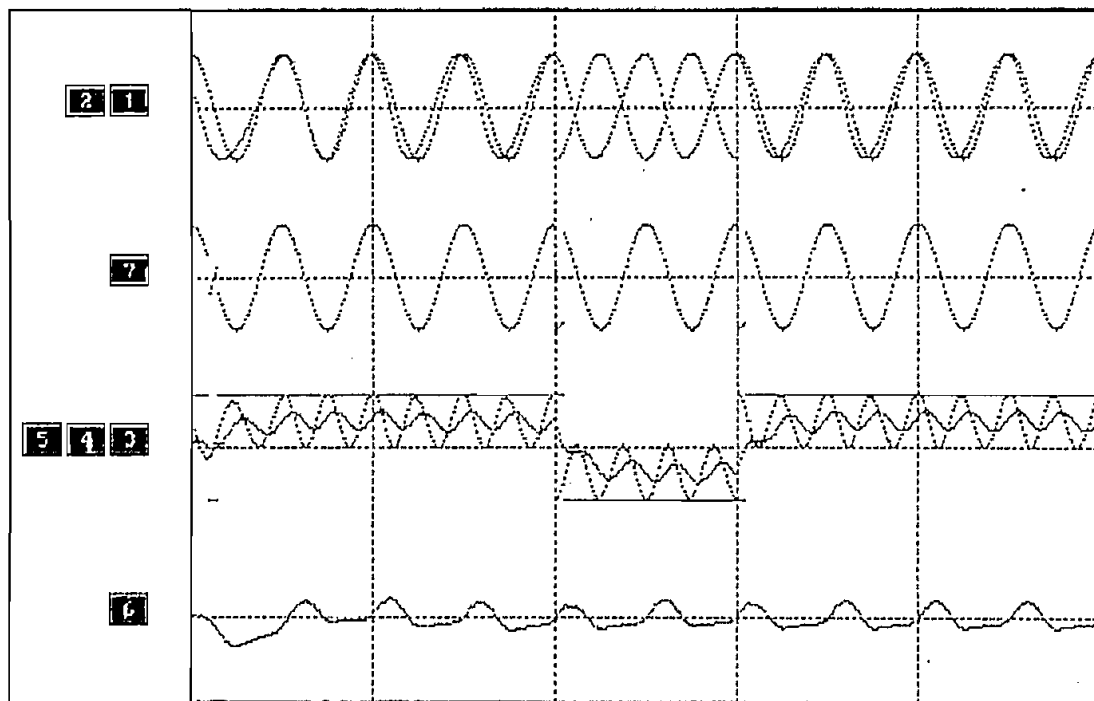
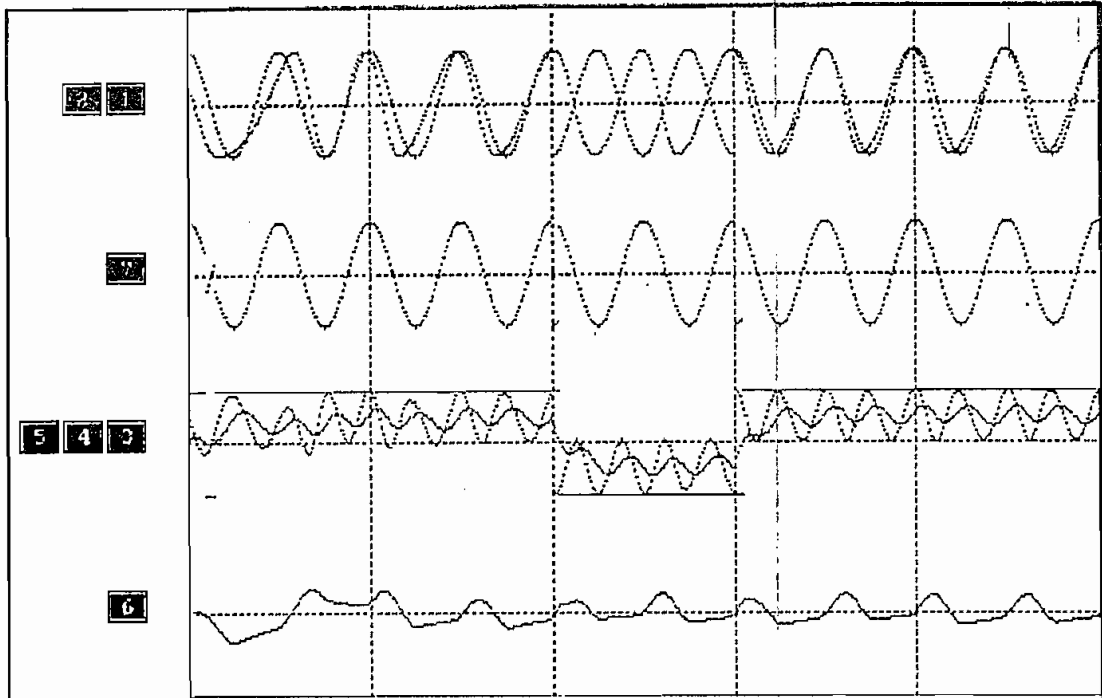
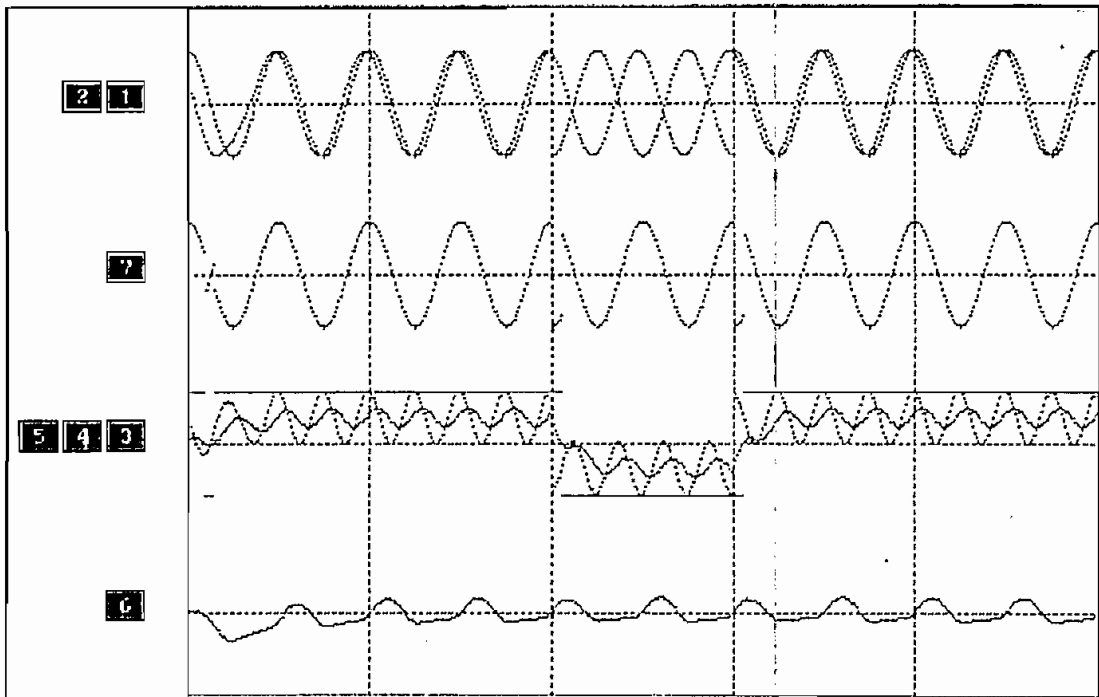


FIGURA B4.1 : Eb X1 (B)

FIGURA 4.2 :  $E_a$  X1 (B)FIGURA 4.3 :  $E_a$  X2 (B)

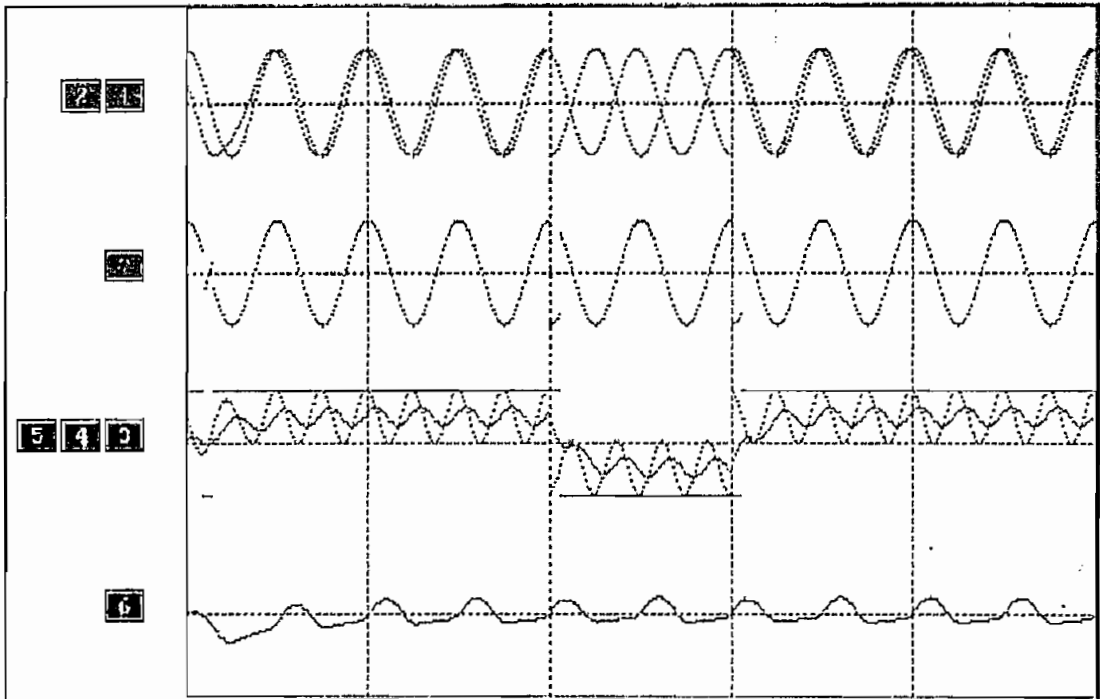


FIGURA B4.4 : Ea X3 (B)

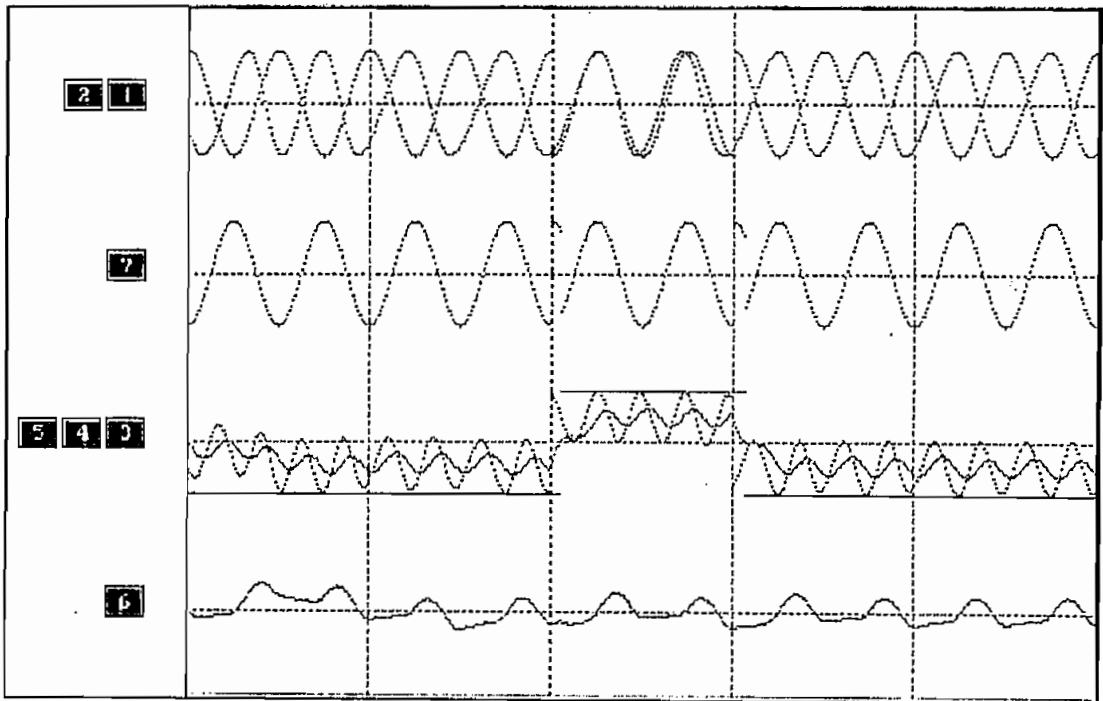


FIGURA B4.5 : Ea X1 (C)

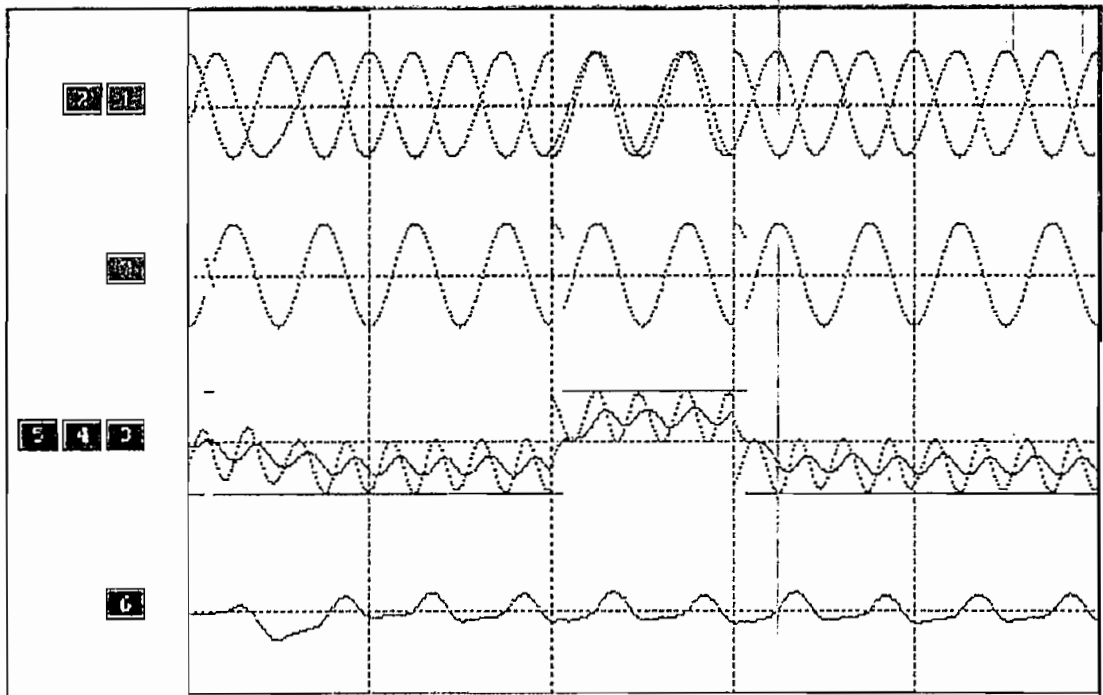


FIGURA B4.6 : Ea X1 (D)

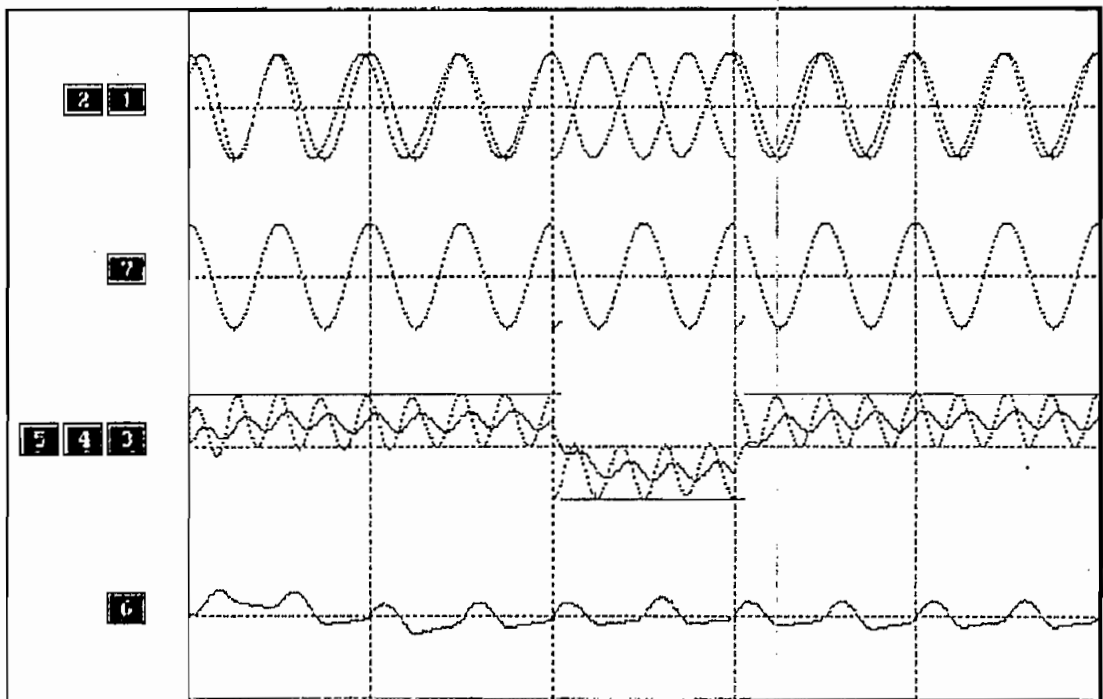


FIGURA B4.7 : Ea X1 (E)

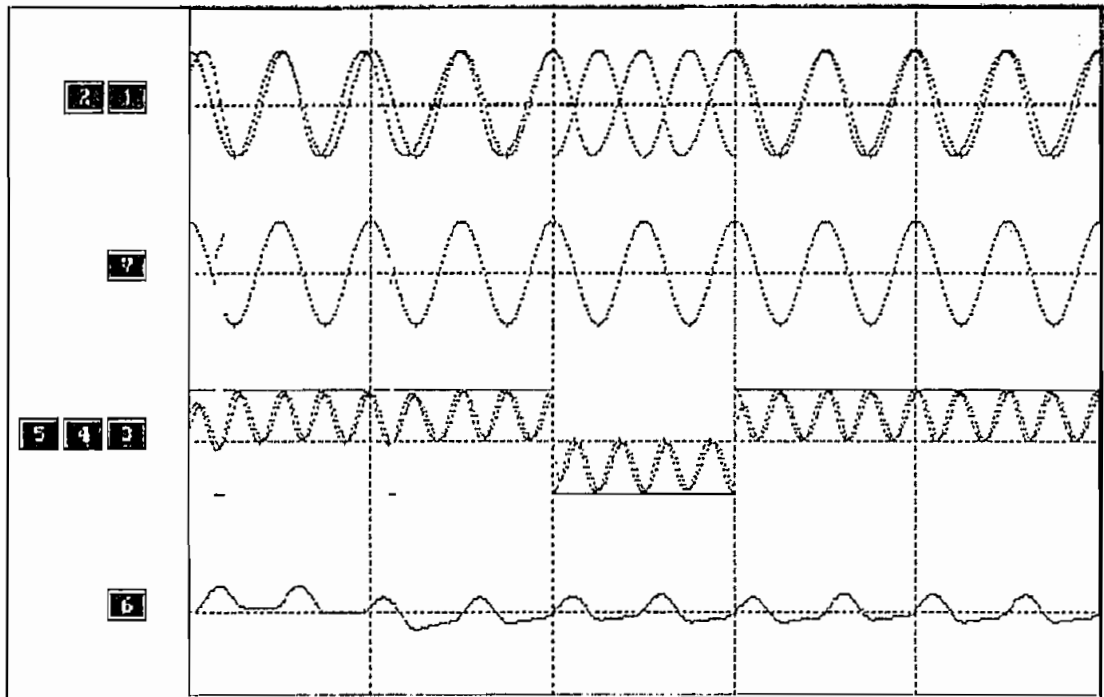


FIGURA B4.8 : Ed X1 (E)

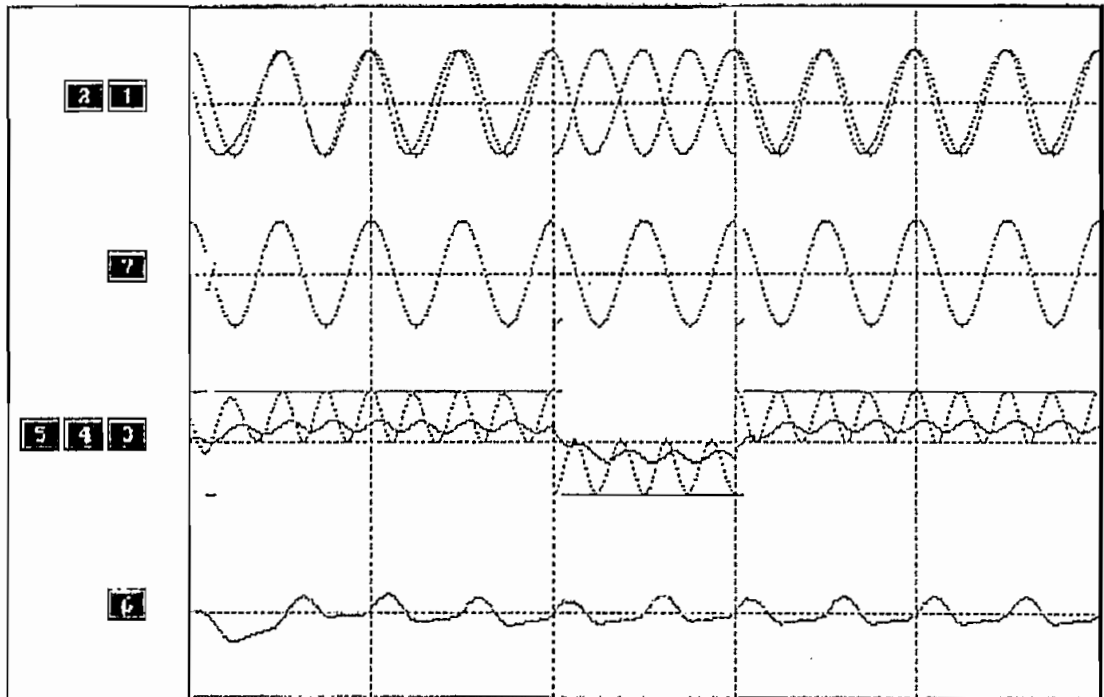


FIGURA B4.9 : Eb X1 (B)



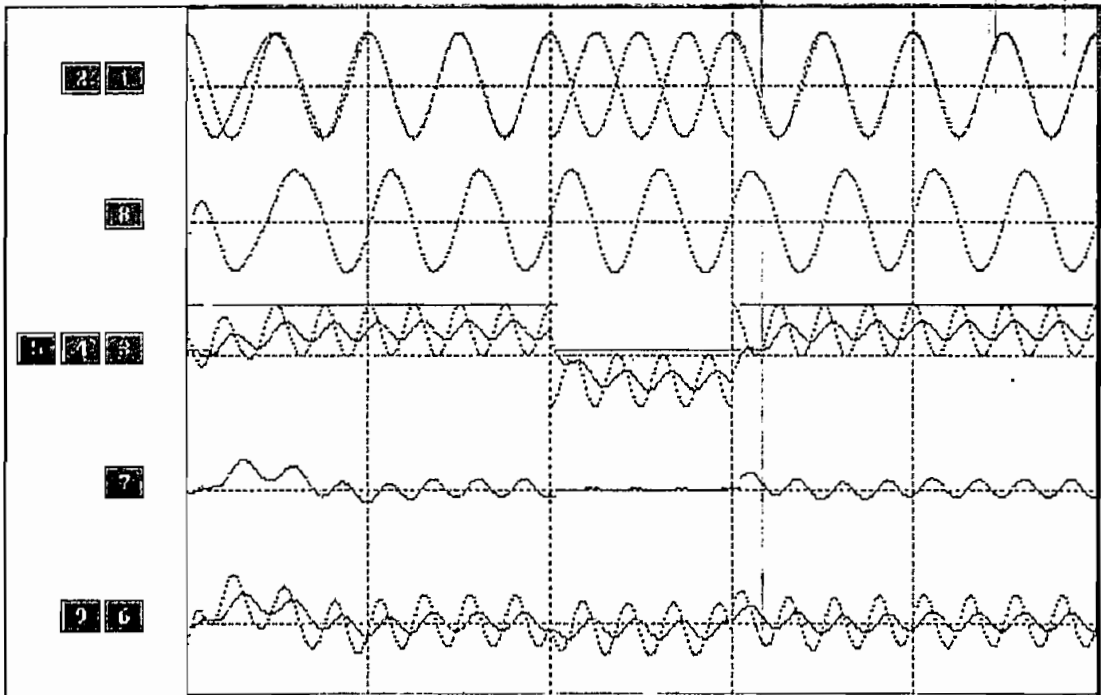


FIGURA B5.1 : Ea Ea X1 (B)

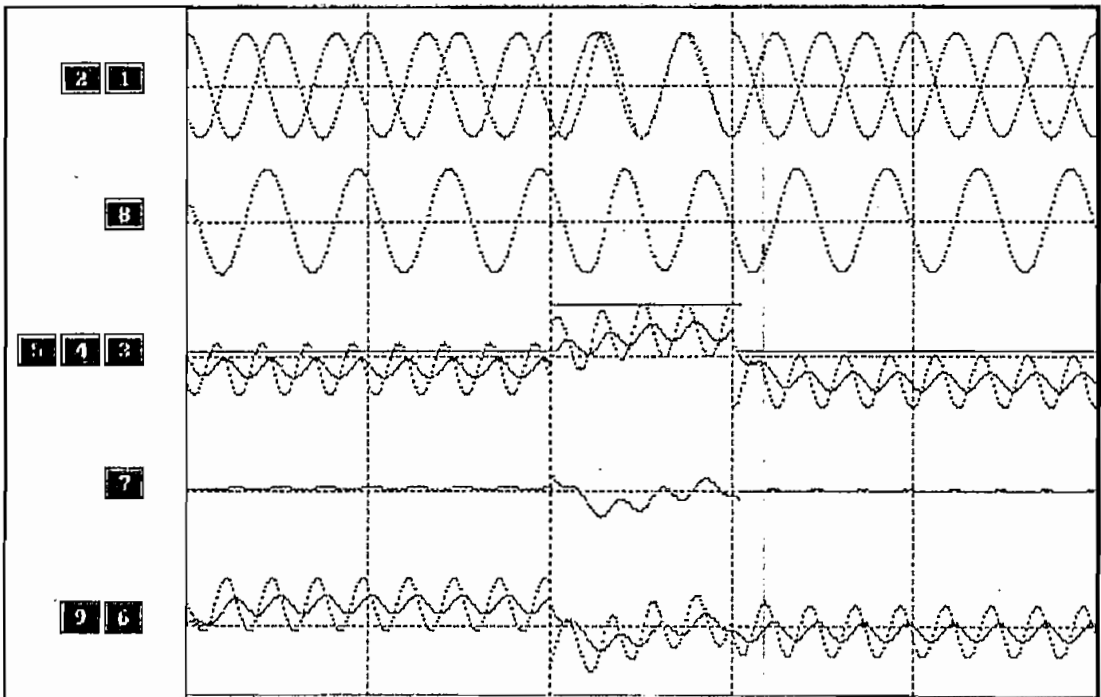


FIGURA B5.2 : Ea Ea X1 (C)

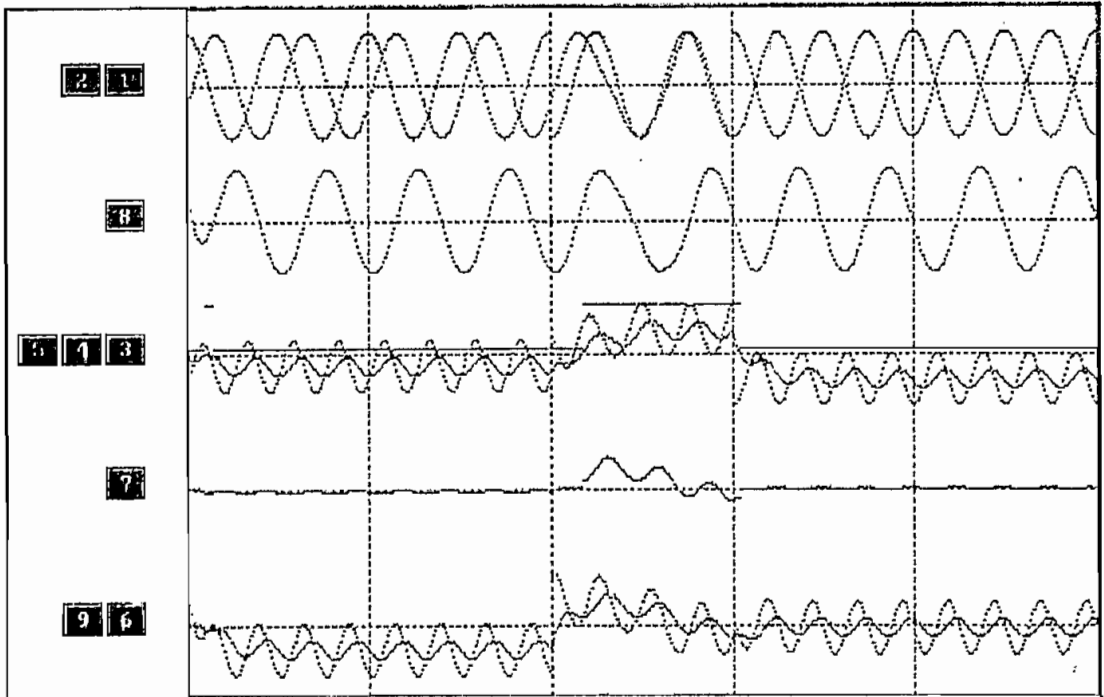


FIGURA B5.3 : Ea Ea X1 (D)

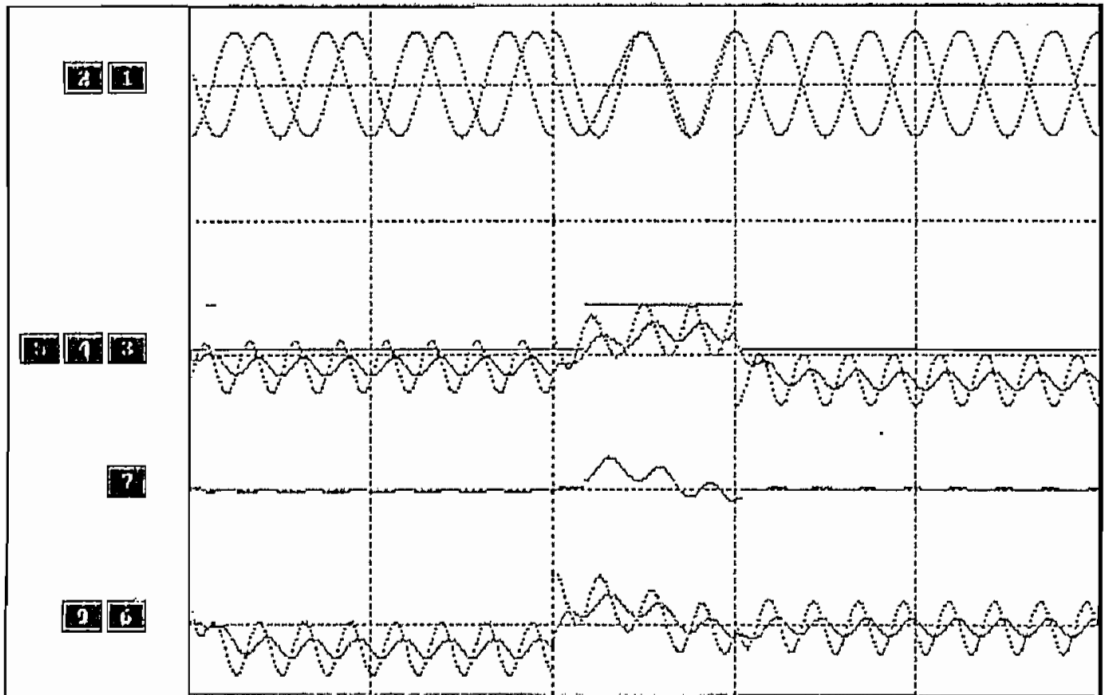


FIGURA B5.4 : Ea Ea X1 (D)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

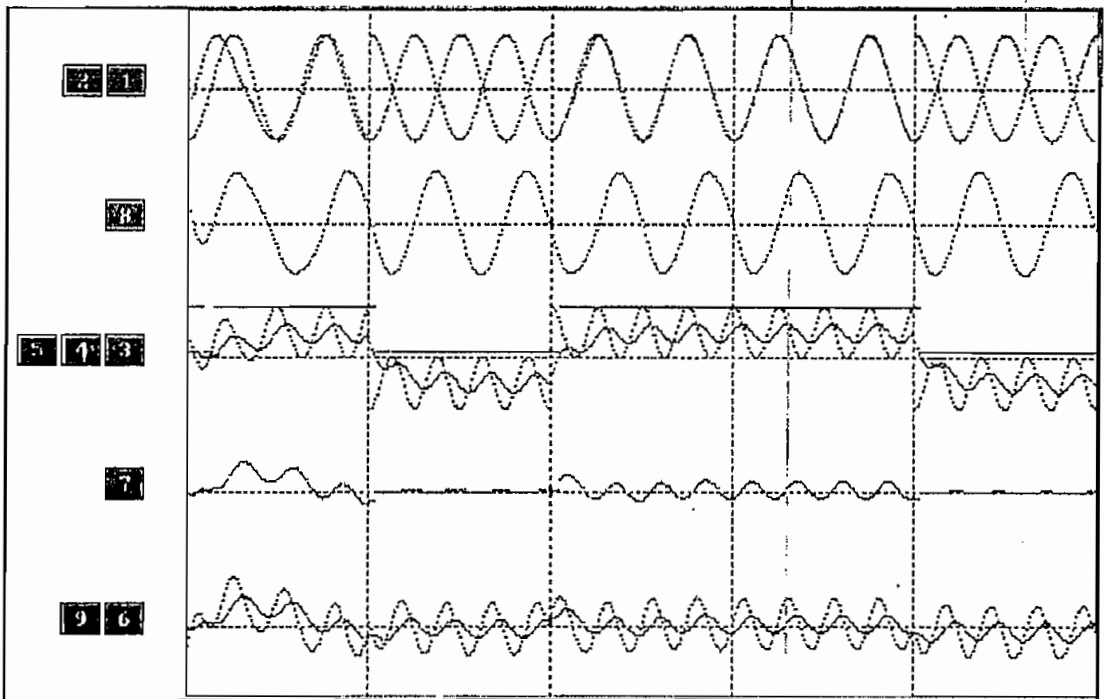


FIGURA B5.5 : Ea Ea X1 (B)

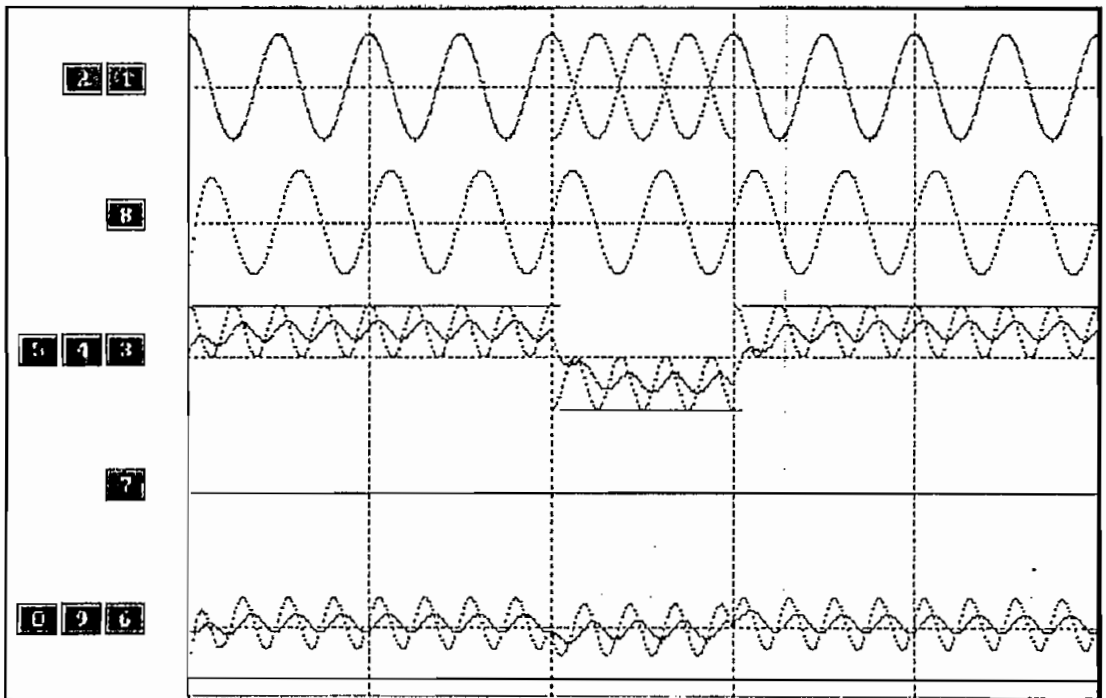


FIGURA B6.1 : Ea Ea X1 (A)

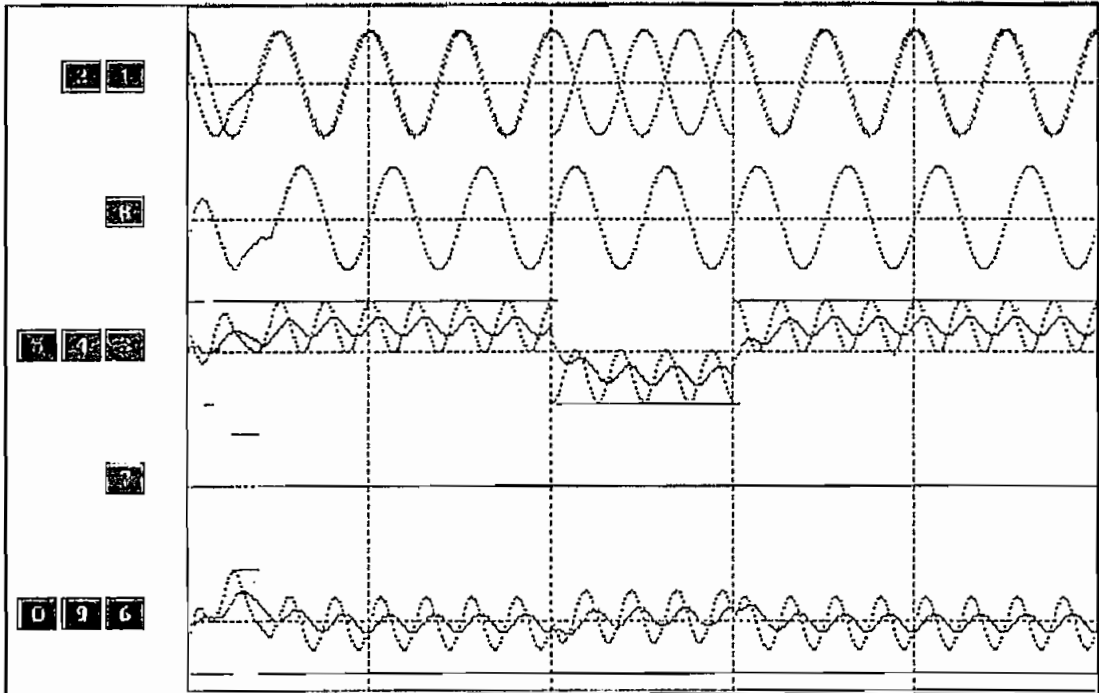


FIGURA B6.2 :  $E_a$   $E_a$  X1 (B)

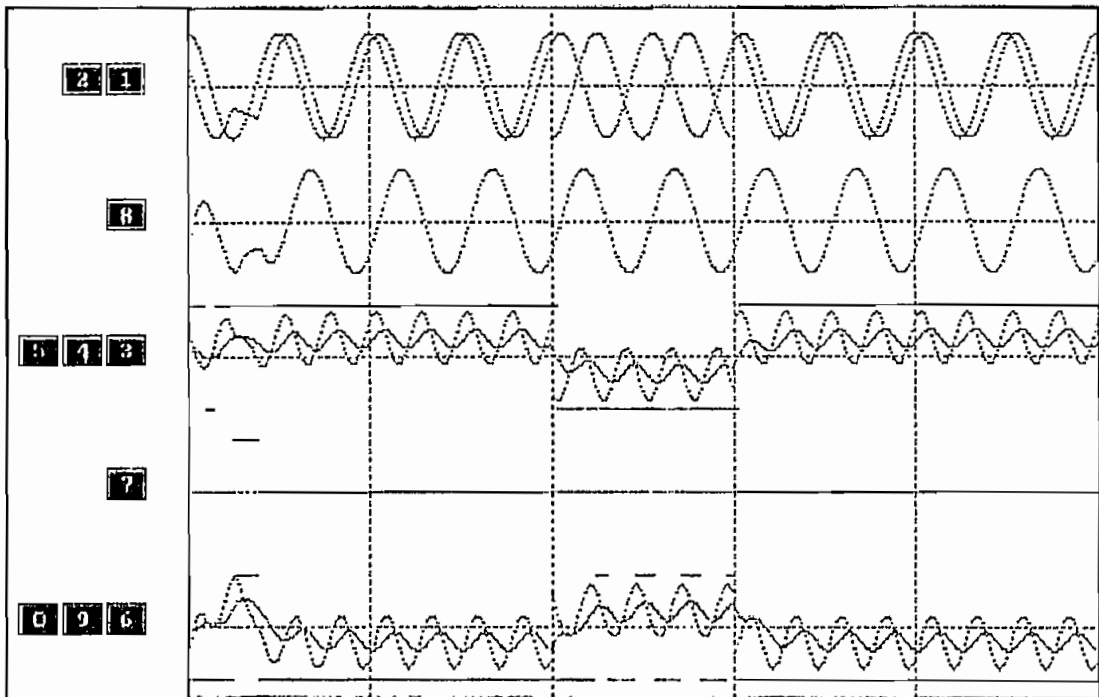


FIGURA B6.3 :  $E_a$   $E_a$  X2 (B)

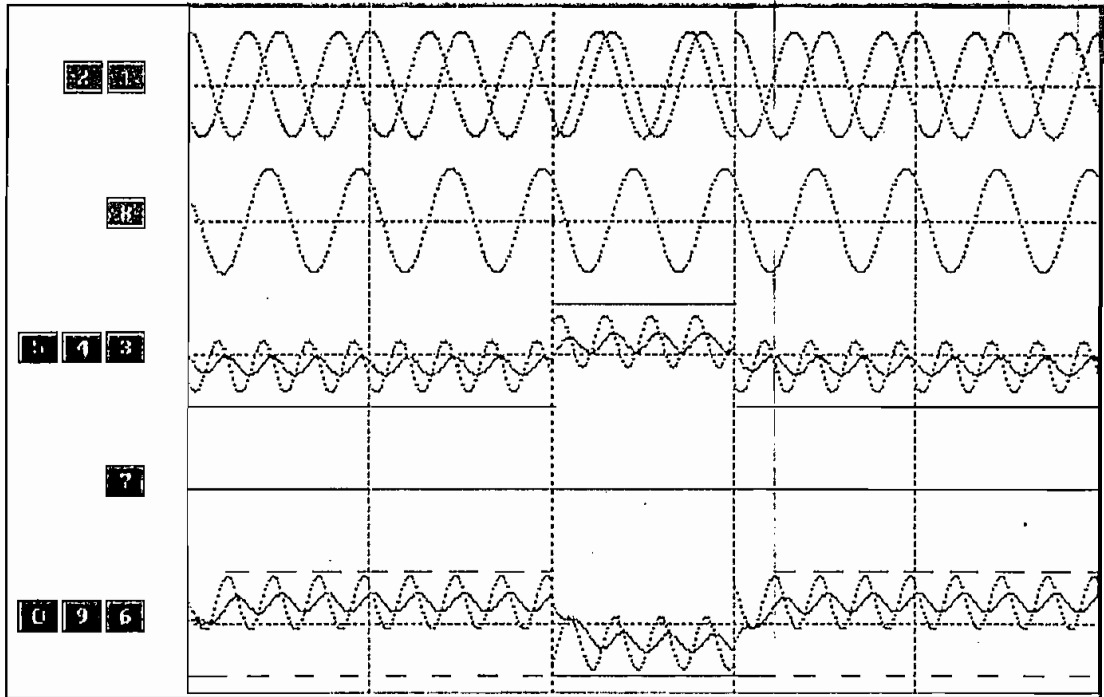


FIGURA B6.4 : Ea Ea X1 (C)

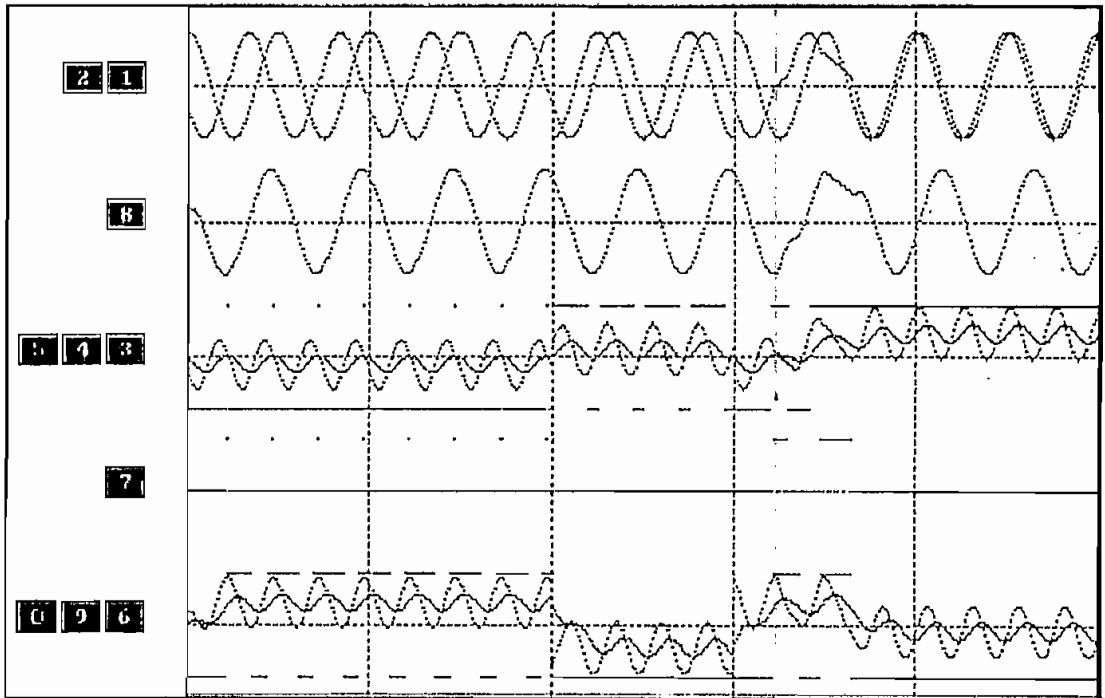


FIGURA B6.5 : Ea Ea X1 (110°)

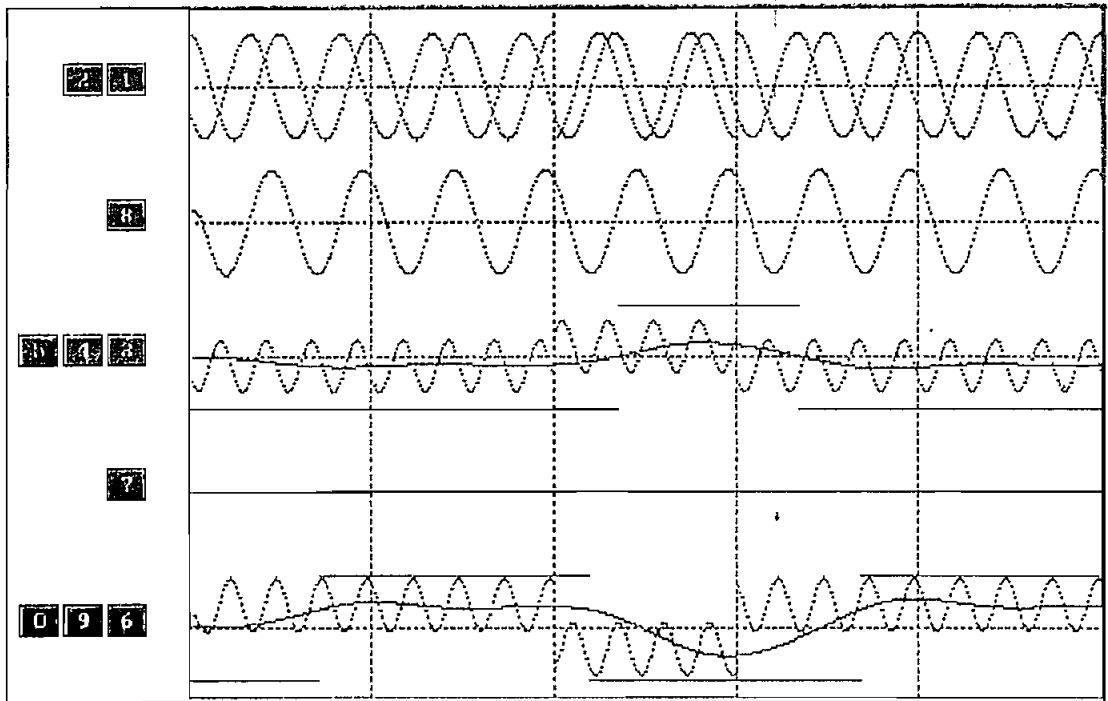


FIGURA B6.6 :  $E_c$   $E_c$   $X1$  ( $110^\circ$ )

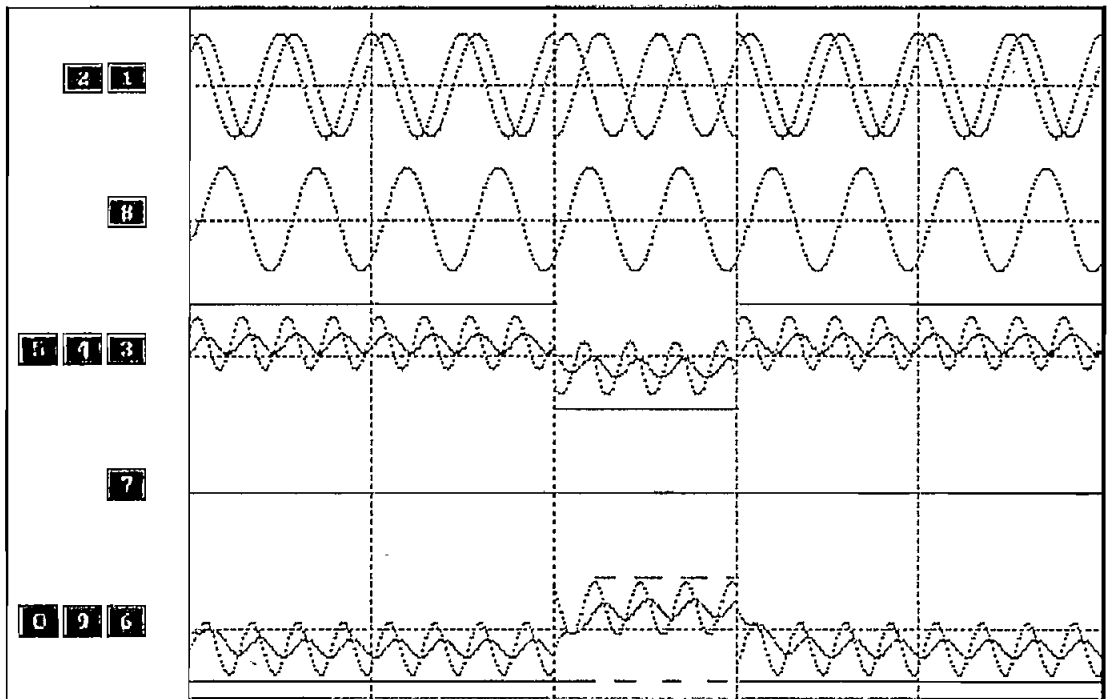


FIGURA B6.7 :  $E_a$   $E_a$   $X1$  (D)

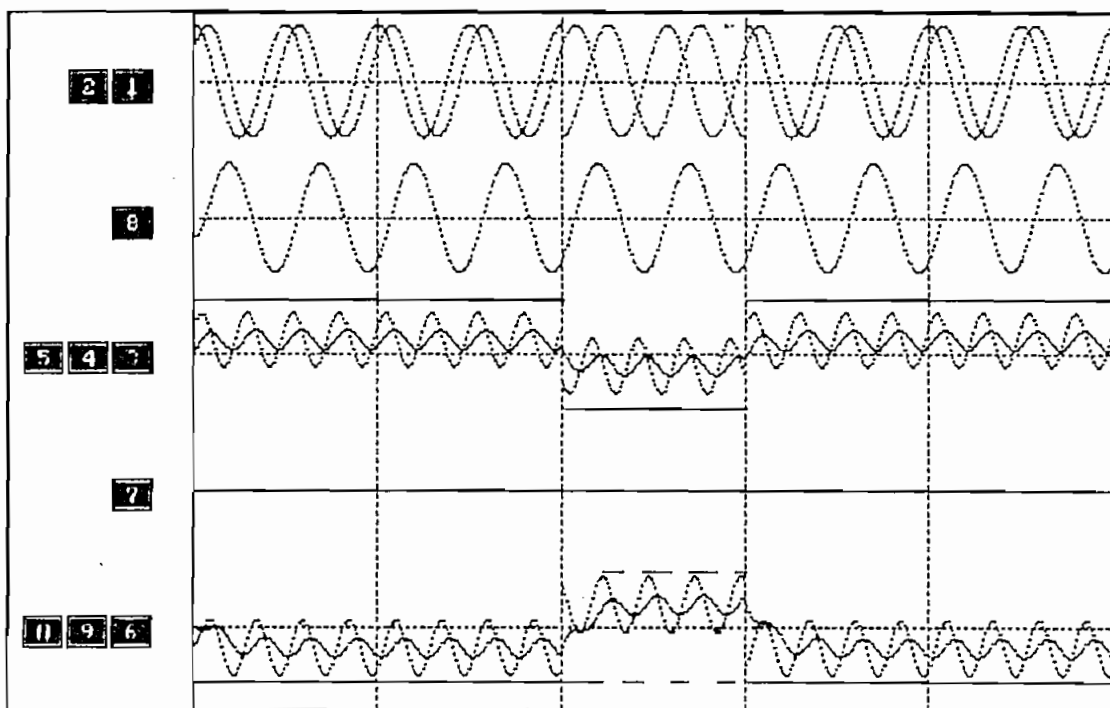


FIGURA 6.8 : Ea Ea X1 (C)

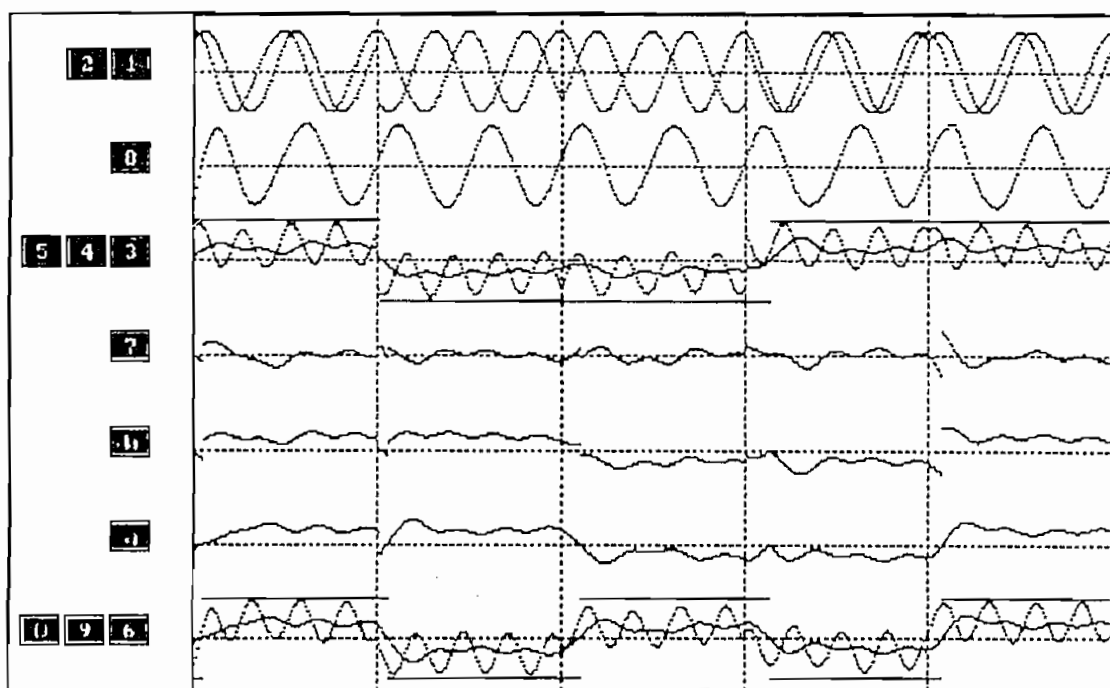


FIGURA B7.1 : Eb Eb X1 (A)

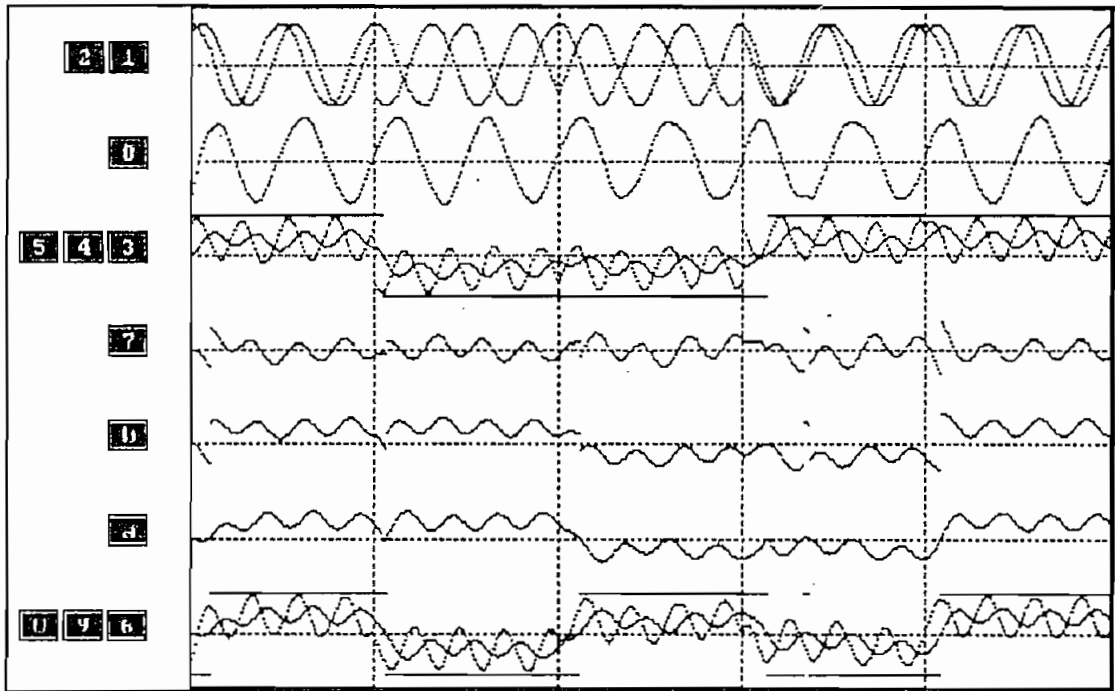


FIGURA B7.2 : Ba Ba X1 (A)

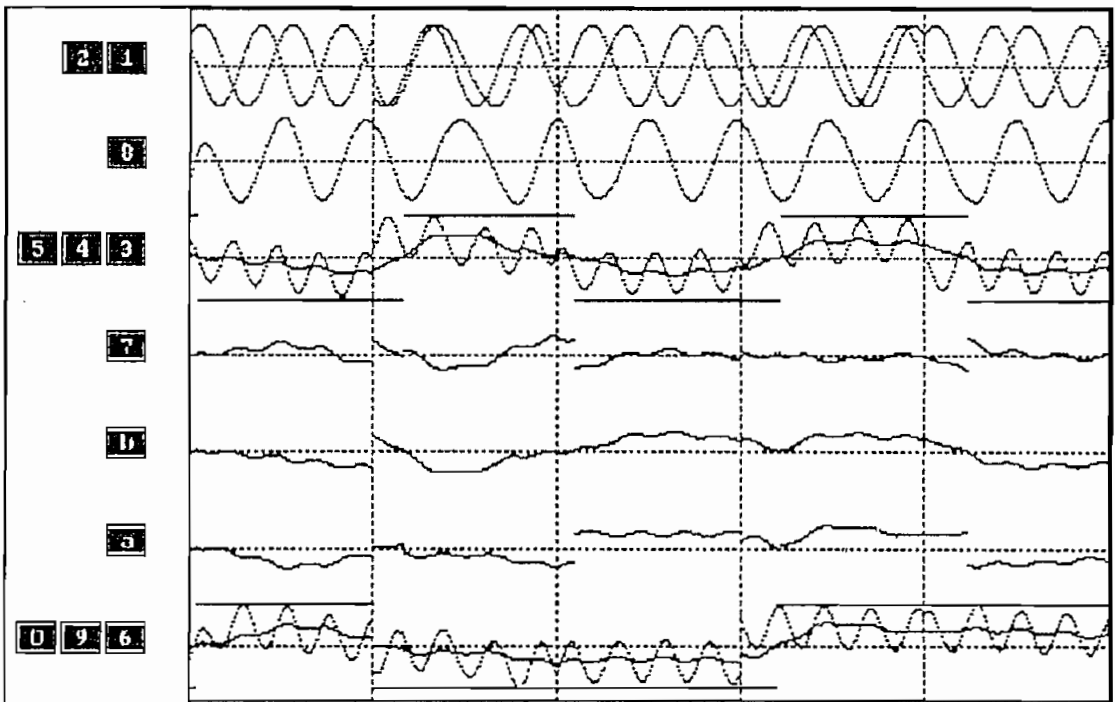


FIGURA B7.3 : Eb Eb X1 (B)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)



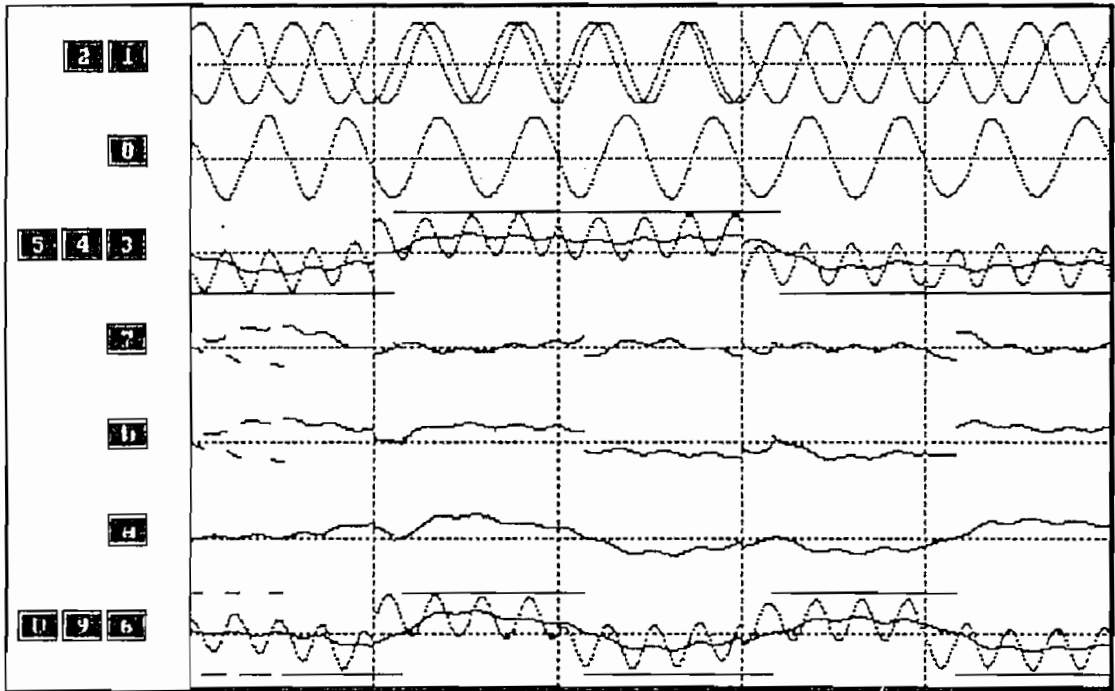


FIGURA B7.4 : Eb Eb X1 (C)

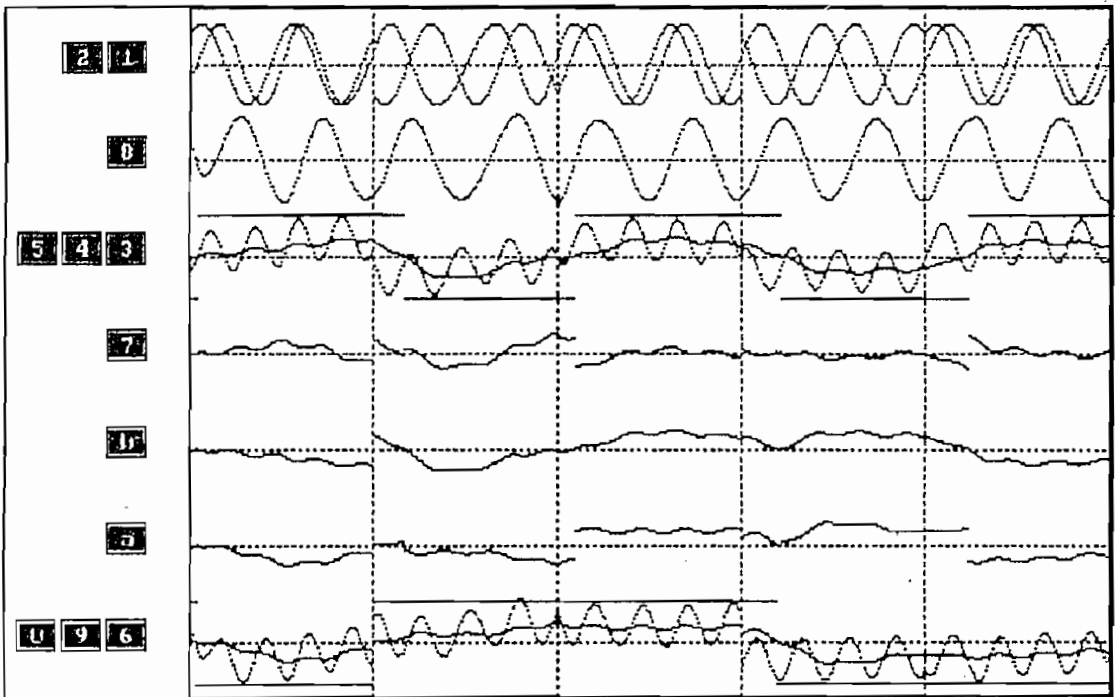


FIGURA B7.5 : Eb Eb X1 (D)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

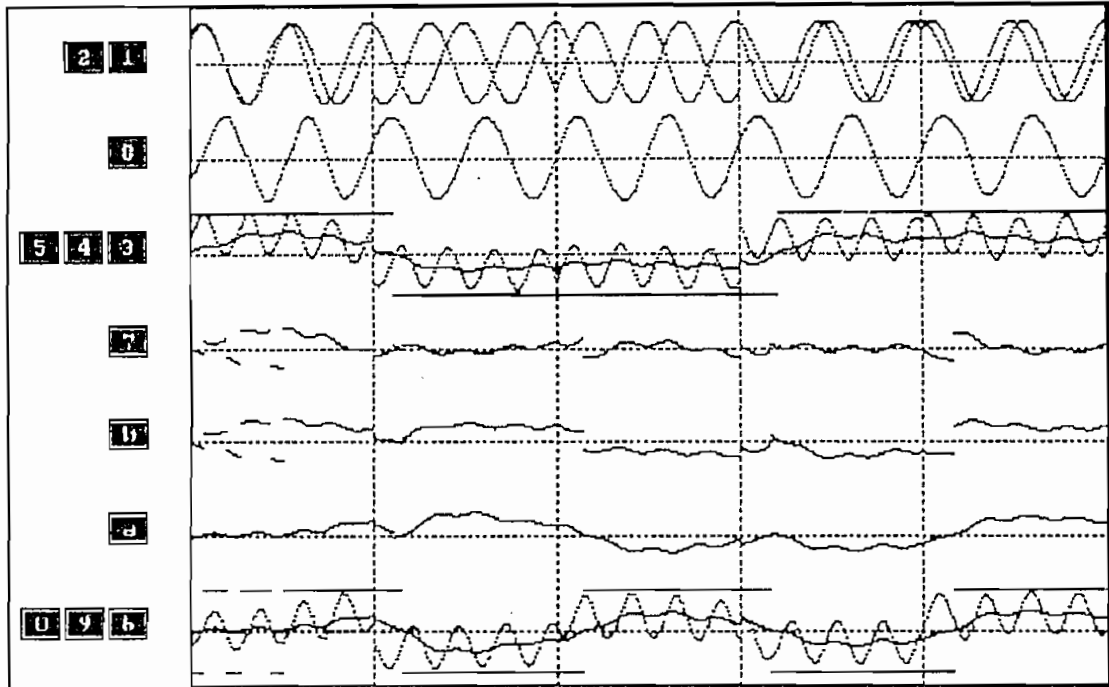


FIGURA B7.6 : Eb Eb X1 (E)

ASIGNACION DE CURVAS: VER: Diagrama de Bloques (CAPITULO II)  
y/o SIMBOLOGIA (Página 285)

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

## CAPITULO VI

## CONCLUSIONES Y RECOMENDACIONES

- En el bucle con Multiplicación de Frecuencia, aún cuando el amarre de fase se hace a doble frecuencia de la señal recibida; el enganche se nota lento cuando se tiene una secuencia periódica en banda base. Se podría disminuir el tiempo de amarre de fase aumentando la sensibilidad del VCO, es decir que pequeñas variaciones de su voltaje de control la variación de frecuencia sea mayor; esto provoca que la señal del VCO varía mas rápidamente, introduciéndose en la señal de doble frecuencia variaciones de amplitud, que al pasar por el Comparador de fase producirán un error de fase comparable al que se quiere eliminar; por tanto, la opción para disminuir el tiempo de enganche no es recomendable pues puede llevar a errores en la detección.

Los filtros de los multiplicadores de frecuencia deben ser lo mas idénticos posible para que las señales entrantes al comparador de fase tengan el mismo tratamiento.

- El Bucle de costas general evita el uso de multiplicadores de frecuencia, se usa un multiplicador análogo como comparador de fase; es decir el sistema se simplifica si lo comparamos con el Bucle con Multiplicación de Frecuencia.

- El bucle de Costas con detección dentro de un brazo simplifica aún más el sistema de recuperación de

sincronismo, pues al usar un ecualizador en el brazo en fase, se puede reemplazar el multiplicador analógico (cuatro cuadrantes) por un switch controlado o un multiplicador de dos cuadrantes.

- El Bucle de Costas numérico al tener ecualizadores tanto en el brazo en fase como en el de cuadratura permite eliminar el componente analógico que se usaba en los bucles de costas anteriores, utilizando en su lugar una compuerta AND. Pero limita la posibilidad de recuperar el sincronismo, pues se dan cambios en el VCO, únicamente si el error de fase inicial ( $\phi$ ) esta dentro del rango  $270^\circ < \phi < 360^\circ$ . Como se nota se ha simplificado los sistemas pero a cambio de limitar las posibilidades de recuperación de sincronismo.

- En el Bucle de Remodulación por el oscilador local, se podría usar un filtro pasabajos entre el VCO y el comparador de fase para mejorar el comportamiento del lazo, si se requiere de mayor confiabilidad del sistema, ya que la señal remodulada es exactamente igual a la señal modulada recibida excepto en el intervalo de tiempo que se origina desde que se da el cambio de fase de la señal recibida, el demodulador cambia el nivel del dato y éste varía la señal modulada. Como la duración de este intervalo es muy pequeña, la integración efectuada por el filtro pasabajos, eliminará prácticamente su influencia sobre el VCO. Además se eliminaría picos de amplitud que da el comparador de fase

que podría traer problemas a la respuesta del VCO, si su sensibilidad es alta.

- El bucle con remodulación por la señal recibida para iguales condiciones tiene un tiempo de enganche menor que el otro bucle con remodulación, ya que al comparar las fases de la señal recibida y la remodulada con la misma señal; la respuesta del comparador de fase es mas rápida, pues la comparación es directa, no así en el bucle con remodulación por el VCO. Se prefiere el bucle con remodulación por el VCO, si la señal recibida que se espera es ruidosa.

-Es recomendable que la frecuencia de portadora usada para modular las señales PSK, sean igual o múltiplo entero que el ritmo de transmisión; de este modo, la recuperación de sincronismo se realiza únicamente en el inicio de la transmisión, de lo contrario el problema del sincronismo se da en cada cambio de fase en la señal modulada.

- En base a la teoría de filtrado y retardos controlados dados, se puede emprender un proyecto de generación de fases usando procesamiento digital de señales. Para este propósito los diagramas de flujo en el dominio Z dados en el capítulo III, servirán para dar buen uso al procesador y sus memorias.

- Debido a la extensión del tema, el tratamiento en detalle de las señales multinivel PSK se deja para estudios posteriores.

## ANEXOS

TEMA	PAGINA
ANEXO A: Funciones Elípticas	133
ANEXO B: Diseño de filtros pasabajos.	135
ANEXO C: Manual del Usuario	140
ANEXO D: Programas	151

## FUNCIONES ELIPTICAS.

## Representación en series:

Las funciones elípticas al igual que otras funciones pueden ser representadas en términos de series:

$$\operatorname{sn}(Z, K) = \frac{1}{\sqrt{K}} \frac{\theta_1\left(\frac{Z}{2K}, q\right)}{\theta_0\left(\frac{Z}{2k}, q\right)} \quad (\text{A.1})$$

$$\operatorname{cn}(Z, K) = \sqrt{\frac{K^1}{k}} \frac{\theta_2\left(\frac{Z}{2K}, q\right)}{\theta_0\left(\frac{Z}{2k}, q\right)} \quad (\text{A.2})$$

$$\operatorname{dn}(Z, K) = \sqrt{K^1} \frac{\theta_3\left(\frac{Z}{2K}, q\right)}{\theta_0\left(\frac{Z}{2k}, q\right)} \quad (\text{A.3})$$

La constante  $q$  está dada por:

$$q = e^{-\pi \frac{K^1}{k}} \quad (\text{A.4})$$

Donde  $\theta_0(Z/2k, q)$  a  $\theta_3(Z/2K, q)$  son llamadas funciones theta y están definidas como:

$$\theta_0\left(\frac{Z}{2k}, q\right) = 1 + 2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cos\left(2m \frac{\pi Z}{2K}\right) \quad (\text{A.5})$$

$$\theta_1\left(\frac{Z}{2k}, q\right) = 2 q^{\frac{1}{4}} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \operatorname{sen}\left[(2m+1) \frac{\pi Z}{2k}\right] \quad (\text{A.6})$$

$$\theta_2\left(\frac{Z}{2K}, q\right) = 2 q^{\frac{1}{4}} \sum_{m=0}^{\infty} q^{m(m+1)} \cos\left[(2m+1) \frac{\pi Z}{2K}\right] \quad (\text{A.7})$$



$$\theta_3\left(\frac{Z}{2K}, q\right) = 1 + 2 \sum_{m=1}^{\infty} q^{m^2} \cos\left(2m \frac{\pi Z}{2K}\right) \quad (\text{A.8})$$

### SOLUCION DE LA FUNCION ELIPTICA DEL DEFASADOR DE 90 GRADOS

Se propone resolver la ecuación:

$$p_i = -\tan(\pi f l) \frac{\text{sn}[(4i+1) K'(k) / 2N, k]}{\text{cn}[(4i+1) K'(k) / 2N, k]} \quad (\text{A.10})$$

Reemplazando (A.5) y (A.6) en (A.1) y (A.7) y (A.5) en (A.2) se tiene:

$$\frac{\text{sn}(z, k)}{\text{cn}(z, k)} = \frac{2 q^{\frac{1}{4}} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \text{sen}\left[(2m+1) \frac{\pi Z}{2K}\right]}{2 q^{\frac{1}{4}} \sum_{m=0}^{\infty} q^{m(m+1)} \cos\left[(2m+1) \frac{\pi Z}{2K}\right]} \quad (\text{A.10})$$

Sabiendo que :

$$z = \frac{(4i+1) K(\sqrt{1-k^2})}{2N} \quad (\text{A.11})$$

Donde  $i = 0, 1, \dots, N-1$

(A.10) es la función que se utiliza para determinar los polos y ceros de la función de transferencia del defasador.

## ANEXO B

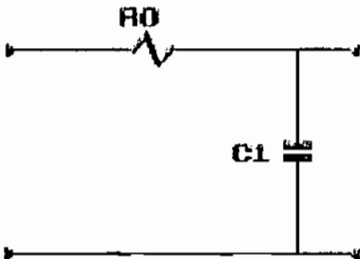
## CONSTRUCCION DE FILTROS PASABAJOS

Una vez determinada la función de transferencia del filtro que se desea construir, en forma de factores de primero y de segundo grado, se puede realizar dicho filtro colocando en cascada tantos filtros elementales de primero y segundo orden como indique el orden del filtro.

Las funciones de transferencia de los filtros están expresados en magnitudes normalizadas, tanto en impedancia (R y C), como en frecuencia ( $\Omega_n$ ). Se toma como unidad de frecuencia el valor de la frecuencia central  $f_0$  en los filtros pasabanda y la frecuencia  $f_p$  o  $f_r$  en los pasabajos. Los valores reales de los elementos se obtienen sustituyendo  $R_0$  por un valor cualquiera expresado en  $[\Omega]$  y sustituyendo  $\Omega_n$  por  $2\pi f_p$ ,  $2\pi f_r$ , o  $2\pi f_0$  según sea el caso.

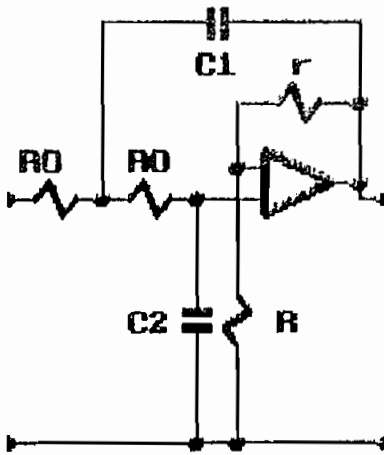
Los elementos básicos que pueden ponerse en cascada para estructurar un filtro de un orden dado son:

## Elemento 1



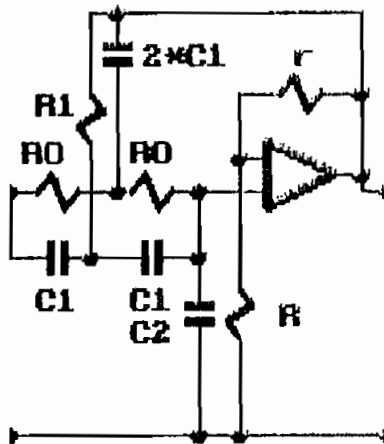
$$F(s) = \frac{1}{ms + 1} \quad (B.1)$$

Elemento 2



$$F(s) = \frac{K}{m q s^2 + [2 m + q (1-K)] s + 1} \quad (B.2)$$

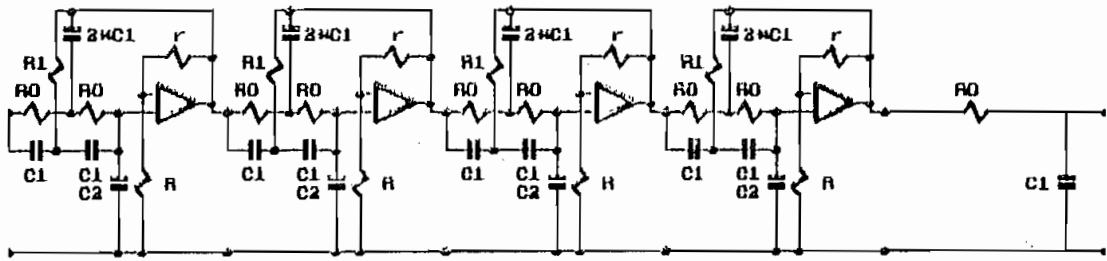
Elemento 3



$$F(s) = \frac{K (1 + q^2 s^2)}{(q^2 + 2mq) s^2 + 2 [m+2q(1-K)] s + 1} \quad (B.3)$$

## CONSTRUCCION DE FILTROS PASABAJOS ELIPTICOS.

La construcción se fundamenta en la función de transferencia obtenida, ecuación (3.38). De acuerdo a ésta, se obtiene un filtro de orden dado poniendo en cascada los elementos básicos, 3 y 1 :



Filtro Elíptico Pasabajos de noveno orden (N=9)

**ELEMENTO 3 :** Relacionando las funciones (B.3) y (3.38), se determina los parámetros normalizados que nos dan el valor de los componentes del elemento

$$q = \sqrt{\frac{1}{A}} \quad (\text{B.4})$$

$$m = \frac{\frac{1}{C} - q^2}{2q} \quad (\text{B.5})$$

$$K = 1 - \frac{B - 2Cm}{qC} \quad (\text{B.6})$$

Dado  $R_0$  se puede calcular los valores de  $C_0$ ,  $C_1$  y  $C_2$ .

$$C_0 = \frac{1}{R_0 \Omega_u} \quad (\text{B.7})$$

$$C_1 = q C_0 \quad (\text{B.8})$$

$$C_2 = m C_0 \quad (\text{B.9})$$

Los valores de R y r se determinan de acuerdo al valor de K

$$K = 1 + \frac{r}{R} \quad (\text{B.10})$$

Asumiendo R, se determina r, aunque se recomienda que r sea un potenciómetro de ajuste de ganancia.

ELEMENTO 1 : Relacionando las funciones de transferencia (3.38) y (B.1) :

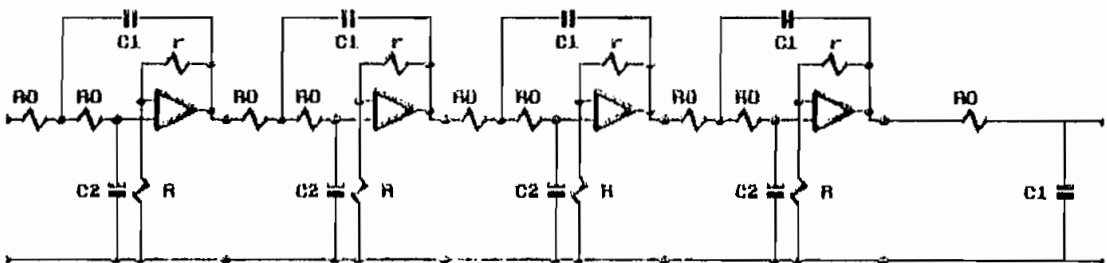
$$m = \frac{1}{\sigma} \quad (\text{B.11})$$

Dado R se puede conseguir C1.

$$C_1 = \frac{1}{\Omega_c m R} \quad (\text{B.12})$$

### CONSTRUCCION DE FILTROS PASABAJOS DE BUTTERWORTH

La función de transferencia del filtro (3.34), en sus componentes de segundo orden, se adapta al elemento 2, mientras que D(s) al elemento 1 que ya fue enlazado en la construcción de Filtros pasabajos Elíptico, en el que  $\sigma$  del filtro elíptico debe ser cambiado por  $P_r$  de (3.34).



Filtro Pasabajos de Butterworth de noveno orden (N=9)

ELEMENTO 2 : Comparando las funciones de transferencia (3.34) y (B.2), se determina los parámetros normalizados del elemento.

$$K = \frac{H_0}{C_1} \quad (B.13)$$

$$m = \frac{b_1}{4 C_1} \pm \frac{\sqrt{b_1^2 - 8 C_1 (1 - k)}}{4 C_1} \quad (B.14)$$

Donde m es real y positivo.

$$q = \frac{1}{C_1 m} \quad (B.15)$$

Para determinar los valores de  $C_0$ ,  $C_1$  y  $C_2$  usando las ecuaciones (B.7), (B.8) y (B.9) respectivamente, se necesita escoger la impedancia  $R_0$ . Se precisa que  $R_0$  sea mucho menor que la impedancia de entrada de los amplificadores y mucho más elevada que su impedancia de salida. Por lo que se recomienda tomar  $R_0 = 10 \text{ k}\Omega$ .

Los valores de  $R$  y  $r$  se determinan igual que en el caso anterior.

**MANUAL DEL USUARIO**

En este anexo se da una breve descripción de las opciones del programa PSK (Versión 1.0), así como también se analizan las características de los componentes de los bucles a ser simulados.

**Hardware.-** El programa necesita del siguiente equipo para operar:

- Requiere un mínimo de 340 Kbytes de RAM
- El co-procesador matemático 8087 es opcional, sin embargo su presencia hace que aumente la rapidez en la ejecución de operaciones matemáticas como multiplicaciones, funciones trigonométricas, etc.
- Monitor a Color o Monocromático es opcional, aunque en monocromático se limita la visualización de resultados.

**Software.-** El programa PSK, está escrito en lenguaje C y requiere del siguiente software para operar.

- DOS(Disk Operating System) de Microsoft, versión 4.0 o mas avanzado.
- Los módulos PSK.EXE, ENTRADA.EXE, SIMULADOR.EXE y OPCIONES.EXE.
- Los drivers gráficos de acuerdo al monitor que se disponga:

ATT.BGI, EGAVGA.BGI, HERC.BGI, IBM8514.BGI, PC3270.BGI

- Las librerías del C: LITT.CHR, SANS.CHR, TRIP.CHR.

El programa se inicializa ejecutando el programa raíz PSK.EXE.

## MENU PRINCIPAL

DOS.- Presenta opciones del Sistema como IMPRIMIR pantalla, salir al DOS.

BUCLES.-Presenta como opciones los diferentes bucles de recuperación de sincronismo a ser simulados.

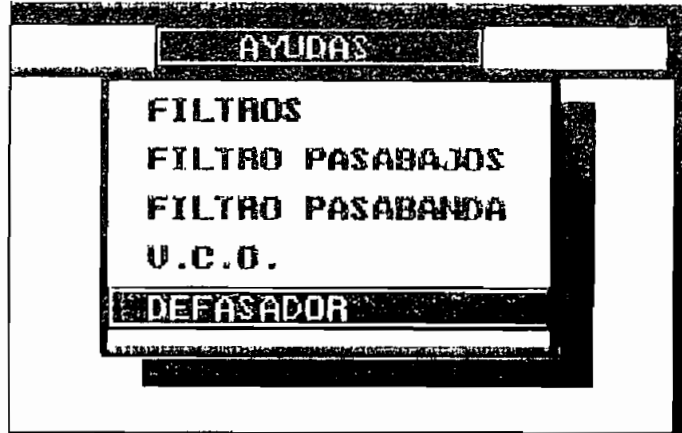
BUCLES	FILTROS	AYU
Con MULTIPLICACION		Alt M
Costas General		Alt G
REMODULACION por V.C.O.		Alt O
REMODULACION por onda RECIBIDA		Alt R
Costas con deteccion en un brazo		Alt D
Costas NUMERICO		Alt N
Costas para QPSK		Alt Q
Phaselocked loop (P.L.L.)		Alt P

FILTROS .- Presenta las opciones para determinar los elementos de un filtro pasabajos activos ELIPTICO o de BUTTERWORTH, los que pueden ser construidos fácilmente en base Amplificadores Operacionales, resistencias y capacitores.

FILTROS	AYUDAS
Pasabajos ELIPTICO	Alt L
Pasabajos BUTTERWORTH	Alt B



AYUDAS - Presta la ayuda necesaria al usuario para trabajar, dentro del programa sin dificultad. Sus opciones:

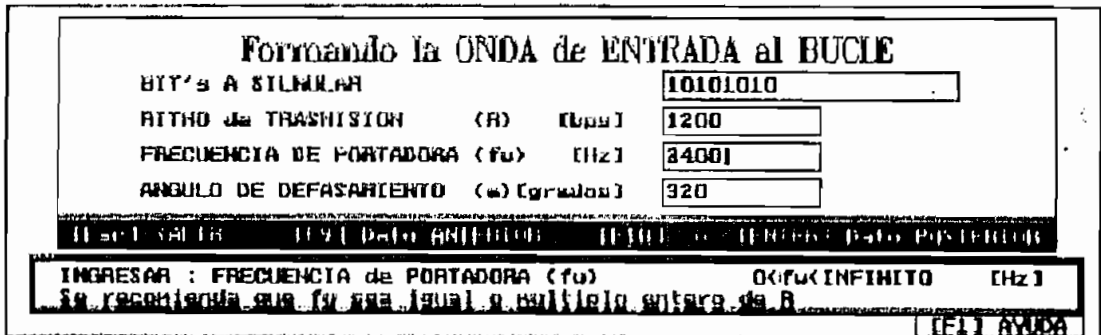


MENU SECUNDARIO

Una vez determinado el bucle que se desea analizar, se procede a estructurar cada elemento del bucle.



ESTRUCTURACION DE LA ONDA DE ENTRADA AL BUCLE



**BITS A SIMULAR.**- Se solicita el ingreso del tren de bits a ser modulados con la técnica PSK, que constituye la señal de entrada al bucle que se desea analizar el comportamiento.

El tren de bits pertenece a un sistema binario, es decir se acepta como datos válidos únicamente un valor 0 y un valor 1, que corresponden a los valores lógicos 0 y 1 respectivamente.

**RITMO DE TRANSMISION (R)** .- Es la velocidad binaria del tren de bits que se ha ingresado anteriormente.

**FRECUENCIA DE PORTADORA (fw)**.- Es la frecuencia de la señal analógica con la que se modula el tren de bits que tiene la velocidad binaria establecidos anteriormente.

**ERROR DE FASE INICIAL (e)**.- Es el ángulo de error inicial que existe entre las señales recibida por el bucle y la del oscilador controlado por voltaje creada en el bucle.

El valor del dato debe estar entre 0 y 360 GRADOS.

#### ESTRUCTURACION DEL FILTRO PASABAJOS

Estructurando Filtro Pasabajos 1 [ F.P.B.1 ]			
QUE FILTRO?		<input type="checkbox"/> ELIPTICO	<input checked="" type="checkbox"/> BUTTERWORTH [ e ]
FREC. de CORTE (fc) [Hz]	<input type="text" value="1000"/>	ATENUACION en fc (Afp) [dB]	<input type="text" value="1"/>
FREC. de SUPRESION (fr) [Hz]	<input type="text" value="1500"/>	ATENUACION en fr (Afr) [dB]	<input type="text" value="15"/>
ORDEN DEL FILTRO (N)	<input type="text" value="1.97"/>		<input type="text" value="21"/>
INGRESAR : ORDEN del FILTRO (N) VALOR RECOMENDADO < N < 11.			

Para la simulación del filtro pasabajos se ha escogido dos modelos obtenidos en base a aproximaciones matemáticas. Se ha escogido estos debido a su comportamiento muy marcado sobre otras aproximaciones tanto en régimen transitorio como en régimen permanente.

Filtro Elíptico.-permite obtener un corte extremadamente, es decir, una pendiente de corte muy buena para un orden de filtro dado. Además se dispone de ceros de transmisión, con los que se puede eliminar frecuencias no deseadas, ya que, a la frecuencia del cero la atenuación toma un valor INFINITO. Pero presenta deformaciones muy grandes en régimen transitorio.

Filtro de Butterworth.- Tiene una pendiente de corte poco abrupta, pero presenta deformaciones muy pequeñas en régimen transitorio.

Un filtro Pasabajos esta totalmente determinado si se conoce su Atenuación máxima en la banda de paso( $A_{fp}$ ), Atenuación mínima en la banda de atenuación ( $A_{fr}$ ) y su selectividad ( $k$ ), donde  $k$  expresa la pendiente mas o menos abrupta de los cortes y es la relación entre la frecuencia de corte( $f_p$ ) y la frecuencia de supresión ( $f_r$ ).

$$K = \frac{f_p}{f_r}$$

$f_p$  y  $f_r$  pueden tomar cualquier valor positivo, se considera dado en [Hz] y se debe respetar la relación  $f_p < f_r$ .

$A_{fp}$  puede tomar un valor entero típicamente menor que 30. Está dado en [dB]

$A_{fr}$  puede tomar un valor entero típicamente menor que

100. Está dado en [dB]. Debe considerarse la relación  $A_{fp} < A_{fr}$ .

**ORDEN DEL FILTRO** .- Dado las frecuencias críticas con sus respectivas atenuaciones, se hacen los cálculos necesarios para determinar el orden del filtro que se requiere para satisfacer las condiciones de frecuencia y atenuación dadas. Es claro que el orden del filtro que Ud. ingresa es entero pues no puede tener en la práctica circuitos mitad, por lo tanto, las características del filtro estructurado no serán exactamente las deseadas, se recomienda que se ingrese el entero inmediato superior a la cantidad recomendada para mejor aproximación.

El orden del filtro debe ser ENTERO y MENOR que 11. Se puede estructurar un filtro de orden INFINITO, pero para el presente trabajo es suficiente cumplir con la restricción impuesta.

#### ESTRUCTURACION DEL OSCILADOR CONTROLADO POR VOLTAJE

<b>Estructurando el Oscilador Controlado por Volteje</b>		
Frecuencia CENTRAL de Oscilacion (fo) [ Hz ]		<input type="text" value="2400"/>
SENSIBILIDAD (S) [ Hz / grado ]		<input type="text" value="20"/>
TOLERANCIA [ porcentaje (%) ]		<input type="text" value="10"/>
<small>Ud. puede ingresar cualquier número en las unidades deseadas.</small>		
<b>INGRESAR: FRECUENCIA CENTRAL de OSCILACION (fo). 0 (fo INFINITO (Hz)</b> <small>Se recomienda fo = fu. Donde fu es la frecuencia de portadora.</small>		
<input type="text" value="OK"/> <input type="text" value="AYUDA"/>		

**FRECUENCIA CENTRAL DEL OSCILADOR.** - Es la frecuencia a

la que oscila cuando el voltaje de control es cero. Esta frecuencia debe ser la misma que la utilizada para modular la onda de entrada al bucle, para que el bucle responda adecuadamente. Su valor esta dado en [Hz] y puede ser cualquier valor positivo.

**SENSIBILIDAD** .- Se dice sensibilidad del Oscilador al cambio de frecuencia por grado de error de fase.

Conocido la frecuencia central de oscilación y la sensibilidad, quedan determinadas las frecuencias superior e inferior de oscilación.

**TOLERANCIA**.- El VCO produce cambio de frecuencia en su salida únicamente si el voltaje de error de fase, supera el valor que se ingresa en porcentaje.

#### ESTRUCTURACION DEL DEFASADOR DE 90 GRADOS

Estructurando el Defasador de 90 grados	
Frecuencia INFERIOR ( f1 ) [ Hz ]	100
Frecuencia SUPERIOR ( f2 ) [ Hz ]	5000
ERROR de FASE ( EPS ) [ grados ]	3
ORDEN del FILTRO (se recomienda) > 4.258246	5
INGRESAR : Frecuencia SUPERIOR del FILTRO PASABANDA ( f2 ). SU VALOR : 5000 [ Hz ]	

**FRECUENCIA INFERIOR (f1) Y FRECUENCIA SUPERIOR (f2)** .- Son las frecuencias límites del filtro pasabanda que se forma. Lo característico de este filtro es que dentro de este rango

de frecuencias dado, todas las componentes de la señal sufren un desplazamiento de fase de 90 grados.

El valor de estos datos pueden ser cualquier número positivo, considerando lógicamente que  $f_1 < f_2$ . Este valor se toma en [Hz].

**MAXIMO ERROR DE FASE** .- Es el error de fase máximo producido por el filtro defasador que se impone. Se acepta como dato válido un valor comprendido entre 0 y 45 [grados].

**ORDEN DEL FILTRO** .- Dado las frecuencias críticas y el error máximo de fase, se realizan los cálculos necesarios para determinar el orden del filtro que se requiere para satisfacer las condiciones dadas y se lo recomienda. Es claro que el orden del filtro que Ud. ingresa es entero pues no puede tener en la práctica circuitos mitad, por lo tanto, las características del filtro estructurado no serán exactamente las deseadas, se recomienda que se ingrese el entero inmediato superior a la cantidad recomendada para mejor aproximación.

El orden del filtro debe ser ENTERO y MENOR que 6. Se puede estructurar un filtro de orden INFINITO, pero para el presente trabajo es suficiente cumplir con la restricción impuesta.

**ESTRUCTURACION DEL DUPLICADOR DE FRECUENCIA**

Estructurando Filtro Pasabanda X 2 [ A ]			
Frecuencia CENTRAL (fo) [Hz]	<input type="text" value="4000"/>	ATENUACION en fp (Afp) [dB]	<input type="text" value="1"/>
Frec. de CORTE (fp) [Hz]	<input type="text" value="4500"/>	ATENUACION en fr (Afr) [dB]	<input type="text" value="20"/>
Frec. de SUPRESION (fr) [Hz]	<input type="text" value="4000"/>	ORDEN del FILTRO > 1.9	<input type="text" value="2"/>
INGRESAR : ATENUACION en la FRECUENCIA de SUPRESION (Afr). SU VALOR : Afr < Afr < 100 dado en [dB]			

$f_p$ ,  $f_r$ ,  $A_{fp}$ ,  $A_{fr}$  y  $N$  tienen el mismo significado que en la estructuración del filtro pasabajos, pues la estructuración del filtro pasabanda se realiza a partir de un prototipo pasabajos, pero en este caso  $f_p$  y  $f_r$  no se considera con respecto a la frecuencia cero, sino con respecto a la frecuencia central del filtro pasabanda deseado ( $f_o$ ). De este modo, para esta estructuración,  $f_r < f_p$  con respecto a cero, mientras que se mantiene  $A_{fp} < A_{fr}$ .

### PANTALLAS DE PRESENTACION DE RESULTADOS

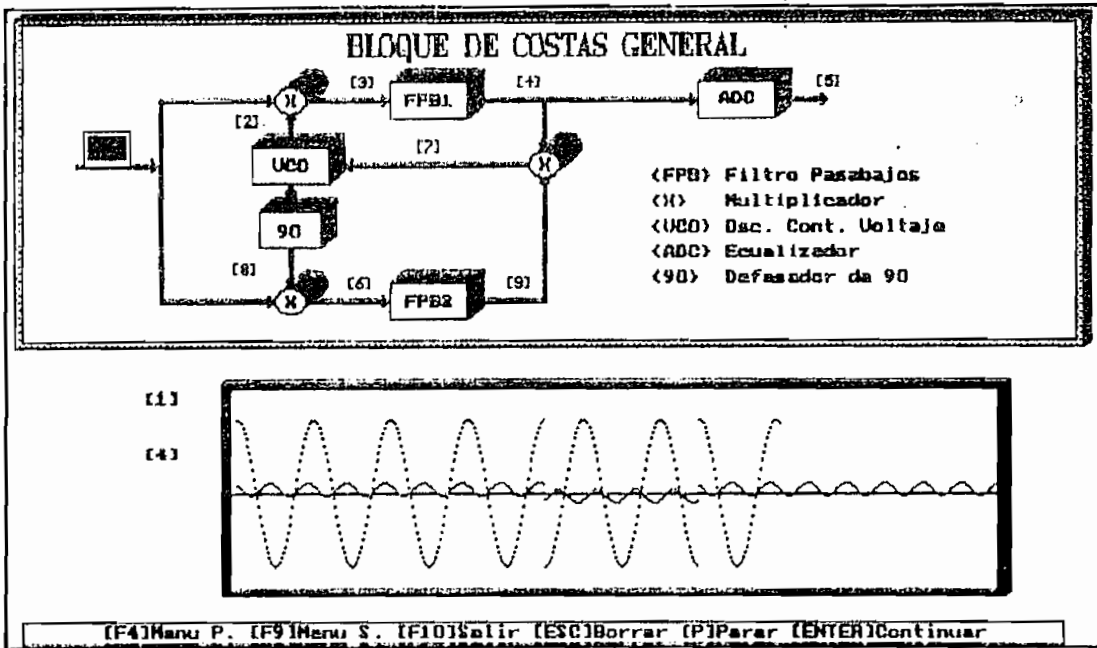
Se tiene 3 formas diferentes para presentar las ondas que producen los bucles en estudio.

#### 1.-UNA A UNA. Se activa con (F2)

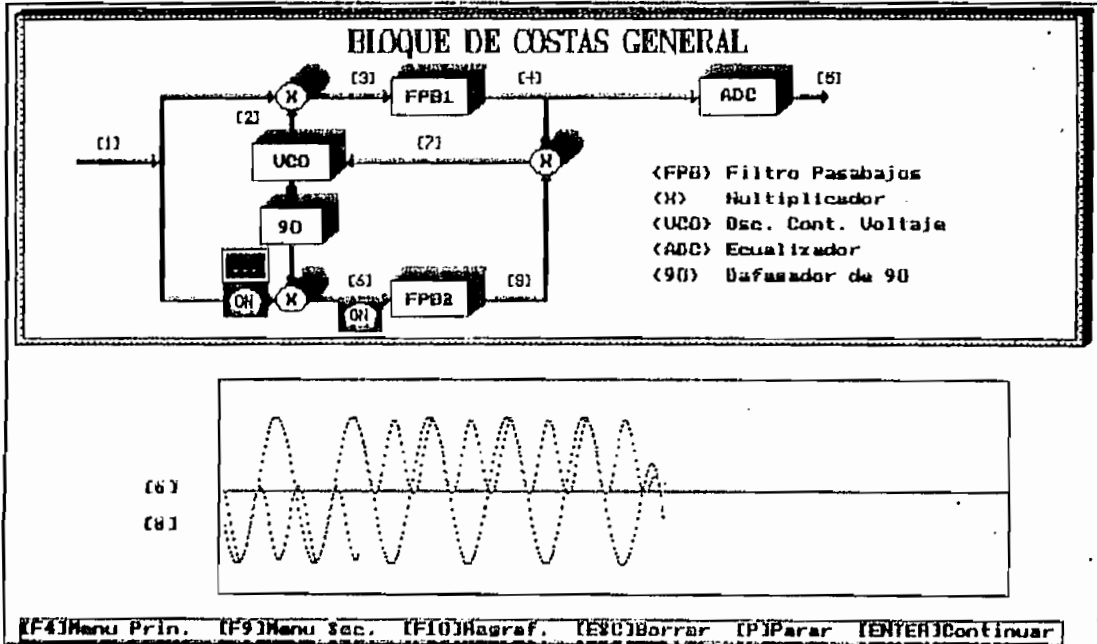
Se ha planteado pensando en el usuario que dispone de un monitor MONOCROMATICO, donde distinguir una dentro de un conjunto de ondas, es una tarea imposible a menos que se conozca previamente la onda que se espera tener como resultado.

En este modo, se gráfica las ondas una a una en un mismo eje y a la misma escala, de tal forma que, se puede

superponer en el orden que se desee varias señales.



2.- VARIAS EN UN MISMO EJE. Se activa con (F5)

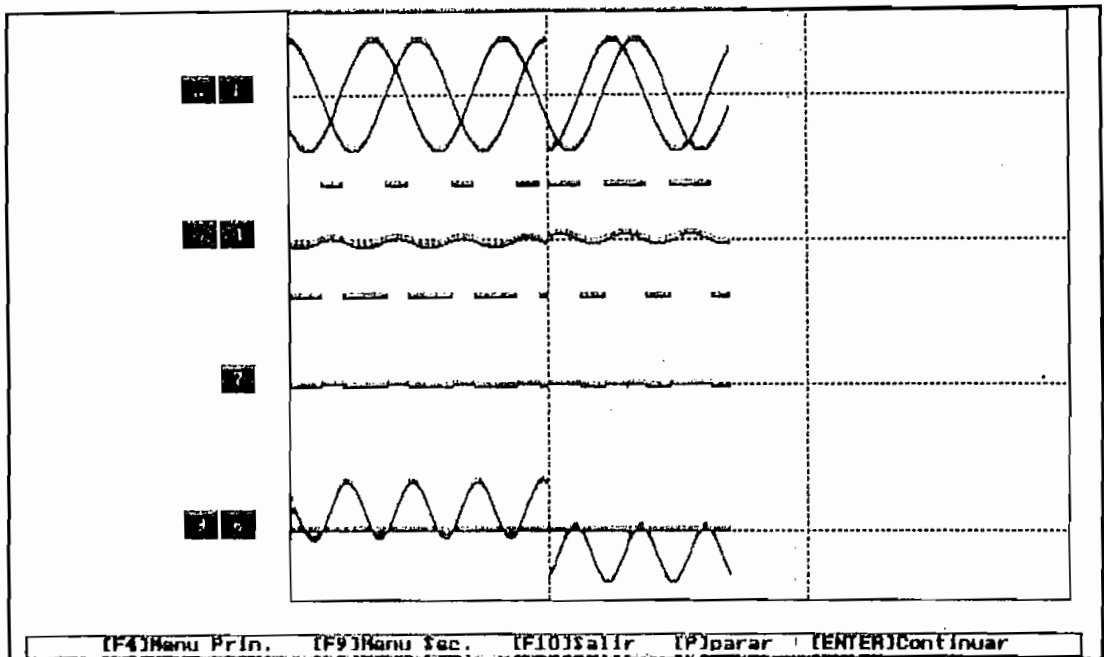


En un monitor que permite distinguir varios colores, el



identificar las respuestas se facilita, así, esta opción permite seleccionar varias ondas y graficarlas simultáneamente. Al igual que la opción anterior, las señales se presentan en un mismo eje y a la misma escala. Se hace corresponder un color al número indicador de onda graficada y a la señal misma.

3.-VARIAS EN DIFERENTES EJES. Se activa con (F3)



Se puede dividir la pantalla total del Monitor en el número de ejes que se desee, pero si esta división es mayor a 9, la resolución es deficiente, por esta razón se ha restringido a 9 ejes.

Se puede estructurar el número de ejes que desee, con el limitante expuesto y ubicar cada onda en el eje de su conveniencia.

## APENDICE D

## PROGRAMAS

El trabajo consta de 4 módulos: PSK.EXE, OPCIONES.EXE, ENTRADA.EXE, SIMULAR.EXE. Estos módulos se han compilado en lenguaje C, usando la opción PROJECT, que permite enlazar archivos (xx.C) mediante un archivo (xx.PRJ).

Los archivos de enlace son:

## OPCIONES.PRJ:

menu.c  
mayuda.c

## ENTRADA.PRJ:

pantalla.c  
entradas.c  
enayuda.c  
enmenu.c  
menusec.c  
enfiltro.c  
bucles.c

## SIMULAR.PRJ

elemento.c  
graficar.c  
control.c  
filtro.c  
pantalla.c  
sayuda.c  
lectura.c  
bucles.c

Se ha definido una librería de constantes y prototipos:

## PROYFIL.H:

```
#define Fnl (256*59)
#define ESC 0x1b
#define PI 3.14159
#define N 500
#define V2 1
void Initialize(void);
void Pause(void);
void MainWindow(int x1,int y1,int x2,int y2,int estilo,int mensaje,char *header);
void MainWindow_linea(int x1,int y1,int x2,int y2,int estilo);
void tema(int y1,int tan,char *header);
void ayuda(int x1,int y1,int x2,int y2,int caso);
void StatusLine(int caso,char *msg);
void DrawBorderg(void);
int gprintf(int *xloc, int *yloc,int separ, char *fmt, ... );
```

```

int mensaje(int yoc,int spar,int tipo,int help, char *msg);
void forma(float x,float y);
void formabuc1(float x,float y);
void formabuc2(float x,float y);
void formabuc3(float x,float y);
void formabuc4(float x,float y);
void formabuc5(float x,float y);
void formabuc6(float x,float y);
void formabuc7(float x,float y);
void circulo(int x,int y,int r);
void barra(int x1,int y1,int x2,int y2);
void barra1(int x1,int y1,int x2,int y2,int espesor,int modo,int relleno);
void BGRPLL(int x,int y,int des);
float PD(float einn,float eoscn);
float doblador(float uno,int caso);
float fillrobanda(float err,int rr);
char eldefasar(void);float AK(float we);
void respuesta_acomodada(int x, int y);
float defasar(float rr);
float VCO(float oid);
char entrada (void);
void elfiltro(int resp,int dc);
float filtro(float b, int a);
void elfillrob(int resp,int dc);
void elfillrobanda(int resp,int dc);
char estructurar_banda(void);
float fillrob(float b, int a);
void respuesta(int x,int y,int caso);
void respuesta_banda(int x,int y,int caso);
void respuesta_Butterworth(int x,int y);
void mod2psk(void);
void mod4psk(void);
void graficar(int caso);
short tecla (void);
char entrada_filtro(int de);
char entrada_banda(int de);
void CALBUC2(void);
void menu(void);
void inil(void);
void ayuda_linea(int num,int caso,int pos);
void prueba(void);
void elegir_grafico(int ycanal,int ncanales);
int ondaread_BUC2 (float x, float y);
int grread (int x,int y, char str[30],int indicador, int l, int esc,int help,int numero);
void cuadro(float x1,float y1);
void iaagen(int dix1,int diy1,int dix2,int diy2,int caso,int paso);
void borrar (int caso);void OrPrint (void);
void ubicar(int i,int ncanales,int ycanal,char *msg);
void poner(int i,int j,int ycanal,char *msg);
void CRUC0 (void);void CRUC1 (void);
void CRUC2 (void);void CRUC3 (void);
void CRUC4 (void);void CRUC5 (void);
void CRUC6 (void);void CRUC7 (void);
int menu_secundario(int x,int y);
char estructurar_defasador(void);
char estructurar_yco(void);
char estructurar_filtro(void);
void bow(float x1,float y1,int l,int ind1);

```

El código de los archivos (xx.C), se presenta a continuación.

```

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
PSX.C
Archivo raíz del paquete
*/

```

```

#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#include <stdio.h>
int GraphDriver,GraphMode;
int ErrorCode,nbucle;
int p_entrada,p_fpb,p_yco,p_def, p_fphan,ch;
void Initialize(void);

/*MAIN:Comandos principales en el comportamiento de PSX.EXE
*Parámetros :Ninguno
*Retorno : Ninguno
***/
void main(void)
{
FILE ffp;
ch=0;
Initialize();
p_entrada=0; p_fpb=0;p_yco=0;p_def=0; p_fphan=0;
do{
cleardevice();
system("menu.exe");
if((fp=fopen("a","rb"))==NULL){
closegraph();
printf("ERROR! DISCO LLENO, no se puede abrir ARCHIVO de enlace \n");
exit(1);
}
rewind(fp);
fread(&nbucle,sizeof(int),1,fp);
fclose(fp);
if(nbucle==21){
closegraph();printf("METODOS DE SINCRONISMO PSX\n");
printf("Ver. 1.0\n");
printf("ROLIVAR QUISPE\n");
printf("MARZO 1994");
exit(1);
}
MS:if((fp=fopen("c","wb+"))==NULL){
closegraph();
printf("no se puede abrir \n");
exit(1);
}
fwrite(&p_entrada,sizeof(int),1,fp);
fwrite(&p_fpb,sizeof(int),1,fp);
fwrite(&p_entrada,sizeof(int),1,fp);
fwrite(&p_yco,sizeof(int),1,fp);
fwrite(&p_def,sizeof(int),1,fp);
fwrite(&p_fphan,sizeof(int),1,fp);
fclose(fp);
system("entrada.exe");
p_entrada=1; p_fpb=1;p_yco=1;p_def=1; p_fphan=1;
if((fp=fopen("d","rb"))==NULL){
closegraph();
printf("ERROR! DISCO LLENO, no se puede abrir ARCHIVO de enlace \n");
exit(1);
}
}

```

```

rewind(fp);
fread(&ch,sizeof(int),1,fp);
fclose(fp);
if(ch!=5)
system("similar.exe");
if((fp=fopen("d","rb"))==NULL){
closegraph();
printf("ERROR! DISCO LLENO, no se puede abrir ARCHIVO de enlace \n");
exit(1);
}
rewind(fp);
fread(&ch,sizeof(int),1,fp);
fclose(fp);
if(ch==323)goto NS;
}while(nbucle !=21);
}

```

```

/*INITIALIZE: Consigue el modo grafico
* Parametros :Ninguno
* Retorno : Ninguno
#####/
void Initialize(void)
{
int xasp, yasp;
GraphDriver = DETECT;
initgraph( &GraphDriver, &GraphMode, "" );
ErrorCode = graphresult();
if( ErrorCode != grOk ){
closegraph();
printf("ERROR EN EL SISTEMA GRAFICO: %s\n",grapherrormsg(ErrorCode));
exit( 1 );
}
}

```

```

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
MENU.C
Presenta y controla el menu principal
*/

```

```

#include "stdlib.h"
#include <io.h>
#include <dir.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <graphics.h>
#include <stdio.h>
int $imagen,$imagen1,$ppcuadro, $pscuadro, $ptcuadro,ps,pl,p2,p3,p4,p5;
int GraphDriver,GraphMode,MaxColors,ErrorCode,nbucle,npl1;
float MaxX,MaxY,Esxc,Escy;
extern char far $buf;
void Initialize(void);
int menu_superior(int x, int y);
short lecla (void);
int menu_secundario1(int x, int y);
int menu_secundario2(int x, int y);
int menu_secundario3(int x, int y);
int menu_secundario4(int x, int y);
int menu_secundario5(int x, int y);
void cuadro (float x,float y,int esp);
void cuadros (float x,float y,int esp);

```

```

void tcuadros (float x,float y,int espx);
void ayuda(int dix1,int diy1,int dix2,int diy2, int caso);

/*MAIN:Comandos principales en el comportamiento de OPCIONES.EXE
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void main(void)
{
FILE *fp;
int nbuc,np;
    nbucle=100;np1=0;ps=0;p1=0;p2=0;p3=0;p4=0;p5=0;
    Initialize();
    ini:menu_superior(30*Escx,10);
    fp=fopen("a","wb+");
    fwrite(&nbucle,sizeof(int),1,fp);
    fwrite(&np1,sizeof(int),1,fp);
    fclose(fp);
    if(nbucle==20){
        GrPrint();
        cleardevice();
        nbucle=100;
        goto ini;
    }
}

/*MENU_SUPERIOR:Presenta y controla el menu horizontal (superior)
# Parametros :x.- coordenada X base
#             y.- coordenada Y base
# Retorno : # de la eleccion
#####/
int menu_superior(int x, int y)
{
FILE *fp;
size_t sscuadro;
int ch2s,pch2s=0,espx,espx1,espy1,espy,j=0,separ;
int dix1,diy1,dix2,diy2,caso,numeros[5],ind1,n=5;
    numeros[1] = 1;
    numeros[2] = 2;
    numeros[3] = 3;
    numeros[4] = 4;
    espy1=textheight("M");
    espy=espy1+8;separ=MaxX/9;
    outtextxy(separ,-y+15+espy1/2 , " DOS ");
    outtextxy(3*separ,-y+13+espy1/2 , " RUCLES ");
    outtextxy(5*separ,-y+15+espy1/2 , " FILTROS");
    outtextxy(7*separ,-y+15+espy1/2 , " AYUDAS ");
    setcolor(12);
    setlinestyle(0,1,3);
    rectangle(5,-y+14,MaxX-5,-y+14+espy);
    rectangle(0,-y+10,MaxX,MaxY);
    rectangle(2,-y+12,MaxX-2,MaxY-2);
    espx1=textwidth("M");      espx=15*espx1;
    setlinestyle(0,0,NORM_WIDTH);
    setfillstyle(SOLID_FILL, getmaxcolor());
    settxtjustify( LEFT_TEXT, TOP_TEXT );
    settxtstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    setcolor(1);
    sscuadro = imagesize (0,0,espx,espy);
    pscuadro = _graphgetmax (sscuardo);
    ind1 = 1;
    do

```

```

{
  if(pch2s==0)
    pch2s=1;
  else
    ind1=0;
  ch2s=328;
  do
  {
    if (ch2s == 328 || ch2s == 331 || ch2s == 333 || ch2s == 336)
      switch (ind1)
      {
        case 1: cuadro(separ-esp1*3,-y+14,espx);break;
        case 2: cuadro(separ*3-esp1*3,-y+14,espx);break;
        case 3: cuadro(separ*5-esp1*3,-y+14,espx);break;
        case 4: cuadro(separ*7-esp1*3,-y+14,espx);break;
      }
    ind1=pch2s;
    if(j != 331 && j != 333)ch2s = tecla();
    if (ch2s == 328 || ch2s == 331 || ch2s == 333 || ch2s == 336 || (ch2s>48 && ch2s<53) ||ch2s==13)
    {
      switch (ind1)
      {
        case 1: putimage(separ*1-esp1*3,-y+14,pscuadro, COPY_PUT);break;
        case 2: putimage(separ*3-esp1*3,-y+14,pscuadro, COPY_PUT);break;
        case 3: putimage(separ*5-esp1*3,-y+14,pscuadro, COPY_PUT);break;
        case 4: putimage(separ*7-esp1*3,-y+14,pscuadro, COPY_PUT);break;
      }
      if(j==331 || j==333) {
        if(j==331){
          if(ind1==1)ind1=n-1;
          else ind1=ind1-1;
        }
        if(j==333){
          if(ind1==(n-1))ind1=1;
          else ind1=ind1+1;
        }
        j=0;
        ch2s=13;
      }
      if(ch2s == '1' || ch2s == '2' || ch2s == '3' || ch2s == '4' )
        ind1=ch2s-48;

      if(ch2s == 328 || ch2s == 331){
        ind1--;
        if(ind1<1)
          ind1=1;
      }
      if(ch2s == 333 || ch2s == 336){
        if (ind1 == (n-1))
          ind1 = 1;
        else ind1++;
      }
      pch2s=ind1;
    }
  } while (ch2s != '\r' && ch2s != '1' && ch2s != '2' && ch2s != '3' && ch2s != '4' );
  switch (ind1)
  {
    case 1: cuadro(separ-esp1*3,-y+14,espx);break;
    case 2: cuadro(separ*3-esp1*3,-y+14,espx);break;
    case 3: cuadro(separ*5-esp1*3,-y+14,espx);break;
    case 4: cuadro(separ*7-esp1*3,-y+14,espx);break;
  }
  switch(ind1){

```

```

        case 1:j = menu_secundario1(separ-5*espx,-y+16+espy);break;
        case 2:j = menu_secundario2(3*separ-3*espx,-y+16+espy);break;
        case 3:j = menu_secundario3(5*separ-3*espx,-y+16+espy);break;
        case 4:j = menu_secundario4(7*separ-5*espx,-y+16+espy);break;
    }
    }while(nbucle==100);
    _graphfreeaem (pscuardo, sscuardo);
    return ind1;
}

```

/\*MENU\_SECUNDARIO1:Presenta y controla el menu DOS

\* Parametros :x.- coordenada X base

\* y.- coordenada Y base

\* Retorno : # de la eleccion

#####/

int menu\_secundario1(int x, int y)

```

{
    size_t sscuardo;
    size_t simagen;
    int ch2,espx,espy1,espx,espy,j,n=3,nueros[3],ind1;
        nueros[1] = 20;
        nueros[2] = 21;
        if(pi==0)
            ind1 = 1;
        else
            ind1=pi;
        espx=textwidth("M");espx=19*espx1;
        espy1=textheight("H")+8;espy=espy1;
        sscuardo = imagesize (0,0,espx,espy1);
        ppcuardo = _graphgetaem (sscuardo);
        simagen = imagesize (x-1,y,x+espx+17*Escx,y+11+espy*(n));
        imagen = _graphgetaem (simagen);
        getimage (x-1,y,x+espx+17*Escx,y+11+espy*(n),imagen);
        setcolor(getmaxcolor());
        outtextxy(x+espx1,y+espy/2+espy1/2-3, " IMPRIMIR F7");
        outtextxy(x+espx1,y+3*espy/2+espy1/2-3, " SALIR F4");
        setcolor(0);
        setlinestyle(SOLID_LINE,1,4);
        settextstyle(0,0,0);
        bar (x+espx,y+espy,x+espx+15*Escx,y+2+espy*(n));
        bar (x+15*Escx,y+espy*(n),x+espx+15*Escx,y+10+espy*(n));
        line(x+espx+3,y+espy,x+espx+3,y+2+espy*(n));
        line(x+15*Escx,y+2+espy*(n),x+espx+1,y+2+espy*(n));
        setcolor(12);
        setlinestyle(0,1,3);
        rectangle(x,y,x+espx,y+espy*(n));
        setlinestyle(0,0,NORM_WIDTH);
        setfillstyle (SOLID_FILL, getmaxcolor());
        setcolor (getmaxcolor());
        settextjustify( LEFT_TEXT, TOP_TEXT );
        settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
        setcolor(1);
        ch2=328;
        do
        {
            if (ch2 == 328 || ch2 == 336)
                switch (ind1)
                {
                    case 1: cuadros(x,y+espy/2,espx);break;
                    case 2: cuadros(x,y+3*espy/2,espx);break;
                }
            ch2 = getch();
        }
    }
}

```



```

if(ch2==321){
    ind1=1;ch2=13;
}
if(ch2==318){
    ind1=2;ch2=13;
}
if (ch2 == 328 !! ch2 == 336 !! (ch2>48 && ch2<51) !!ch2==13)
{
    switch (ind1)
    {
        case 1: putimage(x,y+espy/2,ppcuadro, COPY_PUT);break;
        case 2: putimage(x,y+0+3*espy/2,ppcuadro, COPY_PUT);break;
    }
if(ch2 == '1' !! ch2 == '2' )
    ind1=ch2-49;
if(ch2 == 328 ){
    ind1--;
    if(ind1<1)
        ind1=1;
}
if( ch2 == 336){
    if (ind1 == 2)
        ind1 = 1;
    else ind1++;
}
}
} while (ch2 != '\r' && ch2 !=27 && ch2 != 331 && ch2 !=333 && ch2 != '1' && ch2 != '2' );
switch (ind1)
{
    case 1: cuadros(x,y+espy/2,espx);break;
    case 2: cuadros(x,y+0+3*espy/2,espx);break;
}
_graphfreeem (ppcuadro, sscuadro);
putimage (x-1,y,imagen,COPY_PUT);
_graphfreeem (imagen, siimagen);
pl=ind1;
if(ch2 !=27 && ch2 !=331 && ch2 !=333)
    nbucle=numeros[ind1];
if(ch2 ==331 !! ch2 ==333)
    return ch2;
return ind1;
}

```

/\*MENU\_SECUNDARIO2:Presenta y controla el menu BUCLES

\* Parametros : x.- coordenada X base

\* y.- coordenada Y base

\* Retorno : # de la eleccion

#####

int menu\_secundario2(int x; int y)

{

size\_t sscuadro;

size\_t siimagen;

int ch2,espx,espx1,espy1,espy,j,n=9,numeros[9],ind1,k;

numeros[1] = 1;

numeros[2] = 2;

numeros[3] = 3;

numeros[4] = 4;

numeros[5] = 5;

numeros[6] = 6;

numeros[7] = 7;

numeros[8] = 0;

if(p2==0)

```

else
    ind1=p2;
espx1=tealwidth("M");espy1=espx1*42;
espy1=tealheight("H")+8;espy=espy1;
sscuardo = imagesize(0,0,espx,espy1);
ppcuadro = _graphgetelem(sscuadro);
simagen = imagesize(x-1,y,x+espx+17*Escx,y+11*espy*(n));
imagen = _graphgetelem(simagen);
getimage(x-1,y,x+espx+16*Escx,y+11*espy*(n),imagen);
setcolor(getmaxcolor());
outtextxy(x+espx1,y+espy/2+espy/2-3, " Con MULTIPLICACION           Alt M");
outtextxy(x+espx1,y+espy/2-3+espy/2*3, " Costas General           Alt G");
outtextxy(x+espx1,y+espy/2-3+espy/2*5, " REMODULACION por V.C.D.     Alt O");
outtextxy(x+espx1,y+espy/2-3+espy/2*7, " REMODULACION por onda RECIBIDA Alt R");
outtextxy(x+espx1,y+espy/2-3+espy/2*9, " Costas con detección en un brazo Alt D");
outtextxy(x+espx1,y+espy/2-3+espy/2*11, " Costas NUMERICO           Alt N");
outtextxy(x+espx1,y+espy/2-3+espy/2*13, " Costas para BPSK           Alt B");
outtextxy(x+espx1,y+espy/2-3+espy/2*15, " Phaselocked loop (P.L.L.)   Alt P");
setcolor(0);
setlinestyle(SOLID_LINE,1,4);
settextstyle(0,0,0);
bar(x+espx,y+espy,x+espx+15*Escx,y+2+espy*(n));
bar(x+15*Escx,y+espy*(n),x+espx+15*Escx,y+10+espy*(n));
line(x+espx+3*Escx,y+2+espy*(n),x+espx+3*Escx,y+espy);
line(x+15*Escx,y+2+espy*(n),x+espx+1,y+2+espy*(n));
setcolor(12);
setlinestyle(0,1,3);
rectangle(x,y,x+espx,y+espy*(n));
setlinestyle(0,0,NORM_WIDTH);
setfillstyle(SOLID_FILL, getmaxcolor());
setcolor(getmaxcolor());
settextjustify(LEFT_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
setcolor(1);
ch2=328;
do
{
    if(ch2 == 328 || ch2 == 336)
        switch(ind1)
        {
            case 1: cuadros(x,y+espy/2,espx);break;
            case 2: cuadros(x,y+espy/2*3,espx);break;
            case 3: cuadros(x,y+espy/2*5,espx);break;
            case 4: cuadros(x,y+espy/2*7,espx);break;
            case 5: cuadros(x,y+espy/2*9,espx);break;
            case 6: cuadros(x,y+espy/2*11,espx);break;
            case 7: cuadros(x,y+espy/2*13,espx);break;
            case 8: cuadros(x,y+espy/2*15,espx);break;
        }
    esc:ch2 = tecla();
    if(ch2==306){
        ind1=1;ch2=13;
    }
    if(ch2==290){
        ind1=2;ch2=13;
    }
    if(ch2==280){
        ind1=3;ch2=13;
    }
    if(ch2==275){
        ind1=4;ch2=13;
    }
}

```

```

    }
    if(ch2==288){
        ind1=5;ch2=13;
    }
    if(ch2==305){
        ind1=6;ch2=13;
    }
    if(ch2==272){
        ind1=7;ch2=13;
    }
    if(ch2==281){
        ind1=8;ch2=13;
    }
    if (ch2 == 328 || ch2 == 336 || (ch2>48 && ch2<57) ||ch2==13)
    {
        switch (ind1)
        {
            case 1: putimage(x,y+espy/2,ppcuadro, COPY_PUT);break;
            case 2: putimage(x,y+espy/2*3,ppcuadro, COPY_PUT);break;
            case 3: putimage(x,y+espy/2*5,ppcuadro, COPY_PUT);break;
            case 4: putimage(x,y+espy/2*7,ppcuadro, COPY_PUT);break;
            case 5: putimage(x,y+espy/2*9,ppcuadro, COPY_PUT);break;
            case 6: putimage(x,y+espy/2*11,ppcuadro, COPY_PUT);break;
            case 7: putimage(x,y+espy/2*13,ppcuadro, COPY_PUT);break;
            case 8: putimage(x,y+espy/2*15,ppcuadro, COPY_PUT);break;
        }
        if(ch2 == '1' || ch2 == '2' || ch2 == '3' || ch2 == '4' || ch2 == '5' || ch2 == '6' || ch2 == '7' || ch2 == '8')
            ind1=ch2-48;

        if(ch2 == 328 ){
            ind1--;
            if(ind1<1)
                ind1=1;
        }

        if( ch2 == 336){
            if (ind1 == 8)
                ind1 = 1;
            else ind1++;
        }
    }
    while (ch2 != '\r' && ch2 !=27 && ch2 != 331 && ch2 !=333 && ch2 != '1' && ch2 != '2' && ch2 != '3' &&
        ch2 != '4' && ch2 != '5' && ch2 != '6' && ch2 != '7' && ch2 != '8' );
    switch (ind1)
    {
        case 1: cuadros(x,y+espy/2,espx);break;
        case 2: cuadros(x,y+espy/2*3,espx);break;
        case 3: cuadros(x,y+espy/2*5,espx);break;
        case 4: cuadros(x,y+espy/2*7,espx);break;
        case 5: cuadros(x,y+espy/2*9,espx);break;
        case 6: cuadros(x,y+espy/2*11,espx);break;
        case 7: cuadros(x,y+espy/2*13,espx);break;
        case 8: cuadros(x,y+espy/2*15,espx);break;
    }
    if(ch2 !=27 && ch2 !=331 && ch2 !=333)
    {
        nbucle=numero[ind1];
        if(nbucle==0){
            do{
                k=menu_secundario5(370*Esca,9*espy);
                if(k==27){
                    nbucle=100;
                    goto esc;
                }
            }
        }
    }

```

```

    }
    }while(k!=2 && k != 4 && k!=27);
}

}

p2=ind1;
_graphfree(m (ppcuadro, sscuadro);
putimage (x-1,y,imagen,COPY_PUT);
_graphfree(m (imagen, simagen);
if(ch2 ==331 || ch2 ==333)
    return ch2;
return ind1;
}

/*MENU_SECUNDARIO3:Presenta y controla el menu FILTROS
# Parametros :x.- coordenada X base
#             y.- coordenada Y base
# Retorno : # de la eleccion
#####/
int menu_secundario3(int x, int y)
{
size_t sscuadro;
size_t simagen;
int ch2,espx,espy,j,espx1,espy1,n=3,numeros[3],ind1;
    numeros[1] = 10;
    numeros[2] = 11;
    if(p3==0)
        ind1 = 1;
    else
        ind1=p3;
    espx1=linewidth("M");espx=34*espx1;
    espy1=textheight("H")+8;espy=espy1;
    sscuadro = imagesize (0,0,espx,espy1);
    ppcuadro = _graphgetmem (sscuadro);
    simagen = imagesize (x-1,y,x+espx+17*Escx,y+11*espy*(n));
    imagen = _graphgetmem (simagen);
    getimage (x-1,y,x+espx+17*Escx,y+11*espy*(n),imagen);
    setcolor(getmaxcolor());
    outtextxy(x+espx1,y+espy/2+espy1/2-3, " Pasabajos ELIPTICO Alt L");
    outtextxy(x+espx1,y+espy1/2-3+espy/2*3, " Pasabajos BUTTERWORTH Alt B");
    setcolor(0);
    setlinestyle(SOLID_LINE,1,4);
    settextstyle(0,0,0);
    bar (x+espx,y+espy,x+espx+15*Escx,y+2*espy*(n));
    bar (x+15*Escx,y+espy*(n),x+espx+15*Escx,y+10*espy*(n));
    line(x+espx+3*Escx,y+2*espy*(n),x+espx+3*Escx,y+espy);
    line(x+15*Escx,y+2*espy*(n),x+espx+1,y+2*espy*(n));
    setcolor(12);
    setlinestyle(0,1,3);
    rectangle(x,y,x+espx,y+espy*(n));
    setlinestyle(0,0,NORM_WIDTH);
    setfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    settextjustify( LEFT_TEXT, TOP_TEXT );
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    setcolor(1);
    ch2=328;
    do
    {
        if (ch2 == 328 || ch2 == 336)
            switch (ind1)
            {
                case 1:cuadros(x,y+espy/2,espx);

```

```

        break;
        case 2: cuadros(x,y+espy/2*3,espx);break;
    }
    ch2 = tecla();
    if(ch2==294){
        ind1=1;
        ch2=13;
    }
    if(ch2==304){
        ind1=2;
        ch2=13;
    }
    if (ch2 == 328 || ch2 == 336 || (ch2>48 && ch2<51) ||ch2==13)
    {
        switch (ind1)
        {
            case 1: putimage(x,y+espy/2,ppcuadro, COPY_PUT);break;
            case 2: putimage(x,y+espy/2*3,ppcuadro, COPY_PUT);break;
        }
        if(ch2 == '1' || ch2 == '2' )
            ind1=ch2-48;
        if(ch2 == 328 ){
            ind1--;
            if(ind1<1)
                ind1=1;
        }
        if( ch2 == 336){
            if (ind1 == 2)
                ind1 = 1;
            else ind1++;
        }
    }
    } while (ch2 != '\r' && ch2 !=27 && ch2 != 331 && ch2 !=333 && ch2 != '1' && ch2 != '2' );
switch (ind1)
{
    case 1: cuadros(x,y+espy/2,espx);break;
    case 2: cuadros(x,y+espy/2*3,espx);break;
}
p3=ind1;
_graphfreeea (ppcuadro, sscuadro);
putimage (x-1,y,imagen,COPY_PUT);
_graphfreeea (imagen, simagen);
if(ch2 !=27 && ch2 !=331 && ch2 !=333)
    nbucle=numeros[ind1];
if(ch2 ==331 || ch2 ==333)
    return ch2;
return ind1;
}

/vMENU_SECUNDARIO4:Presenta y controla el menu AYUDAS
# Parametros :x.- coordenada X base
#             y.- coordenada Y base
# Retorno : # de la eleccion
#####/
int menu_secundario4(int x, int y)
{
    size_t sscuadro;
    size_t simagen;
    int ch2,espx,espy,j,espx1,ex,espy1,a,n=6,numeros[6],ind1;
    numeros[1] = 12;
    numeros[2] = 13;
    numeros[3] = 14;

```

```

numeros[4] = 15;
numeros[5] = 16;
if(p4==0)
    ind1 = 1;
else
    ind1=p4;
espx1=ceilwidth("M");espx=20*espx1;
espy1=ceilheight("H")+8;espy=espy1;
sscuardo = imagesize (0,0,espx,espy1);
ppcuadro = _graphgetaem (sscuardo);
siimagen = imagesize (x-1,y,x+espx1/7*Escx,y+11*espy*(n));
iimagen = _graphgetaem (siimagen);
gelimage (x-1,y,x+espx1/6*Escx,y+11*espy*(n),iimagen);
setcolor(gelaaaxcolor());
outtextxy(x+espx1,y+espy/2+espy1/2-3, ' FILTROS ');
outtextxy(x+espx1,y+espy1/2-3+espy/2*3, ' FILTRO PASABAJOS');
outtextxy(x+espx1,y+espy1/2-3+espy/2*5, ' FILTRO PASABANDA');
outtextxy(x+espx1,y+espy1/2-3+espy/2*7, ' V.C.O. ');
outtextxy(x+espx1,y+espy1/2-3+espy/2*9, ' DEFASADOR ');
setcolor(0);
setlinestyle(SOLID_LINE,1,4);
settextstyle(0,0,0);
bar (x+espx,y+espy,x+espx+15*Escx,y+2*espy*(n));
bar (x+15*Escx,y+espy*(n),x+espx+15*Escx,y+10*espy*(n));
line(x+espx+3*Escx,y+2*espy*(n),x+espx+3*Escx,y+espy);
line(x+15*Escx,y+2*espy*(n),x+espx+1,y+2*espy*(n));
setcolor(12);
setlinestyle(0,1,3);
rectangle(x,y,x+espx,y+espy*(n));
setlinestyle(0,0,HORM_WIDTH);
setfillstyle (SOLID_FILL, gelaaaxcolor());
setcolor (gelaaaxcolor());
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
setcolor(1);
ch2=328;
ini:do
(
    if (ch2 == 328 || ch2 == 336)
        switch (ind1)
        (
            case 1: cuadros(x,y+espy/2,espx);break;
            case 2: cuadros(x,y+espy/2*3,espx);break;
            case 3: cuadros(x,y+espy/2*5,espx);break;
            case 4: cuadros(x,y+espy/2*7,espx);break;
            case 5: cuadros(x,y+espy/2*9,espx);break;
        )
    ch2 = tecla();
    if(ch2==274){
        ind1=1;ch2=13;
    }
    if(ch2==302){
        ind1=2;ch2=13;
    }
    if(ch2==289){
        ind1=3;ch2=13;
    }
    if(ch2==287){
        ind1=4;ch2=13;
    }
    if(ch2==286){
        ind1=5;ch2=13;
    }

```

```

    }
    if (ch2 == 328 || ch2 == 336 || (ch2>48 && ch2<54) || ch2==13)
    {
        switch (ind1)
        {
            case 1: pulimage(x,y+espy/2,ppcuadro, COPY_PUT);break;
            case 2: pulimage(x,y+espy/2*3,ppcuadro, COPY_PUT);break;
            case 3: pulimage(x,y+espy/2*5,ppcuadro, COPY_PUT);break;
            case 4: pulimage(x,y+espy/2*7,ppcuadro, COPY_PUT);break;
            case 5: pulimage(x,y+espy/2*9,ppcuadro, COPY_PUT);break;
        }
        if(ch2 == '1' || ch2 == '2' || ch2 == '3' || ch2 == '4' || ch2 == '5')
            ind1=ch2-48;

        if(ch2 == 328 ){
            ind1--;
            if(ind1<1)
                ind1=1;
        }
        if( ch2 == 336){
            if (ind1 == 5)
                ind1 = 1;
            else ind1++;
        }
    }
    } while (ch2 != '\r' && ch2 !=27 && ch2 != 331 && ch2 !=333 && ch2 != '1' && ch2 != '2' && ch2 != '3' &&
    ch2 != '4' && ch2 != '5' );
    switch (ind1)
    {
        case 1: cuadros(x,y+espy/2,espx);break;
        case 2: cuadros(x,y+espy/2*3,espx);break;
        case 3: cuadros(x,y+espy/2*5,espx);break;
        case 4: cuadros(x,y+espy/2*7,espx);break;
        case 5: cuadros(x,y+espy/2*9,espx);break;
    }
    p4=ind1;
    if(ch2 !=27 && ch2 !=331 && ch2 !=333)
        nbucle=numeros[ind1];
    if(nbucle>11 && nbucle <17)
    {
        ex=(MaxX-69*espx)/2;n=MaxX/2;
        switch(nbucle){
            case 12: ayuda(ex,45*Escy,ex+69*espx,45*Escy+29*textheight("H"),1040);break;
            case 13: ayuda(ex,45*Escy,ex+69*espx,45*Escy+34*textheight("H"),1001);break;
            case 14: ayuda(n-espx*39,45*Escy,n+espx*39,45*Escy+26*textheight("H"),1010);break;
            case 15: ayuda(ex,45*Escy,ex+69*espx,45*Escy+20*textheight("H"),1020);break;
            case 16: ayuda(65*Escx,45*Escy,65*Escx+69*espx,45*Escy+29*textheight("H"),1030);break;
        }
        nbucle=100;goto ini;
    }
    _graphfreearea (ppcuadro, sscuadro);
    pulimage (x-1,y,imagen,COPY_PUT);
    _graphfreearea (imagen, siimagen);
    if(ch2 ==331 || ch2 ==333)
        return ch2;
    return ind1;
}

/*MENU_SECUNDARIO5:Presenta y controla el menu PLL
# Parametros : x.- coordenada X base
# y.- coordenada Y base
# Retorno : # de la eleccion
*****/

```

```

int eenu_secundario5(int x, int y)
{
size_t sscuadro;
size_t siaagen;
int ch2,espx,espy,j,espx1,espy1,n=3,numeros[3],ind1;
    numeros[1] = 2;
    numeros[2] = 4;
    if(p5==0)
        ind1 = 1;
    else
        ind1=p5;
    espx1=lexwidth("M");espx=32*espx1;
    espy1=lexheight("H")+8;espy=espy1;
    sscuadro = imagesize (0,0,espx,espy1);
    plcuardro = _graphgetaem (sscuardro);
    siaagen1 = imagesize (x-1,y-1,x+espx+17*Escx,y+11+espy*(n));
    iaagen1 = _graphgetaem (siaagen1);
    getimage (x-1,y-1,x+espx+16*Escx,y+11+espy*(n),iaagen1);
    setviewport(0,0,MaxX,MaxY,1);
    setviewport(x-1,y,x+espx+16*Escx,y+11+espy*(n),1);
    clearviewport();
    setviewport(0,0,MaxX,MaxY,1);
    setcolor(getmaxcolor());
    outtextxy(x+espx1,y+espy/2+espy1/2-3, " entrada Onda BPSK All T");
    outtextxy(x+espx1,y+espy1/2-3+espy/2*3, " entrada Onda QPSK All U");
    setcolor(0);
    setlinestyle(SOLID_LINE,1,4);
    settextstyle(0,0,0);
    bar (x+espx,y+espy,x+espx+15*Escx,y+2+espy*(n));
    bar (x+15*Escx,y+espy*(n),x+espx+15*Escx,y+10+espy*(n));
    line(x+espx+3*Escx,y+2+espy*(n),x+espx+3*Escx,y+espy);
    line(x+15*Escx,y+2+espy*(n),x+espx+1,y+2+espy*(n));
    setcolor(12);
    setlinestyle(0,1,3);
    rectangle(x,y,x+espx,y+espy*(n));
    setlinestyle(0,0,NORM_WIDTH);
    setfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    settextjustify( LEFT_TEXT, TOP_TEXT );
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    setcolor(1);
    ch2=328;
do
{
    if {ch2 == 328 !! ch2 == 336}
        switch (ind1)
        {
            case 1: lcuardros(x,y+espy/2,espx);break;
            case 2: lcuardros(x,y+espy/2*3,espx);break;
        }
    ch2 = lecla();
    if{ch2==276}{
        ind1=1;ch2=13;
    }
    if{ch2==278}{
        ind1=2;ch2=13;
    }
    if {ch2 == 328 !! ch2 == 336 !! (ch2>48 && ch2<51) !!ch2==13}
    {
        switch (ind1)
        {
            case 1: pulimage(x,y+espy/2,plcuardro, COPY_PUT);break;

```



```

        case 2: putimage(x,y+espy/2+3,plcuadro, COPY_PUT);break;
    }
    if(ch2 == '1' || ch2 == '2' )
        ind1=ch2-48;
    if(ch2 == 328 ){
        ind1--;
        if(ind1<1)
            ind1=1;
    }
    if( ch2 == 336){
        if (ind1 == 2)
            ind1 = 1;
        else ind1++;
    }
}
} while (ch2 != '\r' && ch2 !=27 && ch2 != 331 && ch2 !=333 && ch2 != '1' && ch2 != '2' );
    switch (ind1)
    {
        case 1: lcuadros(x,y+espy/2,espx);break;
        case 2: lcuadros(x,y+espy/2+3,espx);break;
    }
    p5=ind1;
    _graphfreeea (plcuadro, sscuadro);
    putimage (x-1,y-1,imagen1,COPY_PUT);
    _graphfreeea (imagen1, simagen1);
    if(ch2 !=27 && ch2 !=331 && ch2 !=333)
        np1=numeros[ind1];
    if(ch2 ==331 || ch2 ==333 || ch2 == 27)
        return ch2;
    return numeros[ind1];
}

/*INITIALIZE: Consigue el modo grafico
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void Initialize(void)
{
    GraphDriver = DETECT;
    initgraph( &GraphDriver, &GraphMode, "" );
    MaxColors = getmaxcolor() + 1;
    MaxX = getmaxx();
    MaxY = getmaxy();
    Escx =(float) getmaxx()/720;
    Escy =(float) getmaxy()/348;
}

/*CUADRO: forma el relleno de la opción actual para menu_superior
# Parametros : x .- coordenada X base
#             y .- coordenada Y base
#             espx .- espesor horizontal del relleno
# Retorno : Ninguno
#####/
void cuadro(float x1,float y1,int espx)
{
    int espy1;
    size_t sscuadro;
    espy1=lextheigh("H")+8;
    getimage (x1,y1,x1+espx,y1+espy1, pscuadro);
    selfillstyle (SOLID_FILL, 0);
    bar (x1,y1, x1+espx,y1+espy1);
    setcolor (getmaxcolor());
}

```

```

rectangle (x1+2, y1+2, x1+espx-2, y1+espyl-2);
selfillstyle (SOLID_FILL, getmaxcolor());
bar (x1+4, y1+4, x1+espx-4, y1+espyl-4);
putimage (x1,y1, pscuadro, XOR_PUT);
setcolor (0);
}

/*CUADROS: forma el relleno de la opción actual para menu_secundario1,menu_secundario2,
* menu_secundario3 y menu_secundario4.
* Parametros : x .- coordenada X base
*              y .- coordenada Y base
*              espx .- espesor horizontal del relleno
* Retorno : Ninguno
*/
void cuadros(float x1,float y1,int espx)
{
int espyl;
size_t sscuadro;
    espyl=textheight("H")+8;
    getimage (x1,y1, x1+espx,y1+espyl, ppcuadro);
    selfillstyle (SOLID_FILL, 0);
    bar (x1,y1, x1+espx,y1+espyl);
    setcolor (getmaxcolor());
    rectangle (x1+2, y1+2, x1+espx-2, y1+espyl-2);
    selfillstyle (SOLID_FILL, getmaxcolor());
    bar (x1+4, y1+4, x1+espx-4, y1+espyl-4);
    putimage (x1,y1, ppcuadro, XOR_PUT);
    setcolor (0);
}

/*TCUADROS: forma el relleno de la opción actual para menu_secundario5
* Parametros : x .- coordenada X base
*              y .- coordenada Y base
*              espx .- espesor horizontal del relleno
* Retorno : Ninguno
*/
void tcuadros(float x1,float y1,int espx)
{
int espyl;
size_t sscuadro;
    espyl=textheight("H")+8;
    getimage (x1,y1, x1+espx,y1+espyl, ptcuadro);
    selfillstyle (SOLID_FILL, 0);
    bar (x1,y1, x1+espx,y1+espyl);
    setcolor (getmaxcolor());
    rectangle (x1+2, y1+2, x1+espx-2, y1+espyl-2);
    selfillstyle (SOLID_FILL, getmaxcolor());
    bar (x1+4, y1+4, x1+espx-4, y1+espyl-4);
    putimage (x1,y1, ptcuadro, XOR_PUT);
    setcolor (0);
}

/*TECLA: Entrada de caracteres desde el teclado
* Parametros :Ninguno
* Retorno : Código del caracter
*/
short lecla (void)
{
union REGS r;
    r.i.ax=0;
    int86(0x16,&r,&r);
    return((r.h.al==0)?(256+(short) r.h.ah):(short) r.h.al);
}

```



```

gprintf(&x,&y,2,"aproximaciones matematicas.");
gprintf(&x,&y,2,"      FILTRO ELIPTICO.- Permiten obtener un corte extremadamente");
gprintf(&x,&y,2,"abrupto y disponer de ceros de transmision, es decir se puede");
gprintf(&x,&y,2,"eliminar frecuencias no deseadas ya que a la frecuencia donde se");
gprintf(&x,&y,2,"da el cero de transmision la atenuacion tiene un valor INFINITO.");
gprintf(&x,&y,2,"      FILTRO DE BUTTERWORTH.- Tienen un corte poco abrupto. Pero");
gprintf(&x,&y,2,"presenta deformaciones insignificantes en regimen Transitorio.");
gprintf(&x,&y,2,"      Otras aproximaciones como Tchebyscheff y Legendre tienen ");
gprintf(&x,&y,2,"las dos caracteristicas mencionadas intermedias entre las de los");
gprintf(&x,&y,2,"dos filtros escogidos.Desde el punto de vista de respuesta tran_");
gprintf(&x,&y,2,"sitoria los fillos de Bessel son los de mejores resultados, pero");
gprintf(&x,&y,2,"su atenuacion crece muy lentamente en funcion de la frecuencia,");
gprintf(&x,&y,2,"aun cuando el ORDEN sea elevado.");
MainWindow(dix1,diy1,dix2,diy2,1,1,"      A Y U D A      ");
do{
    dec=lecla();
    if(dec==321){
        GrPrint();
        setviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27 && dec!=337 );
if(dec==27)break;
case 1002:
    n2:    x=60*Escx;y=2*Escy;
    borrar_pagina( dix1,diy1,dix2,diy2);
    gprintf(&x,&y,2,"RESPUESTA DE FRECUENCIA DEL FILTRO PASABAJOS ELIPTICO");
    x=175*Escx;y=17*Escy;
    gprintf(&x,&y,2,"fp : frecuencia de corte del filtro [Hz]");
    gprintf(&x,&y,2,"fr : frecuencia de rechazo del filtro [Hz]");
    gprintf(&x,&y,2,"Afp: atenuacion en fp [dB]");
    gprintf(&x,&y,2,"Afr: atenuacion en fr [dB]");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Plantilla de un filtro de TERCER orden ");
    respuesta_acomodada(35*Escx,-90*Escy);
    MainWindow(dix1,diy1,dix2,diy2,1,2,"      A Y U D A      ");
    do{
        dec=lecla();
        if(dec==321){
            GrPrint();
            setviewport(dix1,diy1,dix2,diy2,1);
        }
    }while(dec!=27 && dec!=337 && dec!=329);
    if(dec==27)break;
    if(dec==329)goto n1;
case 1003:
    borrar_pagina( dix1,diy1,dix2,diy2);
    x=30*Escx;y=2*Escy;
    gprintf(&x,&y,2,"RESPUESTA DE FRECUENCIA DEL FILTRO PASABAJOS DE BUTTERWORTH");
    x=175*Escx;y=17*Escy;
    gprintf(&x,&y,2,"fp : frecuencia de corte del filtro [Hz]");
    gprintf(&x,&y,2,"fr : frecuencia de rechazo del filtro [Hz]");
    gprintf(&x,&y,2,"Afp: atenuacion en fp [dB]");
    gprintf(&x,&y,2,"Afr: atenuacion en fr [dB]");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Plantilla de un filtro de SEGUNDO orden");
    respuesta_Butlerworth(35*Escx,-80*Escy);
    MainWindow(dix1,diy1,dix2,diy2,1,3,"      A Y U D A      ");
    do{
        dec=lecla();
        if(dec==321){
            GrPrint();

```

```

        selviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27 && dec!=329);
if(dec==27)break;
if(dec==329)goto a7;
case 1010:
a10:   borrar_pagina( dix1,diy1,dix2,diy2);
x=10*Escy;y=-21*Escy;
gprintf(&x,&y,2,"          ESTRUCTURACION DEL FILTRO PASABANDA ");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Para la simulacion del filtro pasabanda se estructura un filtro ");
gprintf(&x,&y,2," simetrico (La respuesta de frecuencia del filtro es simetrica con respecto");
gprintf(&x,&y,2," a la frecuencia central fo). Se lo obtiene mediante una transformacion ");
gprintf(&x,&y,2," de frecuencia de un filtro prototipo pasabajos. fo corresponde a f=0 en ");
gprintf(&x,&y,2," el prototipo, de igual manera fp del filtro pasabajos [VER FILTRO PASABA_");
gprintf(&x,&y,2," 309] se transforman en dos frecuencias geometricamente simetricas, lo mismo ");
gprintf(&x,&y,2," sucede con fr.");
gprintf(&x,&y,2," Un filtro pasabanda simetrico esta totalmente definido si se conoce");
gprintf(&x,&y,2," 5 parametros: fp,fr,Afp,Afr,fo [ver GRAFICO en la siguiente pagina] ");
MainWindow(dix1,diy1,dix2,diy2,1,1,"          A Y U D A          ");
do{
    dec=tecla();
    if(dec==321){
        GrPrint();
        selviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27 && dec!=337 );
if(dec==27)break;
case 1011:
borrar_pagina( dix1,diy1,dix2,diy2);
x=dix2-54*textwidth("N");y=17*Escy;
gprintf(&x,&y,2,"ESTRUCTURACION DEL FILTRO PASABANDA ELIPTICO ");
x=dix2-54*textwidth("N");y=31*Escy;
gprintf(&x,&y,2,"fo :frecuencia central      0<fo<[INFINITOCHZ]");
gprintf(&x,&y,2,"fp :frecuencia de corte      0<pf<fo [Hz]");
gprintf(&x,&y,2,"fr :frecuencia de supresion 0<fr<fp [Hz]");
gprintf(&x,&y,2,"Afp:atenuacion en fp      0<Afp<20 [dB]");
gprintf(&x,&y,2,"Afr:atenuacion en fr      0<Afr<100 [dB]");
gprintf(&x,&y,2," s=j2nf con k=2n");
respuesta_banda(32*Escx,-172*Escy,3);
MainWindow(dix1,diy1,dix2,diy2,1,3,"          A Y U D A          ");
do{
    dec=tecla();
    if(dec==321){
        GrPrint();
        selviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27 && dec!=329);
if(dec==27)break;
if(dec==329)goto a10;
case 1020:
a20:   borrar_pagina( dix1,diy1,dix2,diy2);
x=10*Escy;y=-2*Escy;
gprintf(&x,&y,2," ESTRUCTURACION del OSCILADOR CONTROLADO por VOLTAJE.");
gprintf(&x,&y,2," Se utiliza un VCO de respuesta lineal de frecuencia con");
gprintf(&x,&y,2," respecto al voltaje de control (Vc). El voltaje de ERROR de fase");
gprintf(&x,&y,2," (Ve) esta normalizado, asi: -1 < Ve < 1. ");
gprintf(&x,&y,2," Se considera Vc = 1 + Ve , en consecuencia 0 < Vc < 2.");
gprintf(&x,&y,2," Para la estructuracion se consideran tres caracteristicas");
gprintf(&x,&y,2," del VCO : Frecuencia central, Estabilidad y Tolerancia. ");
gprintf(&x,&y,2," FRECUENCIA CENTRAL DE OSCILACION (fo):");
gprintf(&x,&y,2," Es la frecuencia a la que oscila el VCO cuando Ve es cero. Debe");

```

```

gprintf(&x,&y,2,"ser igual a la utilizada para modular la onda de entrada a los");
gprintf(&x,&y,2,"bucles, para que sus respuestas sean las deseadas.");
gprintf(&x,&y,2," Su valor esta dado en [Hertz].");
MainWindow(dix1,diy1,dix2,diy2,1,1," A Y U D A ");
do{
    dec=lecla();
    if(dec==321){
        GrPrint();
        setviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27 && dec!=337 );
if(dec==27)break;
case 1021:
    borrar_pagina( dix1,diy1,dix2,diy2);
    x=0xFEscx;y=-2fEscy;
    gprintf(&x,&y,2," ESTABILIDAD del VCO (E) :");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Es el cambio de frecuencia por la variacion del error de");
    gprintf(&x,&y,2," fase en un grado.");
    gprintf(&x,&y,2," Conocidas fo y E estan definidas las frecuencias INFERIOR ");
    gprintf(&x,&y,2," y SUPERIOR de oscilacion.");
    gprintf(&x,&y,2," EJEMPLO : frecuencia CENTRAL (fo) = 1800 [Hz]");
    gprintf(&x,&y,2," ESTABILIDAD = 20 [Hz/grado]");
    gprintf(&x,&y,2," Cuando Vc=2, la frec. de oscilacion es 3600[Hz]");
    gprintf(&x,&y,2," Cuando Vc=0 el VCO no oscila");
    gprintf(&x,&y,2," Es decir, F.INFERIOR = 0 [Hz] y F. SUPERIOR = 3600 [Hz]");
    MainWindow(dix1,diy1,dix2,diy2,1,2," A Y U D A ");
    do{
        dec=lecla();
        if(dec==321){
            GrPrint();
            setviewport(dix1,diy1,dix2,diy2,1);
        }
    }while(dec!=27 && dec!=337 && dec!=329);
    if(dec==27)break;
    if(dec==329)goto a20;
case 1022:
    borrar_pagina( dix1,diy1,dix2,diy2);
    x=10fEscx;y=-2fEscy;
    gprintf(&x,&y,2," TOLERANCIA del V.C.O. (T) :");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Su valor esta dado en porcentaje : 0 < TOLERANCIA <100 % ");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," El V.C.O. produce cambio de frecuencia en su salida uni_");
    gprintf(&x,&y,2," caente si el voltaje de ERROR de fase (Ve), supera el valor que");
    gprintf(&x,&y,2," se ingresa en porcentaje.");
    gprintf(&x,&y,2," El ERROR maximo (100 %) se da cuando la diferencia de ");
    gprintf(&x,&y,2," fase es de 90 grados. Ya que (Ve) es proporcional a la funcion ");
    gprintf(&x,&y,2," trigonometrica seno del ERROR de FASE.");
    MainWindow(dix1,diy1,dix2,diy2,1,3," A Y U D A ");
    do{
        dec=lecla();
        if(dec==321){
            GrPrint();
            setviewport(dix1,diy1,dix2,diy2,1);
        }
    }while(dec!=27 && dec!=329);
    if(dec==27)break;
    if(dec==329)goto a21;
case 1030:
    borrar_pagina( dix1,diy1,dix2,diy2);
    x=10fEscx;y=-2fEscy;

```

```

gprintf(&x,&y,2,"      ESTRUCTURACION DEL DEFASADOR DE 90 GRADOS ");
gprintf(&x,&y,2,"      Se analiza un filtro que produce una diferencia de fase de");
gprintf(&x,&y,2,"90 grados con un error inferior a E, a todas las componentes de ");
gprintf(&x,&y,2,"frecuencia de la onda de entrada, dentro de un rango de frecuen_");
gprintf(&x,&y,2,"cias ( f1 , f2 ) dadas.");
gprintf(&x,&y,2,"      f1 y f2 son las frecuencias inferior y superior del filtro");
gprintf(&x,&y,2,"pasabanda que se forma,por lo tanto su valor se acepta en [Hz].");
gprintf(&x,&y,2,"      E es el error de fase maximo que produce el defasador. Su");
gprintf(&x,&y,2,"valor esta dado en [grados].");
gprintf(&x,&y,2,"      Dadas las frecuencias criticas y el error de fase maximo,");
gprintf(&x,&y,2,"se hacen los calculos necesarios para determinar el orden de");
gprintf(&x,&y,2,"filtro, el que se recomienda al usuario.");
MainWindow(dix1,diy1,dix2,diy2,1,6," ");
do{
    dec=tecla();
    if(dec==321){
        GrPrint();setviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27);
break;
case 1040:horrar_pagina( dix1,diy1,dix2,diy2);
x=104Esc; y=-24Esc;
gprintf(&x,&y,2,"      CONSTRUCCION DE FILTROS PASABAJOS      ");
gprintf(&x,&y,2,"      Los dos tipos de filtros utilizados en la simulacion pue_");
gprintf(&x,&y,2,"den ser construidos en base a Amplificadores operacionales, re_");
gprintf(&x,&y,2,"sistencias y capacitores, es decir, construir un filtro activo");
gprintf(&x,&y,2,"Elíptico o de Butterworth.");
gprintf(&x,&y,2,"      En base a la teoria de aproximacion de los filtros mencio_");
gprintf(&x,&y,2,"nados se obtiene la funcion de transferencia normalizada con res");
gprintf(&x,&y,2,"pecto a impedancias y frecuencia, en el caso de pasabajos se ");
gprintf(&x,&y,2,"normaliza con respecto a la frecuencia de corte del filtro (wp)");
gprintf(&x,&y,2,"FVER FILTRO PASABAJOS.");
gprintf(&x,&y,2,"      El programa le permite conocer los valores de las resisten_");
gprintf(&x,&y,2,"cias, capacitores y amplificacion de cada etapa del filtro.");
MainWindow(dix1,diy1,dix2,diy2,1,6," ");
do{
    dec=tecla();
    if(dec==321){
        GrPrint();setviewport(dix1,diy1,dix2,diy2,1);
    }
}while(dec!=27);
break;
}
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
clearviewport();
setviewport(0,0,MaxX,MaxY,1);
putimage(dix1-1,diy1-1,buf,COPY_PUT);
_graphfreeen (buf,la);
setfillstyle (SOLID_FILL, getmaxcolor());
setcolor (getmaxcolor());
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
}
/*las siguientes funciones de este archivo, tienen su código en el archivo(X.C)
#GPRINTF: Semejante a PRINTF pero en modo grafico (PANTALLA.C)
#MAINWINDOW:despliega en pantalla un recuadro (PANTALLA.C)
#DRAWBORDE: Dibuja un cuadro sobre en los limites de la ventana grafica actual (PANTALLA.C)
#MENSAJE: Despliega un mensaje de acuerdo al caso (PANTALLA.C)
#RESPUESTA_ACOMODADA:Respuesta de frecuencia prototipo del filtro pasabajos ELIPTICO (ENFILTRO.C)
#RESPUESTA_BUTTERWORTH:Respuesta de frecuencia prototipo del filtro pasabajos de BUTTERWORTH (ENFILTRO.C)
#RESPUESTA_BANDA:Respuesta de frecuencia prototipo del filtro pasabanda (ENFILTRO.C)
/#GrPrint:Imprime la pantalla actual del monitor (PANTALLA.C)t/

```

```

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
ENMENU.C
Archivo principal de ENTRADA.EXE
*/

```

```

#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include "proyfil.h"
#include <graphics.h>
#include <stdio.h>
float AfpeA,AfpeB,AfreA,AfreB,AfpbA,AfpbB,AfrbA,AfrbB, fpeA, fpeB, freA, freB, fpbA, fpbB, frbA, frbB;
int Dn,Ann,Bn,An,Bn,DAnn,DAn,DBn,DBn,BAnn,BAn,BBn,BBn;
float Aho,A0[6],AB[6],AC[6],As0,Aho,BA0[6],BB[6],BC[6],Bs0,BA0[5],DA1[5],DA2[5],DB0[5],DB1[5],DB2[5];
float Dha,Dhb,fsea,fseb,fsba,fsbb,fsbana,bsbanb,AfpbanA,AfpbanB,AfrbanA,AfrbanB,pra,prb;
float frbanA,frbanB,AW,BW,Afo,Bfo,fsbana,fsbanh,ncb,ff,XO,ff,nc;
float AH0[5],Ah1[5],Ah2[5],Ah3[5],Ah4[5],AH0[5],AH1[5],AH2[5],AH3[5],AH4[5];
float Bh0[5],Bh1[5],Bh2[5],Bh3[5],Bh4[5],BH0[5],BH1[5],BH2[5],BH3[5],BH4[5];
float Aj2,Aj0,Aj1,Aj2,Aj3,Aj4,Bj2,Bj0,Bj1,Bj2,Bj3,Bj4,BAho,BPho,Ofo,Opend,Oini,Dfp,Dfr,EPS,OO1;
int p_fpban,nbucle,np1lp2,p_entrada,p_fpb,p_vco,p_def,lp_entrada,lp_fpb,lp_vco;
int H2,in[20],c1,c2,decision_banda,p*1,*ppcvadro,lp_def,lp_fpban;
char decision_bajos;
extern float MaxX,MaxY,Esca,Escy,R10,R11;

```

```

*MAIN:función principal de ENTRADA.EXE

```

```

*Parámetros: Ninguno

```

```

*Retorno : Ninguno

```

```

#####

```

```

void main(void)

```

```

{

```

```

char dec;

```

```

int eleccion,r,il,op;

```

```

float co,eoscn,eoscn2n;

```

```

FILE *fp;

```

```

lp_entrada=0;lp_fpb=0;lp_vco=0;lp_def=0; lp_fpban=0;

```

```

initialize();

```

```

fp=fopen("a","rb");

```

```

rewind(fp);

```

```

fread(&nbucle,sizeof(int),1,fp);

```

```

fclose(fp);

```

```

cleardevice();

```

```

fp=fopen("c","rb");

```

```

rewind(fp);

```

```

fread(&p_entrada,sizeof(int),1,fp);

```

```

fread(&p_fpb,sizeof(int),1,fp);

```

```

fread(&p_vco,sizeof(int),1,fp);

```

```

fread(&p_def,sizeof(int),1,fp);

```

```

fread(&p_fpban,sizeof(int),1,fp);

```

```

fclose(fp);

```

```

if(p_fpb !=0){

```

```

fp=fopen("j","rb");

```

```

rewind(fp);

```

```

fread(&in,sizeof(int),20,fp);

```

```

fread(&H2,sizeof(int),1,fp);

```

```

fread(&ff,sizeof(float),1,fp);

```

```

fread(&ncb,sizeof(float),1,fp);

```

```

fread(&OO1,sizeof(float),1,fp);

```

```

fclose(fp);nc=ncb*H2;

```

```

}

```

```

if(nbucle !=10 &&nbucle !=11)

```

```

{

```



```

REll:switch(obucle){
    case 2:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(15*Escy,1,"BLOQUE DE COSTAS GENERAL ");
        formaabuc2(-30*Escx,43*Escy);
        QGRPLL(Escx*420,Escy*80,0);
        break;
    case 5:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(20*Escx,1,"BLOQUE DE COSTAS CON DECISION DENTRO DE UN BRAZO");
        formaabuc5(-40*Escx,50*Escy);
        QGRPLL(Escx*440,Escy*70,0);
        break;
    case 6:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(20*Escx,1,"BLOQUE DE COSTAS NUMERICO");
        formaabuc6(-40*Escx,60*Escy);
        QGRPLL(Escx*440,Escy*70,0);
        break;
    case 0:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(Escy*20,2,"BLOQUE IGUALADOR DE FASE [ FLL ]");
        forma(-80,50*Escy);
        QGRPLL(Escx*440,Escy*70,0);
        break;
    case 3:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(70*Escy,1,"BUCLE CON REMODULACION POR EL OSCILADOR LOCAL ");
        formaabuc3(-50*Escx,45*Escy);
        QGRPLL(Escx*470,Escy*70,0);
        break;
    case 4:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(17*Escy,1,"BUCLE CON REMODULACION POR LA SENAL RECIBIDA ");
        formaabuc4(-40*Escx,45*Escy);
        QGRPLL(Escx*480,Escy*70,0);
        break;
    case 1:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(20*Escx,1,"BLOQUE CON MULTIPLICACION DE FRECUENCIA");
        formaabuc1(-45*Escx,40*Escy);
        QGRPLL(Escx*440,Escy*80,0);
        break;
    case 7:
        barra1(5*Escx,10*Escy,MaxX-30*Escx,187*Escy,20*Escx,11,1);
        lema(20*Escx,1,"BLOQUE DE COSTAS CON REMODULACION PARA RFSK");
        formaabuc7(-45*Escx,45*Escy);
        QGRPLL(Escx*480,Escy*60,0);
        break;
}
pe4=0;
do
{
it=menu_secundario(13*Escx,-14*Escy);
switch(it){
    case 1:
        dec= entrada();
        if(dec!=0) lp_entrada=1;
        borrar(1);break;
    case 2:
        dec=estructurar_filtro();
        if(dec !=0)lp_fpb=1;
}
}

```

```

        borrar(1);break;
    case 3:
        dec=estructurar_vco();
        if (dec !=0)lp_vco=1;
        borrar(1);break;
    case 5:goto SAL1;
    case 6:if(nbucle !=1) {
        dec=estructurar_defasador();
        borrar(1);
        if(dec !=0)lp_def=1;
        }
        else {
        dec=estructurar_banda();
        borrar(1);
        if(dec !=0)lp_fpban=1;
        }
    }
    if(p_entrada == 0 && it ==4){
        mensaje(180+Escy,52,0,23, "E R R O R ! NO se ha estructurado la ONDA de ENTRADA al bucle");
        borrar(1);continue;
    }
    if(p_fpb == 0 && it ==4){
        mensaje(180+Escy,51,0,23, "E R R O R ! NO se ha estructurado el FILTRO PASABAJOS ");
        borrar(1);continue;
    }
    if( p_vco == 0 && it== 4){
        mensaje(180+Escy,51,0,23, "E R R O R ! NO se ha estructurado el OSCILADOR ( V. C. O )");
        borrar(1);continue;
    }
    if(nbucle == 1 && p_fpban == 0 && it ==4){
        mensaje(180+Escy,51,0,23, "E R R O R ! NO se ha estructurado el DUPLICADOR DE FRECUENCIA");
        borrar(1);continue;
    }
    if((nbucle == 2 ;;nbucle == 5 ;;nbucle == 6 ;;nbucle == 7) && p_def == 0 && it== 4){
        mensaje(180+Escy,51,0,23, "E R R O R ! NO se ha estructurado el DEFASADOR de 90 grados");
        borrar(1);continue;
    }
}while( (p_entrada != 0 && p_fpb != 0 && p_vco != 0 &&
( (nbucle == 2 ;;nbucle == 5 ;;nbucle == 6 ;;nbucle == 7) && p_def!=0)
|| (nbucle == 1 && p_fpban!=0) ;; nbucle==0 ;; nbucle==3 ;; nbucle==4 ) ) ;; it!=4 );
if(lp_fpb !=0){
    if (cfl=='E';;cfl=='e')
        elfiltro(10,1);
    else
        elfillrob(10,1);
    if(nbucle !=0 && nbucle !=3 && nbucle !=4){
        if(decision_bajos==67){
            if (cf2=='E';;cf2=='e')
                elfiltro(10,2);
            else
                elfillrob(10,2);
        }
        if(decision_bajos==68){
            cf2=cfl;
            if (cfl=='E';;cfl=='e')
            {
                r=Ann==0?An/2:(An-1)/2;
                Bnn=Ann;Bn=An;Bho=Aho;Bs0=As0;fseb=fsea;
                for(it=1;it<r+1;it++)
                {
                    PAD[it]=AAQ[it];
                    DC[it]=AC[it];
                }
            }
        }
    }
}

```



```

fwrite(&Dha,sizeof(float),1,fp);
fwrite(&Dhb,sizeof(float),1,fp);
fwrite(&Aa0,sizeof(float),1,fp);
fwrite(&Bs0,sizeof(float),1,fp);
fwrite(&pra,sizeof(float),1,fp);
fwrite(&prb,sizeof(float),1,fp);
fwrite(&fsea,sizeof(float),1,fp);
fwrite(&fseb,sizeof(float),1,fp);
fwrite(&fsba,sizeof(float),1,fp);
fwrite(&fsbb,sizeof(float),1,fp);
fwrite(&AA0,sizeof(float),6,fp);
fwrite(&AB0,sizeof(float),6,fp);
fwrite(&AC,sizeof(float),6,fp);
fwrite(&BA0,sizeof(float),6,fp);
fwrite(&BB,sizeof(float),6,fp);
fwrite(&BC,sizeof(float),6,fp);
fwrite(&DA0,sizeof(float),5,fp);
fwrite(&DA1,sizeof(float),5,fp);
fwrite(&DA2,sizeof(float),5,fp);
fwrite(&DB0,sizeof(float),5,fp);
fwrite(&DB1,sizeof(float),5,fp);
fwrite(&DB2,sizeof(float),5,fp);
fclose(fp);
}
if(lp_fpban !=0){
fp=fopen("g","wb+");
fwrite(&BAnn,sizeof(int),1,fp);
fwrite(&BAa,sizeof(int),1,fp);
fwrite(&BBnn,sizeof(int),1,fp);
fwrite(&BBn,sizeof(int),1,fp);
fwrite(&Aj2,sizeof(float),1,fp);
fwrite(&Aj0,sizeof(float),1,fp);
fwrite(&Aj1,sizeof(float),1,fp);
fwrite(&Aj2,sizeof(float),1,fp);
fwrite(&Aj3,sizeof(float),1,fp);
fwrite(&Aj4,sizeof(float),1,fp);
fwrite(&Kj2,sizeof(float),1,fp);
fwrite(&Bj0,sizeof(float),1,fp);
fwrite(&Bj1,sizeof(float),1,fp);
fwrite(&Bj2,sizeof(float),1,fp);
fwrite(&Bj3,sizeof(float),1,fp);
fwrite(&Bj4,sizeof(float),1,fp);
fwrite(&BAho,sizeof(float),1,fp);
fwrite(&BBho,sizeof(float),1,fp);
fwrite(&fsbana,sizeof(float),1,fp);
fwrite(&fsbanb,sizeof(float),1,fp);
fwrite(&Ah0,sizeof(float),5,fp);
fwrite(&Ah1,sizeof(float),5,fp);
fwrite(&Ah2,sizeof(float),5,fp);
fwrite(&Ah3,sizeof(float),5,fp);
fwrite(&Ah4,sizeof(float),5,fp);
fwrite(&Bh0,sizeof(float),5,fp);
fwrite(&Bh1,sizeof(float),5,fp);
fwrite(&Bh2,sizeof(float),5,fp);
fwrite(&Bh3,sizeof(float),5,fp);
fwrite(&Bh4,sizeof(float),5,fp);
fwrite(&AH0,sizeof(float),5,fp);
fwrite(&AH1,sizeof(float),5,fp);
fwrite(&AH2,sizeof(float),5,fp);
fwrite(&AH3,sizeof(float),5,fp);
fwrite(&AH4,sizeof(float),5,fp);
fwrite(&BH0,sizeof(float),5,fp);
}

```

```

        fwrite(&H1, sizeof(float), 5, fp);
        fwrite(&H2, sizeof(float), 5, fp);
        fwrite(&H3, sizeof(float), 5, fp);
        fwrite(&H4, sizeof(float), 5, fp);
        fclose(fp);
    }
    if(lp_def != 0){
        fp=fopen("h", "wb+");
        fwrite(&Dfr, sizeof(float), 1, fp);
        fwrite(&Dfp, sizeof(float), 1, fp);
        fwrite(&EPG, sizeof(float), 1, fp);
        fwrite(&Dn, sizeof(int), 1, fp);
        fclose(fp);
    }
    if(lp_vco != 0){
        fp=fopen("i", "wb+");
        fwrite(&fo, sizeof(float), 1, fp);
        fwrite(&ini, sizeof(float), 1, fp);
        fwrite(&pend, sizeof(float), 1, fp);
        fclose(fp);
    }
    if(lp_entrada != 0){
        fp=fopen("j", "wb+");
        fwrite(&in, sizeof(int), 20, fp);
        fwrite(&N2, sizeof(int), 1, fp);
        fwrite(&ff, sizeof(float), 1, fp);
        fwrite(&nch, sizeof(float), 1, fp);
        fwrite(&O1, sizeof(float), 1, fp);
        fclose(fp);
    }
}
    if(nbucle==10 ;; nbucle==11){
        estructurar_filtro();
        it=5;
    }
    SALL: fp=fopen("d", "wb+");
    fwrite(&it, sizeof(int), 1, fp);
    fclose(fp);
}

```

\*BORRAR: Borra las ventanas de texto creadas para el ingreso de datos

\*Parametros: caso.- Indica que ventana es la borrada

\*Retorno : Ninguno

\*\*\*\*\*

void borrar(int caso)

```

{
    int i,j;
    switch(caso){
        setviewport(0,0,MaxX,MaxY,1);
        case 1: setviewport(14*Escx, 193*Escy, MaxX, MaxY,1);
                clearviewport();break;
        case 2: setviewport(146*Escx,199*Escy,649*Escx,306*Escy,1);
                clearviewport();break;
        case 4: setviewport((MaxX-24*Escx)/2-5*textwidth("N"),271*Escy,(MaxX-24*Escx)/2+15*textwidth("N"),288*Escy,1);
                clearviewport();break;
        case 5: setviewport((MaxX-24*Escx)/2+18*textwidth("N"),267*Escy,(MaxX-24*Escx)/2+29*textwidth("N"),288*Escy,1);
                clearviewport();break;
        case 6: setviewport((MaxX-24*Escx)/2-5*textwidth("N"),272*Escy,(MaxX-24*Escx)/2+15*textwidth("N"),288*Escy,1);
                clearviewport();break;
    }
    setviewport(0,0,MaxX,MaxY,1);
    setfillstyle (SOLID_FILL, getmaxcolor());
}

```

```

        setcolor(getmaxcolor());
        settextjustify( LEFT_TEXT, TOP_TEXT );
        settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        setviewport(0,0,MaxX,MaxY,1);
    }

/*AK: Integral ELIPTICO completo de primer orden
*Parametros: Y .- Constante del integral
*Retorno : Valor del integral
*****/
float AK(float Y)
{
    float X,res2;
        X=pow(Y,2);
        res2=PI*(.5+.125*X+.0703125*pow(X,7)+.48828125*pow(X,3)+.037384*pow(X,4));
        return res2;
}

/*RESISTENCIA: Presenta una resistencia en pantalla
* Parametros: x .- coordenada X de inicio
*             y .- coordenada Y de inicio
*             espx .- ancho de la resistencia
*             espy .- altura de la resistencia
*             vh .- Indica si es horizontal o vertical
* Retorno : Ninguno
*****/
void resistencia(int x,int y,int espx,int espy,int vh)
{
    setlinestyle(0,1,THICK_WIDTH);
    setcolor(getmaxcolor());
    switch(vh){
        case 1: line(x,y,x+espx,y-espy);
                line(x+espx,y-espy,x+2*espx,y+espy);
                line(x+2*espx,y+espy,x+3*espx,y);
                break;
        case 2: line(x,y,x+espx,y+espy);
                line(x+espx,y+espy,x-espx,y+2*espy);
                line(x-espx,y+2*espy,x,y+3*espy);
    }
    setlinestyle(0,1,NORM_WIDTH);
}

/*AMPLIFICADOR: Presenta un amplificador operacional en pantalla
* Parametros: x .- coordenada X de inicio
*             y .- coordenada Y de inicio
*             espx .- ancho de la resistencia
*             espy .- altura de la resistencia
*             vh .- Indica si es horizontal o vertical
* Retorno : Ninguno
*****/
void amplificador(int x,int y,int espx,int espy)
{
    setlinestyle(0,1,THICK_WIDTH);
    setcolor(getmaxcolor());
    line(x,y-3*espy,x,y+espy);
    line(x,y-3*espy,x+5*espx,y-espy);
    line(x,y+espy,x+5*espx,y-espy);
    setlinestyle(0,1,NORM_WIDTH);
}

/*CAPACITOR: Presenta un capacitor en pantalla
* Parametros: x .- coordenada X de inicio

```

```

*      y .- coordenada Y de inicio
*      espx .- ancho de la resistencia
*      espy .- altura de la resistencia
* Retorno : Ninguno
#####/
void capacitor(int x,int y,int espx,int espy,int hv)
{
    sellinestyle(0,1,THICK_WIDTH);
    setcolor(getmaxcolor());
    switch(hv){
        case 1: line(x,y-espy,x,y+espy);
                line(x+espx,y-espy,x+espx,y+espy);
                break;
        case 2: line(x-espx,y,x+espx,y);
                line(x-espx,y+espy,x+espx,y+espy);
    }
    sellinestyle(0,1,NORM_WIDTH);
}

/*PUNTO:Presenta un punto de conexión
* Parametros: x .- coordenada X del punto
*             y .- coordenada Y del punto
* Retorno : Ninguno
#####/
void punto(int x,int y)
{
    sellinestyle(0,1,THICK_WIDTH);
    setcolor(getmaxcolor());
    circle(x,y,2);
    sellinestyle(0,1,NORM_WIDTH);
}

/*FILTRO_E1:Presenta un filtro de primer orden en pantalla
* Parametros: x .- coordenada X de inicio
*             y .- coordenada Y de inicio
*             espx .- ancho de la resistencia
*             espy .- altura de la resistencia
* Retorno : Ninguno
#####/
void filtro_e1(int x, int y ,int espx,int espy)
{
    line(x,y,x+10*espx,y);
    resistencia(x+10*espx,y,espx,espy,1);
    outtextxy(x+9*espx,y-3*espy,"R0");
    line(x+13*espx,y,x+28*espx,y);
    line(x+23*espx,y,x+23*espx,y+7*espy);
    capacitor(x+23*espx,y+7*espy,espx,espy,2);
    outtextxy(x+18*espx,y+7*espy,"C1");
    line(x+23*espx,y+8*espy,x+23*espx,y+15*espy);
    line(x,y+15*espy,x+28*espx,y+15*espy);
    punto(x,y);
    punto(x+28*espx,y);
    punto(x,y);
    punto(x+28*espx,y+15*espy);
    punto(x,y+15*espy);
}

/*FILTRO_E2:Presenta un filtro de segundo orden en pantalla
* Parametros: x .- coordenada X de inicio
*             y .- coordenada Y de inicio
*             espx .- ancho de la resistencia
*             espy .- altura de la resistencia

```

```

† Retorno : Ninguno
#####/
void filtro_b2(int x, int y, int espx, int espy)
{
    amplificador(x+19*espx,y,espx,espy);
    line(x,y,x+2*espx,y);
    resistencia(x+2*espx,y,espx,espy,1);
    outtextxy(x+espx,y-3*espy,"R0");
    line(x+5*espx,y,x+9*espx,y);
    resistencia(x+9*espx,y,espx,espy,1);
    outtextxy(x+9*espx,y-3*espy,"R0");
    line(x+12*espx,y,x+19*espx,y);
    line(x+24*espx,y-espy,x+26*espx,y-espy);
    line(x+26*espx,y,x+28*espx,y);
    line(x+26*espx,y,x+26*espx,y-espy);
    line(x+14*espx,y,x+14*espx,y+7*espy);
    capacitor(x+14*espx,y+7*espy,espx,espy,2);
    line(x+14*espx,y+9*espy,x+14*espx,y+15*espy);
    if(nbucle==11){
        line(x+7*espx,y,x+7*espx,y-11*espy);
        line(x+7*espx,y-11*espy,x+15*espx,y-11*espy);
        capacitor(x+15*espx,y-11*espy,espx,espy,1);
        outtextxy(x+14*espx,y-9*espy,"C1");
        line(x+16*espx,y-11*espy,x+26*espx,y-11*espy);
        outtextxy(x+9*espx,y+7*espy,"C2");
        punto(x+7*espx,y);
    }
    if(nbucle==10){
        line(x+8*espx,y,x+8*espx,y-8*espy);
        capacitor(x+8*espx,y-9*espy,espx,espy,2);
        outtextxy(x+10*espx,y-9*espy,"2C1");
        line(x+8*espx,y-9*espy,x+8*espx,y-11*espy);
        line(x+6*espx,y+4*espy,x+6*espx,y-3*espy);
        resistencia(x+6*espx,y-6*espy,espx,espy,2);
        outtextxy(x+2*espx,y-6*espy,"R1");
        line(x+6*espx,y-6*espy,x+6*espx,y-11*espy);
        line(x+6*espx,y-11*espy,x+26*espx,y-11*espy);
        line(x,y+4*espy,x+3*espx,y+4*espy);
        capacitor(x+3*espx,y+4*espy,espx,espy,1);
        outtextxy(x+2*espx,y+6*espy,"C1");
        line(x+4*espx,y+4*espy,x+10*espx,y+4*espy);
        capacitor(x+10*espx,y+4*espy,espx,espy,1);
        outtextxy(x+9*espx,y+6*espy,"C1");
        line(x+11*espx,y+4*espy,x+14*espx,y+4*espy);
        line(x,y,x,y+4*espy);
        outtextxy(x+9*espx,y+8*espy,"C2");
        punto(x+8*espx,y);
        punto(x+8*espx,y-11*espy);
        punto(x+6*espx,y+4*espy);
        punto(x+14*espx,y+4*espy);
    }
    line(x,y+15*espy,x+28*espx,y+15*espy);
    line(x+17*espx,y-6*espy,x+17*espx,y+6*espy);
    resistencia(x+17*espx,y+6*espy,espx,espy,2);
    outtextxy(x+20*espx,y+7*espy,"R");
    line(x+17*espx,y+9*espy,x+17*espx,y+15*espy);
    line(x+17*espx,y-6*espy,x+20*espx,y-6*espy);
    resistencia(x+20*espx,y-6*espy,espx,espy,1);
    outtextxy(x+21*espx,y-9*espy,"r");
    line(x+23*espx,y-6*espy,x+26*espx,y-6*espy);
    line(x+26*espx,y-espy,x+26*espx,y-11*espy);
    line(x+17*espx,y-2*espy,x+19*espx,y-2*espy);
}

```



```

    punto(x,y);
    punto(x+28*espx,y);
    punto(x+28*espx,y+15*espy);
    punto(x,y+15*espy);
    punto(x+14*espx,y);
    punto(x+17*espx,y-2*espy);
    punto(x+26*espx,y-6*espy);
    punto(x+26*espx,y-espy);
    punto(x+14*espx,y+15*espy);
    punto(x+17*espx,y+15*espy);
}

/*FILTRO_E2:Presenta los valores de las constantes de los filtros determinados
# Parámetros: x .- coordenada X de inicio
#             y .- coordenada Y de inicio
#             espx .- ancho de la resistencia
#             espy .- altura de la resistencia
# Retorno : Ninguno
#####
void escribir(int x,int y,int espx,int lipo,int i)
{
    y-=espy;
    if(lipo==1){
        if(nbucl==10)qprintf("%x,%y,2,% .3E",As0);
        if(nbucl==11)qprintf("%x,%y,2,% .3E",prb);
        y+=5*espy;
        qprintf("%x,%y,2,% .2E",R10);
    }
    if(lipo==2){
        if(nbucl==10){
            qprintf("%x,%y,2,% .3E",AA0[i+1]);
            y+=espy;
            qprintf("%x,%y,2,% .3E",AB0[i+1]);
            y+=espy;
            qprintf("%x,%y,2,% .3E",R11);
            y+=espy;
            qprintf("%x,%y,2,% .3E",AC[i+1]);
            y+=espy;
            qprintf("%x,%y,2,% .3E",AA0[i+1]);
        }
        if(nbucl==11){
            qprintf("%x,%y,2,% .3E",DB0[i]);
            y+=espy;
            qprintf("%x,%y,2,% .3E",DB1[i]);
            y+=espy;
            qprintf("%x,%y,2,% .3E",R11);
            y+=5*espy;
            qprintf("%x,%y,2,% .3E",DA0[i]);
        }
    }
}

}

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
MENUSEC.C
Presenta y controla el menu secundario
*/

```

```

#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>

```

```

#include <stdio.h>
int #ppacvadro;
extern float MaxX,MaxY,Escx,Escy;
void cuadro_aenu(float x,float y);
extern char far #buf;
extern float Escx,Escy;
extern int nbucle,pa4;

/*MENU_SECUNDARIO:Presenta y controla el menu secundario
# Parametros : x.- coordenada X base
#             y.- coordenada Y base
# Retorno : # de la eleccion
#####/
int menu_secundario(int x,int y)
{
size_t sscuadro;
int ch2,espx,espx1,espy,a,n=7,numeros[7],indl,y1,x1;
float yin;
numeros[1] = 1;
numeros[2] = 2;
numeros[3] = 3;
if(nbucle == 0 || nbucle == 3 || nbucle == 4)numeros[4] = 4;
else numeros[4] = 6;
if(nbucle == 0 || nbucle == 3 || nbucle == 4)numeros[5] = 5;
else numeros[5] = 4;
numeros[6] = 5;
espx1=linewidth("H");a=(MaxX-24*Escx)/2;x1=a-16*linewidth("H");
switch(nbucle){
case 0:
case 3:
case 4: yin=-y+Escy*211;y1=2.4*linewidth("H");
outtextxy(x1,yin, "1. CARACTERISTICAS DE LA ENTRADA");
outtextxy(x1,yin+y1, "2. ESTRUCTURAR FILTRO PASABAJOS");
outtextxy(x1,yin+2*y1, "3. ESTRUCTURAR V.C.D.");
outtextxy(x1,yin+3*y1, "4. GRAFICAR");
outtextxy(x1,yin+4*y1, "5. MENU PRINCIPAL");
break;
case 1: yin=-y+Escy*200;y1=2*linewidth("H");
outtextxy(x1,yin, "1. CARACTERISTICAS DE LA ENTRADA");
outtextxy(x1,yin+y1, "2. ESTRUCTURAR FILTRO PASABAJOS");
outtextxy(x1,yin+2*y1, "3. ESTRUCTURAR V.C.D.");
outtextxy(x1,yin+3*y1, "4. EST. DUPLICADOR de FRECUENCIA");
outtextxy(x1,yin+4*y1, "5. GRAFICAR");
outtextxy(x1,yin+5*y1, "6. MENU PRINCIPAL");
break;
case 2:
case 5:
case 6:
case 7: yin=-y+Escy*200;y1=2*linewidth("H");
outtextxy(x1,yin, "1. CARACTERISTICAS DE LA ENTRADA");
outtextxy(x1,yin+y1, "2. ESTRUCTURAR FILTRO PASABAJOS");
outtextxy(x1,yin+2*y1, "3. ESTRUCTURAR V.C.D.");
outtextxy(x1,yin+3*y1, "4. ESTRUCTURAR DEFASADOR");
outtextxy(x1,yin+4*y1, "5. GRAFICAR");
outtextxy(x1,yin+5*y1, "6. MENU PRINCIPAL");
break;
}
setcolor(0);
setlinestyle(SOLID_LINE,1,4);
settextstyle(0,0,0);
if(nbucle ==0 || nbucle==3 || nbucle==4){
bar (a-35*espx1,209*Escy+6*y1+4,a+35*espx1,209*Escy+7*y1);

```

```

line(a-35*espx1,209*Escy+64*y1+5,a+35*espx1,209*Escy+64*y1+5);
outtextxy(a-33*espx1,209*Escy+64*y1+5+textheight("N")/2,"ELIGIR OPCION CON EL CURSOR O PRESIONANDO EL
NUMERO CORRESPONDIENTE");
}
else {
bar (a-35*espx1,210*Escy+7*y1+4,a+35*espx1,210*Escy+8*y1+2);
line(a-35*espx1,210*Escy+7*y1+5,a+35*espx1,210*Escy+7*y1+5);
outtextxy(a-33*espx1,210*Escy+7*y1+5+textheight("N")/2,"ELIGIR OPCION CON EL CURSOR O PRESIONANDO EL
NUMERO CORRESPONDIENTE");
}
}
selcolor(12);
sellinesstyle(0,1,3);
if(nbucle==0 || nbucle==3 || nbucle==4) {
rectangle(a-21*espx1,205*Escy,a+21*espx1,209*Escy+6*y1);
rectangle(a-20*espx1,209*Escy,a+20*espx1,205*Escy+6*y1);
}
if(nbucle !=0 && nbucle !=3 && nbucle !=4){
rectangle(a-21*espx1,205*Escy,a+21*espx1,209*Escy+7*y1);
rectangle(a-20*espx1,209*Escy,a+20*espx1,205*Escy+7*y1);
}
rectangle(10*Escx,195*Escy,MaxX-30*Escx,MaxY);
rectangle(134*Escx,197*Escy,MaxX-33*Escx,MaxY-2*Escy);
selfillstyle(EMPTY_FILL,0);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
bar (a-7*textwidth("N"),200*Escy,a+7*textwidth("N"),210*Escy);
outtextxy(a-6*textwidth("N"),198*Escy,"MENU SECUNDARIO");
selfillstyle(SOLID_FILL,getmaxcolor());
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
sellinesstyle(0,0,HORIZ_DIR);
if(pn4==0)
ind1 = 1;
else
ind1=pn4;

espx=39*espx1;espy=textheight("N")+10;
escuadro = imagesize (0,0,espx,espy);
ppacuadro = _graphgetmen (escuadro);
selfillstyle (SOLID_FILL, getmaxcolor());
selcolor (getmaxcolor());
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
selcolor(1);
yin=yin-textheight("N")/2-2;ch2=320;
do
{
if (ch2 == 328 || ch2 == 331 || ch2 == 333 || ch2 == 336)
switch (ind1)
{
case 1: cuadro_menu(a-10*espx1,yin);break;
case 2: cuadro_menu(a-18*espx1,yin+1);break;
case 3: cuadro_menu(a-18*espx1,yin+2);break;
case 4: cuadro_menu(a-18*espx1,yin+3);break;
case 5: cuadro_menu(a-18*espx1,yin+4);break;
case 6: cuadro_menu(a-18*espx1,yin+5);break;
}
}
ch2 = tecla();
if (ch2 == 328 || ch2 == 331 || ch2 == 333 || ch2 == 336 || (ch2>48 && ch2<55) ||ch2==13)
{
switch (ind1)
{
case 1: pulinaga(a-10*espx1,yin,ppacuadro, COPY_FUT);break;
case 2: pulinaga(a-18*espx1,yin+1,ppacuadro, COPY_FUT);break;

```

```

        case 3: putimage(a-18*espxl,yin+2*yl,ppcuardro, COPY_PUT);break;
        case 4: putimage(a-18*espxl,yin+3*yl,ppcuardro, COPY_PUT);break;
        case 5: putimage(a-18*espxl,yin+4*yl,ppcuardro, COPY_PUT);break;
        case 6: putimage(a-18*espxl,yin+5*yl,ppcuardro, COPY_PUT);break;
    }
    if(ch2 == '1' || ch2 == '2' || ch2 == '3' || ch2 == '4' || ch2 == '5' || ch2 == '6')
        ind1=ch2-48;
    if(ch2 == 328 || ch2 == 333){
        ind1--;
        if(ind1<1)
            ind1=1;
    }
    if(ch2 == 331 || ch2 == 336){
        if (nbucle==0 || nbucle==3 || nbucle==4){
            if (ind1 == 5 )
                ind1 = 1;
            else ind1++;
        }
        else {
            if (ind1 == 6)
                ind1 = 1;
            else ind1++;
        }
    }
}
} while (ch2 != '\r' && ch2 != '1' && ch2 != '2' && ch2 != '3' && ch2 != '4' && ch2 != '5' &&
(ch2 != '6' !(nbucle == 0 ||nbucle == 3 ||nbucle == 4)));
switch (ind1)
{
    case 1: cuadro_menu(a-18*espxl,yin);break;
    case 2: cuadro_menu(a-18*espxl,yin+yl);break;
    case 3: cuadro_menu(a-18*espxl,yin+2*yl);break;
    case 4: cuadro_menu(a-18*espxl,yin+3*yl);break;
    case 5: cuadro_menu(a-18*espxl,yin+4*yl);break;
    case 6: cuadro_menu(a-18*espxl,yin+5*yl);break;
}
pa4=ind1;
_graphfreeeen (ppcuardro, sscuardro);
return numeros[ind1];
}

/*CUADRO: forma el relleno de la opción actual para menu_superior
# Parámetros : x1 .- coordenada X base
#              y1 .- coordenada Y base
# Retorno : Ninguno
#####/
void cuadro_menu(float x1,float y1)
{
    int espx,espy;
    size_t sscuardro;
    espx=36*textwidth("M");
    espy=textheight("H")+10;
    getimage (x1,y1, x1+espx,y1+espy, ppcuardro);
    setfillstyle (SOLID_FILL, 0);
    bar (x1,y1, x1+espx,y1+espy);
    setcolor (getmaxcolor());
    rectangle (x1+2, y1+2, x1+espx-2, y1+espy-2);
    setfillstyle (SOLID_FILL, getmaxcolor());
    bar (x1+4, y1+4, x1+espx-4, y1+espy-4);
    putimage (x1,y1, ppcuardro, XOR_PUT);
    setcolor (0);
}

```

/\*

Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.

ENTRADAS.C

Archivo que presenta ventanas de entradas de datos

\*/

```

#include "math.h"
#include "stdio.h"
#include "conio.h"
#include "proyfil.h"
#include "stdlib.h"
#include "string.h"
#include "graphics.h"
extern float Dfp,Dfr,EFS,R10,R11,Afo,Upend,Oni;
extern int npl,in[20],cf1,cf2,decision_banda,nbucle;
extern char decision_bajos;
extern int p_entrada,p_fpb,p_vco,p_def, p_fphan;
extern float AfpbanA,AfpbanB,AfrbanA,AfrbanB,frbanA,frbanB,AW,BW,Afo,Rfo;
extern float AfpA,AfpB,AfrA,AfrB,AfpbA,AfpbB,AfrbA,AfrbB;
extern float fpeA,fpeB,freA,freB,fpbA,fpbB,frbA,frbB;
extern float Escx,Escy,MaxY,MaxX,W0,ff,az,dfp,dfr,EPS;
extern int BA0,BA1,BA2,BA3,BA4,BA5,BA6,BA7,BA8,BA9,BA10,BA11,BA12;
extern float ncb,AA0[6],AB0[6],AC[6],As0,OO1,BA0[6],BB[6],CC[6];
extern float BA0[5],BA1[5],BA2[5],BA3[5],BA4[5],BA5[5],BA6[5],BA7[5],BA8[5],BA9[5],BA10[5],BA11[5],BA12[5],pra,prb;
int ch;
void punto(int x,int y);
void resistencia(int x,int y,int espx,int espy,int hv);
void capacitor(int x,int y,int espx,int espy,int hv);
void amplificador(int x,int y,int espx,int espy);
void filtro_el(int x,int y,int espx,int espy);
void filtro_b2(int x, int y ,int espx,int espy);
void escribir(int x,int y,int espy,int tipo,int i);

/*ESTRUCTURAR_BANDA:Entrada de datos para estructurar filtros pasabanda
† Parámetros: Ninguno
† Retorno : Indicador (Datos acoplados? si o no)
#####/
char estructurar_banda(void)
{
char ban;
int n,espx1;
n=(MaxX-24*Escx)/2;
espx1=lexwidth("M");
horrar(1);
setviewport(0,0,MaxX,MaxY,1);
setcolor(0);
setlinestyle(SOLID_LINE,1,4);
settextstyle(0,0,0);
bar(n-38*espx1,290*Escy,n+38*espx1,305*Escy);
line(n-38*espx1,292*Escy,n+38*espx1,292*Escy);
outtextxy(n-34*espx1,296*Escy,"[Esc] SALIR [F7] Dato ANTERIOR [F10] o [ENTER] Dato
POSTERIOR ");
setfillstyle (SOLID_FILL, getmaxcolor());
setcolor (getmaxcolor());
text(200*Escy,1, "Estructurando Filtro Pasabanda X 2 [ A ]");
settextjustify( LEFT_JEXT, TOP_JEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
rectangle(n-38*espx1, 195*Escy, n+38*espx1, MaxY-41*Escy);
image(n-38*espx1,7*Escy,n+38*espx1,190*Escy,4,1);
ban=entrada_banda(1);
if (ban==0)goto sal;
elfiltrobanda(0,1);
do{

```

```

        decision_banda=mensaje(292+Escy,50,5,6, " [F9]Estructurar el otro filtro [F10]Usar el
        mismo filtro");
    }while(decision_banda !=323 && decision_banda != 324);
    if(decision_banda==323){
        setviewport(0,0,MaxX,MaxY,1);
        setviewport(n-37*espx1, 199+Escy, n+37*espx1, 285+Escy,1);
        clearviewport();
        setfillstyle (SOLID_FILL, 0);
        setcolor (getmaxcolor());
        setlinestyle(0,0,HORN_WIDTH);
        setviewport(0,0,MaxX,MaxY,1);
        leaa(200+Escy,1, "Estructurando Filtro Pasabanda X 2 [ R ]");
        settextjustify( LEFT_TEXT, TOP_TEXT );
        settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        ban=entrada_banda(2);
        if(ban==0)goto sal;
        efillrqbanda(0,2);
    }

    p_fpban=1;
    iaagen(n-38*espx1,7+Escy,n+38*espx1,190+Escy,1,2);
    return 1;
    sal:iaagen(n-38*espx1,7+Escy,n+38*espx1,190+Escy,1,2);
    return 0;
}

```

```

/*ESTRUCTURAR_FILTRO:Entrada de datos para estructurar filtros pasabajos
# Parámetros: Ninguno
# Retorno : Indicador (Datos aceptados? si o no)
#####/
char estructurar_filtro(void)
{
    float CO;
    char linea[30],baj;
    char *pl;
    int op=0,n,nn,r,espx1,espx,espy,x,y,i,decision,n;
    ini:  if(nbucle==10 ;nbucle==11)cleardevice();
        n=(MaxX-24*Escy)/2;
        espx1=lexwidth("H");
        espx=15*espx1;
        espy=2*lexheight("H");
        borrar(1);
        setviewport(0,0,MaxX,MaxY,1);
        setcolor(0);
        setlinestyle(SOLID_LINE,1,4);
        settextstyle(0,0,0);
        bar (n-38*espx1,290+Escy,n+38*espx1,305+Escy);
        line(n-38*espx1,292+Escy,n+38*espx1,292+Escy);
        outtextxy(n-32*espx1,294+Escy,"[Esc] SALIR [F9] Dato ANTERIOR [F10] o [ENTER] Dato POSTERIOR ");
        setfillstyle (SOLID_FILL, getmaxcolor());
        setcolor (getmaxcolor());
        if(nbucle !=10 &&nbucle !=11)
            leaa(200+Escy,1, "Estructurando Filtro Pasabajos I [ F.P.B.I ]");
        else
            leaa(200+Escy,1, " Caracteristicas del Filtro Pasabajos ");
        settextjustify( LEFT_TEXT, TOP_TEXT );
        settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        rectangle(n-38*espx1, 195+Escy, n+38*espx1, MaxY-41+Escy);
        if(nbucle !=10 &&nbucle !=11){
            ayuda_linea(74,1,1);
            outtextxy (n-24*espx1, 225+Escy, "¿QUE FILTRO? [E] ELIPTICO [R] BUTTERWORTH");
            op = gread (n+21*espx1, 225+Escy, linea,0, 1, op,1,5);
            if(linea[1]=='\r')goto sal;
        }
}

```

```

cfl = linea[1];
if (cfl=='E' || cfl=='e'){
    imagen(n-38*espxl,7*Escy,n+38*espxl,190*Escy,2,1);
    baj=entrada_filtro(1);
    if(baj==0) goto sal;
    elfiltro(0,1);
}
else {
    imagen(n-38*espxl,7*Escy,n+38*espxl,190*Escy,3,1);
    baj=entrada_filtro(3);
    if(baj==0) goto sal;
    elfiltro(0,1);
}
}
if(nbucle ==10 || nbucle ==11){
    if (nbucle==10){
        cfl='e';
        imagen(n-38*espxl,7*Escy,n+38*espxl,190*Escy,2,1);
        baj=entrada_filtro(1);
        if(baj==0) goto sal;
        elfiltro(1,1);
        nn=0n;
        n=0n;
    }
    else {
        cfl='h';
        imagen(n-38*espxl,7*Escy,n+38*espxl,190*Escy,3,1);
        baj=entrada_filtro(3);
        if(baj==0) goto sal;
        elfiltro(1,1);
        nn=0n;
        n=0n;
    }
}
while (nn>0)nn=nn-2;
r=nn==0?n/2:(n-1)/2;
setviewport(0,0,MaxX,MaxY,1);
clearviewport();
n=(MaxX)/2;
rectangle(n-39.5*espxl,180*Escy,n+39.5*espxl, MaxY-25*Escy);
rectangle(0,0, MaxX,MaxY-20*Escy);
lema(180*Escy,1, "Componentes pasivos del filtro estructurado ");
selfillstyle (SQLID_FILL, getmaxcolor());
setcolor (getmaxcolor());
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
outtextxy(n-39*espxl,200*Escy, "ETAPAS      :");
outtextxy(n-39*espxl,220*Escy, "CAPACITORES :C1:");
if(n>1)outtextxy(n-39*espxl,240*Escy,"      C2:");
outtextxy(n-39*espxl,260*Escy, "RESISTENCIAS:R0:");
if(n>1)outtextxy(n-39*espxl,280*Escy,"      R1:");
outtextxy(n-39*espxl,300*Escy, "AMPLIFICACION:k:");
outtextxy(n-21*espxl,200*Escy,"primera");
if(n>2)outtextxy(n-21*espxl+13*espxl,200*Escy,"segunda");
if(n>4)outtextxy(n-21*espxl+13*2*espxl,200*Escy,"tercera");
if(n>6)outtextxy(n-21*espxl+13*3*espxl,200*Escy,"cuarta");
if(n>8)outtextxy(n-21*espxl+13*4*espxl,200*Escy,"quinta");
if(n==1)outtextxy(n-23*espxl,312*Escy,"ELEMENTO 1");
if(n>=2)outtextxy(n-23*espxl,312*Escy,"ELEMENTO 2");
if(n==3)outtextxy(n-23*espxl+13*espxl,312*Escy,"ELEMENTO 1");
if(n>=4)outtextxy(n-23*espxl+13*2*espxl,312*Escy,"ELEMENTO 2");
if(n==5)outtextxy(n-23*espxl+13*2*espxl,312*Escy,"ELEMENTO 1");
if(n>=6)outtextxy(n-23*espxl+13*2*espxl,312*Escy,"ELEMENTO 2");

```

```

if(n==7)outlextry(n-23*espx+13*3*espx,312*Escy,"ELEMENTO 1");
if(n==8)outlextry(n-23*espx+13*3*espx,312*Escy,"ELEMENTO 2");
if(n==9)outlextry(n-23*espx+13*4*espx,312*Escy,"ELEMENTO 1");
if(n==10)outlextry(n-23*espx+13*4*espx,312*Escy,"ELEMENTO 2");
switch(n){
    case 1:
    case 2:espx=13*Escx;espy=5*Escy;a=n-14*espx;break;
    case 3:
    case 4:espx=12*Escx;espy=5*Escy;a=n-28*espx;break;
    case 5:
    case 6:espx=8*Escx;espy=5*Escy;a=n-14*3*espx;break;
    case 7:
    case 8:espx=6*Escx;espy=5*Escy;a=n-14*4*espx;break;
    case 9:
    case 10:espx=5*Escx;espy=5*Escy;a=n-14*5*espx;break;
}
if(n>=2)outlextry(9*espx,180*Escy-lextheight("H"),"k=1+r/R");
if(n==1)outlextry(n+12*espx,180*Escy-2*lextheight("H"),"ELEMENTO 1");
if(n==2)outlextry(n+9*espx,180*Escy-2*lextheight("H"),"ELEMENTO 2");
if(n==3)outlextry(n+(9+28)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 1");
if(n==4)outlextry(n+(9+28)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 2");
if(n==5)outlextry(n+(9+28*2)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 1");
if(n==6)outlextry(n+(9+28*2)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 2");
if(n==7)outlextry(n+(9+28*3)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 1");
if(n==8)outlextry(n+(9+28*3)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 2");
if(n==9)outlextry(n+(9+28*4)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 1");
if(n==10)outlextry(n+(9+28*4)*espx,180*Escy-2*lextheight("H"),"ELEMENTO 2");
if(n==1)filtro_el(n,70*Escy,espx,espy);
if(n==2)filtro_b2(n,70*Escy,espx,espy);
if(n==3)filtro_el(n+28*espx,70*Escy,espx,espy);
if(n==4)filtro_b2(n+28*espx,70*Escy,espx,espy);
if(n==5)filtro_el(n+2*28*espx,70*Escy,espx,espy);
if(n==6)filtro_b2(n+2*28*espx,70*Escy,espx,espy);
if(n==7)filtro_el(n+3*28*espx,70*Escy,espx,espy);
if(n==8)filtro_b2(n+3*28*espx,70*Escy,espx,espy);
if(n==9)filtro_el(n+4*28*espx,70*Escy,espx,espy);
if(n==10)filtro_b2(n+4*28*espx,70*Escy,espx,espy);
if(nn !=0)mensaje(320*Escy,60,10,6, " Resistencia del ELEMENTO 1 (Se recomienda 10000)[Ohmios]");
if(n>1)mensaje(320*Escy,60,11,6, " Resistencia del ELEMENTO 2 (Se recomienda 10000)[Ohmios]");
if(nbucle==11){
    if(n>1){
        CO=1/(R11+2*PI*f*pbA);
        for(i=0;i<r;+i){
            DB0[i]=DΔ2[i]*CO;
            DB1[i]=DΔ1[i]*CO;
        }
        if(nn !=0)prb=pra/(2*PI*f*pbA*R10);
    }
}
if(nbucle==10){
    if(n>1){
        CO=1/(R11+2*PI*f*peA);
        for(i=1;i<r+1;+i){
            AΔ0[i]=BC[i]*CO;
            ABB[i]=BB[i]*CO;
            AC[i]=R11/2;
        }
        if(nn !=0)As0=(float)1/(As0+2*PI*f*peA*R10);
    }
}
n=MaxX/2;
if(n==1)escribir(n-22*espx,220*Escy,espy,1,0);

```



```

if(n>=2)escribir(a-22*esp1,220*Escy,espy,2,0);
if(n==3)escribir(a-22*esp1+13*esp1,220*Escy,espy,1,0);
if(n=4)escribir(a-22*esp1+13*esp1,220*Escy,espy,2,1);
if(n=5)escribir(a-22*esp1+2*13*esp1,220*Escy,espy,1,0);
if(n=6)escribir(a-22*esp1+2*13*esp1,220*Escy,espy,2,2);
if(n=7)escribir(a-22*esp1+3*13*esp1,220*Escy,espy,1,2);
if(n=8)escribir(a-22*esp1+3*13*esp1,220*Escy,espy,2,3);
if(n=9)escribir(a-22*esp1+4*13*esp1,220*Escy,espy,1,3);
if(n=10)escribir(a-22*esp1+4*13*esp1,220*Escy,espy,2,4);
do{
    decision=mensaje(322*Escy,56,5,2), "[F4]Menu Principal [F10]Otro filtro [F1]Ayuda
[F7]Imprimir";
    if(decision==315)ayuda(a-30*esp1,5*Escy,a+34*esp1,190*Escy,90);
    if(decision==324){
        imagen(a-38*esp1,7*Escy,a+38*esp1,190*Escy,1,2);
        goto ini;
    }
    if(decision == 321) BrPrint();
    }while( decision != 318 );
cleardevice();
}

if(nbucle != 0 && nbucle != 3 && nbucle != 4 && nbucle !=10 && nbucle !=11)
{
    do{
        decision_bajos=mensaje(292*Escy,50,5,6, " [F9]Estructurar el otro filtro [F10]Usar el
        mismo filtro");
    }while(decision_bajos !=67 && decision_bajos != 68);
    if(decision_bajos==67){
        selviewport(0,0,MaxX,MaxY,1);
        selviewport(a-37*esp1, 199*Escy, a+37*esp1, 285*Escy,1);
        clearviewport();
        selviewport(0,0,MaxX,MaxY,1);
        lrea(200*Escy,1, "Estructurando filtro Pasabajos 2 [ F.P.R.2 ]");
        selfillstyle (SOLID_FILL, getmaxcolor());
        setcolor (getmaxcolor());
        sellextjustify( LEFT_TEXT, TOP_TEXT );
        seltextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        ayuda_linea(74,1,1);
        outtextxy (a-24*esp1, 225*Escy, "QUE FILTRO? [E] ELIPTICO [R] BUTTERWORTH");
        op = grread (a+21*esp1, 225*Escy, linea,0, 1, op,1,5);
        if(linea[1]== '\x1B')goto sal;
        cf2 = linea[1];
        if (cf2=='E' || cf2=='e'){
            baj=entrada_filtro(2);
            if(baj==0)goto sal;
            elfiltro(0,2);
        }
        else {
            baj=entrada_filtro(4);
            if(baj==0)goto sal;
            elfiltroh(0,2);
        }
    }
}

p_fpb=1;
imagen(a-38*esp1,7*Escy,a+38*esp1,190*Escy,1,2);
return 1;
sal:if(cf1=='E' || cf1=='e' || cf1=='b' || cf1=='B'){
    imagen(a-38*esp1,7*Escy,a+38*esp1,190*Escy,1,2);
    cf1=0;
}
}

```

```

    return 0;
}

/*ESTRUCTURAR_VCO:Entrada de datos para estructurar el VCO
* Parámetros: Ninguno
* Retorno : Indicador (Datos acaplados? si o no)
#####/
char estructurar_vco(void)
{
int ind1=0,ind2=0,ind3=0,prin,y,a,espx1,op1=0,op2=0,op3=0;
char linea1[30],linea2[30],linea3[30],tp11,tp12,tp13,tp;
    n=(MaxX-24*Esca)/2;
    espx1=(textwidth("M"));
    prin=n-30*espx1;borrar(1);
    setviewport(0,0,MaxX,MaxY,1);
    setcolor(0);
    setlinestyle(SOLID_LINE,1,4);
    settextstyle(0,0,0);
    bar (n-32*espx1,290*Esca,n+32*espx1,305*Esca);
    line(n-32*espx1,292*Esca,n+32*espx1,292*Esca);
    outtextxy(n-31*espx1,296*Esca,"[Esc]SALIR  [F9]Dato ANTERIOR  [F10] o [ENTER]Dato POSTERIOR");
    setfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    text(200*Esca,1, "Estructurando el Oscilador Controlado por Volteje");
    settextjustify( LEFT_TEXT, TOP_TEXT );
    settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
    rectangle(n-32*espx1, 195*Esca, n+32*espx1, MaxY-41*Esca);
    dato1:do{
        p11=linea1;
        ayuda_linea(74,51,1);
        outtextxy (prin, 235*Esca, "Frecuencia CENTRAL de Oscilacion (fo)  [ Hz ] ");
        op1 = gread (n+20*espx1, 235*Esca, linea1,ind1, 9, op1,51,2);
        p11++;
        ind1=1;
        p=strstr(p11,"\\x1B");
        if(tp == "\\x1F")return 0;
        p=strstr(p11,"C");
        if(tp == 'C'){
            p11--;
            goto dato1;
        }
        Ofo = atof (p11);
        p=strstr(p11,"D");
        if(!tp == 'D' && op1==0 )Ofo==0)
            mensaje(205*Esca,54,0,23, " E R R O R !  NO existe dato para el Frecuencia CENTRAL");
    }while (op1 == 0 || Ofo==0) ;
    dato2:p12 = linea2;
    ayuda_linea(74,52,1);
    outtextxy (prin, 252*Esca, "SENSIBILIDAD (S)  [ Hz / grado ] ");
    op2 = gread (n+20*espx1, 252*Esca, linea2,ind2, 9, op2,52,2);
    p12++;
    ind2=1;
    p=strstr(p12,"\\x1B");
    if(tp == "\\x1F")return 0;
    Opend = atof (p12);
    p=strstr(p12,"C");
    if(tp == 'C'){
        p11--;
        goto dato1;
    }
    p=strstr(p12,"D");
    if(tp == 'D' && op2>0){

```

```

        goto dato3;
    }
    if(!p == 'D' && op2==0){
        mensaje(220*Escy,46,0,23, " E R R O R ! NO existe dato para SENSIBILIDAD");
        goto dato2;
    }

    R10=Ofa-Opand+90;
    if(R10<0)R10=0;
    R11=Ofa+Opand+90;
    mensaje(205*Escy,56,20,23, " ");
    dato3;
    p13 = linea3;
    ayuda_linea(74,53,1);
    outtextxy (prin, 269*Escy, "TOLERANCIA [ porcentaje (%) ]");
    op3 = gread (a+20*espx1, 267*Escy, linea3,ind3, 9, op3,53,2);
    p13++;
    ind3=1;
    p=strslr(p13,"\\1B");
    if(!p == "\\1B")return 0;
    Dini = atof (p13);
    if(Dini>100){
        mensaje(240*Escy,46,0,23, " E R R O R ! Ingresar: TOLERANCIA < 100 ");
        goto dato3;
    }
    Dini= (float)Dini/100;
    p=strslr(p13,"C");
    if(!p == 'C'){
        p11--;
        goto dato2;
    }
    p=strslr(p13,"D");
    if(!p == 'D' && op3==0){
        mensaje(240*Escy,46,0,23, " E R R O R ! NO existe dato para TOLERANCIA");
        goto dato3;
    }

    Opand=Opand+90/Ofa;
    p_vco=1;
    return 1;
}

```

/!ESTRUCTURAR\_DEFASADOR:Entrada de datos para estructurar el DEFASADOR  
 # Parámetros: Ninguno

# Retorno : Indicador (Datos aceptados? si o no)

#####

char estructurar\_defasador(void)

{

float n,nd,wr,wp,fs1,AK1,AK2,AK3,AK4,DK,DK1,DKP;

int ind1=0,ind2=0,ind3=0,ind4=0,locx,y,n,espx1;

char linea1[30],linea2[30],linea3[30],linea4[30];

char #p11,#p12,#p13,#p14,#p;

int op1=0,op2=0,op3=0,op4=0;

n=(MaxX-24\*Escx)/2; espx1=textwidth("N");

locx=n-30\*espx1;borrar(1);

setviewport(0,0,MaxX,MaxY,1);

setcolor(0);

setlinestyle(SOLID\_LINE,1,4);

settextstyle(0,0,0);

bar (n-36\*espx1,290\*Escy,n+36\*espx1,305\*Escy);

line(n-36\*espx1,292\*Escy,n+36\*espx1,292\*Escy);

outtextxy(a-35\*espx1,296\*Escy,"[Esc] SALIR [F9] Dato ANTERIOR [F10] o [ENTER] Dato POSTERIOR ");

```

setfillstyle (SOLID_FILL, getcolor());
setcolor (getcolor());
leaa(200*Escy,1, 'Estructurando el Defasador de 90 grados*');
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
rectangle(a-36*espx1, 195*Escy, a+36*espx1, MaxY-41*Escy);
dato1: do{ p11=linea1;
ayuda_linea(74,32,2);
outtextxy (locx, 225*Escy, "Frecuencia INFERIOR ( f1 ) [ H z ]*");
op1 = qread (a+20*espx1, 225*Escy, linea1,ind1, 9, op1,7,2);
p11++;
ind1=1;
p=strstr(p11, "\n");
if(!p == "\n")return 0;
p=strstr(p11, "C");
if(!p == "C"){
p11--;
goto dato1;
}
p=strstr(p11, "D");
if(!p == "D"&& op1==0 )
mensaje(180*Escy,47,0,23, " E R R O R ! NO existe dato para Frecuencia INFERIOR*");
}while (op1 == 0);
Dfp = atof (p11);
dato2:
p12 = linea2;
ayuda_linea(74,33,2);
outtextxy (locx, 240*Escy, "Frecuencia SUPERIOR ( f2 ) [ H z ]*");
op2 = qread (a+20*espx1, 240*Escy, linea2,ind2, 9, op2,7,2);
p12++;
ind2=1;
p=strstr(p12, "\n");
if(!p == "\n")return 0;
Dfr = atof (p12);
p=strstr(p12, "C");
if(!p == "C"){
p12--;
goto dato1;
}
p=strstr(p12, "D");
if(!p == "D"&& op2==0 ){
goto dato3;
}
if(!p == "D"&& op2==0 ){
mensaje(205*Escy,47,0,23, " E R R O R ! NO existe dato para Frecuencia SUPERIOR*");
goto dato2;
}
if(Dfp>=Dfr){
mensaje(190*Escy,55,0,23, " E R R O R ! INGRESAR: Frecuencia INFERIOR < Frecuencia SUPERIOR*");
goto dato2;
}
dato3:
p13 = linea3;
ayuda_linea(74,34,2);
outtextxy (locx, 255*Escy, "ERROR de FASE ( EPS ) [ grados ]*");
op3 = qread (a+20*espx1, 255*Escy, linea3,ind3, 9, op3,7,2);
p13++;
ind3=1;
p=strstr(p13, "\n");
if(!p == "\n")return 0;
EPS= atof (p13);
p=strstr(p13, "C");

```

```

if(*p == 'C'){
    p12--;
    goto dato2;
}
p=ptrstr(p13,"D");
if(*p == 'D' && op3>0){
    goto dato4;
}
if(*p == 'D' && op3==0){
    mensaje(220*Escy,09,0,23, " E R R O R ! NO existe dato para el FPPOR de CASE ");
    goto dato3;
}

if(EPS>45){
    mensaje(190*Escy,40,0,23, " E R R O R ! INGRESAR: e < 45 grados ");
    goto dato3;
}

EPS=F1*EPS/100;
dato4: do{
    fs1=2.1*DFr;
    wp=PI*DFr/fs1;
    wr=PI*DFr/fs1;
    DK=tan(wp)/tan(wr);
    DK1=(1-tan(EPS/2))/(1+tan(EPS/2));DK1=pow(DK1,2);
    DKP=sqrt(1-pow(DK,2));
    AK1=AK(DK);AK2=AK(DK1);
    AK3=AK(DKP);AK4=AK(sqrt(1-pow(DK1,2)));
    n=(AK2*AK3)/(AK1*AK4);if(n<0)n=n*-1;
    p14 = linea4;
    y = 264*Escy;
    borrar(A);
    gprintf (&locx, &y,2, "ORDEN del FILTRO (se recomienda) > %f",n);
    ayuda_linea(74,35,2);
    op4 = gread (&+20*espa, 270*Escy, linea4,ind4, 9, op4,7,2);
    ind4=1;
    p14++;
    p=ptrstr(p14,"X1R");
    if(*p == 'X1R')return 0;
    p=ptrstr(p14,"C");
    if(*p == 'C'){
        p13--;
        goto dato3;
    }
    p=ptrstr(p14,"D");
    if(*p == 'D' && op4==0)
        mensaje(210*Escy,46,0,23, " E R R O R ! NO existe dato para ORDEN del DEFASADOR ");
}while( op4 == 0);
nd = atof (p14);
Dn=nd;
if(Dn>5){
    mensaje(210*Escy,44,0,23, " E R R O R ! ORDEN del FILTRO debe ser menor que 5 ");
    goto dato4;
}

p_def=1;
return j;
}

```

/\*ENTRADA\_BANDA:Control de entrada de datos para estructurar el filtro pasabanda

\* Parámetros: caso .- Indicador de que filtro (existen 2)

\* Retorno : Indicador (Datos acoplados? si o no)

#####

char entrada\_banda(inl dc)

{

```
float fp,fr,frf,fpr,awr,awp,n,nd,fsi,k0,k1,k2,q0,q,DD,fo,B;
```

```
int x,y,locxx;
int ind1=0,ind2=0,ind3=0,ind4=0,ind5=0,ind6=0;
char linea1[30],linea2[30],linea3[30],linea4[30],linea5[30],linea6[30];
char *p11,*p12,*p13,*p14,*p15,*p16,*p;
int op1=0,op2=0,op3=0,op4=0,op5=0,op6=0,l,*espx1;
    m=(MaxX-24*Escx)/2;
    espx1=textwidth("M");
    locxx=m-36*espx1;
dato1: do( p11=linea1;
    ayuda_linea(74,42,2);
    outtextxy (locxx, 230*Escy, "Frecuencia CENTRAL (fo)[Hz]");
    op1 = grread (locxx+28*espx1, 230*Escy, linea1,ind1, 9, op1,7,2);
    p11++;
    ind1=1;
    p=strstr(p11,"\x1B");
    if(*p == '\x1B')return 0;
    p=strstr(p11,"C");
    if(*p == 'C'){
        p11--;
        goto dato1;
    }
    fo = atof (p11);
    p=strstr(p11,"D");
    if((*p == 'D' && op1==0) || fo==0 )
        mensaje(200*Escy,54,0,23, " E R R O R ! NO existe dato para el
Frecuencia CENTRAL");
    }while (op1 == 0 || fo==0);
dato2:
p12 = linea2;
ayuda_linea(74,43,2);
outtextxy (locxx, 250*Escy, "Frec. de CORTE {fp}[Hz]");
op2 = grread (locxx+28*espx1, 250*Escy, linea2,ind2, 9, op2,7,2);
p12++;
ind2=1;
p=strstr(p12,"\x1B");
if(*p == '\x1B')return 0;
fpr = atof (p12);
p=strstr(p12,"C");
if(*p == 'C'){
    p12--;
    goto dato1;
}
p=strstr(p12,"D");
if(*p == 'D' && op2>0){
    goto dato3;
}
if((*p == 'D' && op2==0) || fpr==0){
    mensaje(220*Escy,52,0,23, " E R R O R ! NO existe dato para Frecuencia de CORTE");
    goto dato2;
}
if(fpr==fo){
    mensaje(195*Escy,51,0,23, " E R R O R ! INGRESAR: Frec. de CORTE < Frecuencia CENTRAL");
    goto dato2;
}
dato3:
p13 = linea3;
ayuda_linea(74,44,2);
outtextxy (locxx, 270*Escy, "Frec. de SUPRESION {fr}[Hz]");
op3 = grread (locxx+28*espx1, 270*Escy, linea3,ind3, 9, op3,7,2);
```

```

p13++;
ind3=1;
p=strstr(p13, "\x1B");
if(!p == "\x1B")return 0;
frr= atof (p13);
p=strstr(p13, "C");
if(!p == 'C'){
    p12--;
    goto dato2;
}
p=strstr(p13, "D");

if(!p == 'D' && op3>0 ){
    goto dato4;
}
if(!p == 'D' && op3==0) ;{frr==0 ){
    mensaje[225*Escy,51,0,23, " E R R O R !   NO existe dato para (frec. de SUPRESION)";
    goto dato3;
}

if(frr>fpr){
    mensaje[200*Escy,51,0,23, " E R R O R !   INGRESAR: Frec. de CORTE < (frec. de SUPRESION)";
    goto dato3;
}

dato4: p14 = linea4;
ayuda_linea(74,45,2);
outlexiy (locxx+40*espx1, 230*Escy, "ATENUACION en fp (Afp)[dB]");
op4 = gread (locxx+67*espx1, 230*Escy, linea4, ind4, 4, op4,7,2);
ind4=1;
p14++;
p=strstr(p14, "\x1B");
if(!p == "\x1B")return 0;
awp = atof (p14);
p=strstr(p14, "C");
if(!p == 'C'){
    p13--;
    goto dato3;
}
p=strstr(p14, "D");
if(!p == 'D' && op4>0 ){
    goto dato5;
}
if(!p == 'D' && op4==0) ;{awp==0){
    mensaje[195*Escy,60,0,23, " E R R O R !   NO existe dato para ATENUACION en FRECUENCIA
de CORTE");
    goto dato4;
}

dato5: p15 = linea5;
ayuda_linea(74,46,2);
outlexiy (locxx+40*espx1, 250*Escy, "ATENUACION en fr (Afr)[dB]");
op5 = gread (locxx+67*espx1, 250*Escy, linea5, ind5, 4, op5,7,2);
ind5=1;
p15++;
p=strstr(p15, "\x1B");
if(!p == "\x1B")return 0;
awr = atof (p15);
p=strstr(p15, "C");
if(!p == 'C'){
    p14--;
    goto dato4;
}
p=strstr(p15, "D");

```

```

if(*p == 'D' && op5 > 0 ){
    goto dato6;
}
if(*p == 'D' && op5 == 0 ){
    mensaje(215*Escy, 59, 0, 23, "ERROR! NO existe dato para ATENUACION en FRECUENCIA de
SUPRESION");
    goto dato5;
}
if(amp > awr){
    mensaje(175*Escy, 59, 0, 23, "ERROR! INGRESAR: ATENUACION en fp < ATENUACION en fr");
    goto dato5;
}
dato6: do{
    fr = frr;
    fp = fpr;
    fr = fo - fr;
    R = (fo - fp) * 2;
    fp = R / 2;
    fs1 = 2 * fo;
    k0 = tan(PI * fp / fs1) / tan(PI * fr / fs1);
    k1 = sqrt(1 - pow(k0, 2));
    k2 = sqrt(k1);
    q0 = .5 * (1 - k2) / (1 + k2); q = q0 + 2 * pow(q0, 5) + 15 * pow(q0, 7) + 150 * pow(q0, 13);
    DD = (pow(10, 0.1 * awr) - 1) / (pow(10, 0.1 * amp) - 1);
    n = log10(16 * DD) / log10(1 / q);
    pl6 = linea6;
    x = locxx + 40 * espx1;
    y = 263 * Escy;
    borrar(5);
    gprintf (&x, &y, 2, "ORDEN del FILTRO > % .1f", n);
    ayuda_linea(74, 47, 2);
    op6 = gread (locix + 67 * espx1, 270 * Escy, linea6, ind6, 4, op6, 7, 2);
    ind6 = 1;
    pl6++;
    p = strstr(pl6, "\\x1R");
    if(*p == '\\x1R') return 0;
    p = strstr(pl6, "C");
    if(*p == 'C'){
        pl6--;
        goto dato5;
    }
    nd = atoi (pl6);
    n = nd;
    p = strstr(pl6, "D");
    if((*p == 'D' && op6 == 0) || n == 0)
        mensaje(210*Escy, 45, 0, 23, "ERROR! NO existe dato para ORDEN DEL FILTRO ");
    }while( op6 == 0 || n == 0);
    if(n > 11){
        mensaje(210*Escy, 46, 0, 23, "ERROR! ORDEN del FILTRO debe ser menor que 10 ");
        goto dato6;
    }

    switch(dc){
        case 1:
            Afo=fo;AW=R;frbanA=fr;AfpbanA=amp;AfrbanA=awr;BAN=n;break;
        case 2:
            Bfo=fo;BW=R;frbanB=fr;AfpbanB=amp;AfrbanB=awr;BBn=n;break;
    }
    return 1;
}

```

/4ENTRADA\_FILTRO:Control de entrada de datos para estructurar el filtro pasabajos



```

# Parámetros : caso .- Indicador de que filtro (existen 4)
# Retorno : Indicador (Datos acapilados? si o no)
#####/
char entrada_filtro(int dc)
{
float fp, fr, awr, awp, n, nd, wr, wp, (s1, k0, k1, k2, q0, q, q0);
int x, y, esp1, s, locx, f, ind1=0, ind2=0, ind3=0, ind4=0, ind5=0;
char linea1[30], linea2[30], linea3[30], linea4[30], linea5[30];
char *p1, *p2, *p3, *p4, *p5, *p;
int op1=0, op2=0, op3=0, op4=0, op5=0, l;
a=(MaxX-24*Escx)/2;
esp1=lexlwidth("M");
locx=a-36*esp1;
dato1: do( p1=linea1;
ayuda_linea(74,2,2);
outlexlx (locx, 240*Escy, "FREC. de CORTE (fp) [Hz]");

op1 = gread (locx+28*esp1, 240*Escy, linea1, ind1, 9, op1, 7, 2);
p1++;
ind1=1;
p=strstr(p1, "\n");
if(*p == "\n")return 0;
p=strstr(p1, "C");
if(*p == 'C'){
p1--;
goto dato1;
}

fp = atof (p1);
p=strstr(p1, "D");
if((*p == 'D' && op1==0) || fp==0)
mensaje(205*Escy, 51, 0, 23, " E R R O R ! NO existe dato para FRECUENCIA de CORTE ");
}while (op1 == 0 || fp==0);

dato2:
p2 = linea2;
ayuda_linea(74,3,2);
outlexlx (locx, 258*Escy, "FREC. de SUPRESION (fr)[Hz]");
op2 = gread (locx+28*esp1, 258*Escy, linea2, ind2, 9, op2, 7, 2);
p2++;
ind2=1;
p=strstr(p2, "\n");
if(*p == "\n")return 0;
fr = atof (p2);
p=strstr(p2, "C");
if(*p == 'C'){
p1--;
goto dato1;
}

p=strstr(p2, "D");

if(*p == 'D' && op2>0 ){
goto dato3;
}
if(*p == 'D' && op2==0 ){
mensaje(225*Escy, 55, 0, 23, " E R R O R ! NO existe dato para FRECUENCIA de
SUPRESION");
goto dato2;
}

if(fp>=fr){
mensaje(210*Escy, 51, 0, 23, " E R R O R ! INGRESAR: Frec. de CORTE < Frec. de SUPRESION");
goto dato2;
}

```

```

    }

dato3:  p13 = linea3;
ayuda_linea(74,4,2);
outltxly (locx+40*espx1, 240*Escy, "ATENUACION en fp (Afp)[dB]");
op3 = gread (locx+67*espx1, 240*Escy, linea3,ind3, 4, op3,7,2);
ind3=1;
p13++;
p=slrstr(p13,"\\x1B");
if(tp == "\\x1B")return 0;
awp = arof (p13);
p=slrstr(p13,"C");
if(tp == 'C'){
    p12--;
    goto dato2;
}
p=slrstr(p13,"D");
if(tp == 'D' && op3>0 ){
    goto dato4;
}
if((tp == 'D' && op3==0) || awp==0){
    mensaje(212*Escy,60,0,23, " E R R O R !   NO existe dato para ATENUACION en FRECUENCIA
de CORIE");
    goto dato3;
}

dato4:  p14 = linea4;
ayuda_linea(74,5,2);
outltxly (locx+40*espx1, 250*Escy, "ATENUACION en fr (Afr)[dB]");
op4 = gread (locx+67*espx1, 250*Escy, linea4,ind4, 4, op4,7,2);
ind4=1;
p14++;
p=slrstr(p14,"\\x1B");
if(tp == "\\x1B")return 0;
awr = arof (p14);
p=slrstr(p14,"C");
if(tp == 'C'){
    p13--;
    goto dato3;
}
p=slrstr(p14,"D");
if(tp == 'D' && op4>0 ){
    goto dato5;
}
if(tp == 'D' && op4==0 ){
    mensaje(230*Escy,58,0,23, " ERROR !   NO existe dato para ATENUACION en FRECUENCIA de
SUPRESION");
    goto dato4;
}

if(awp>awr){
    mensaje(212*Escy,50,0,23, "ERROR!   INGRESAR: ATENUACION en fp < ATENUACION en fr");
    goto dato4;
}

dato5:  do{
wp=2*PI*fp;wr=2*PI*fr;
    if(dc==1 || dc == 2) fsl=3.5*fr;
    if(dc==3 || dc == 4) fsl=5*fr;
if(dc==1 || dc ==2) {
    if(nbuclt !=10 )k0=lan(wp/2/fsl)/lan(wr/2/fsl);
    else          k0=wp/wr;

```

```

k1=sqrt(1-pow(k0,2));
k2=sqrt(k1);
q0=.5*(1-k2)/(1+k2);q=q0+2*pow(q0,5)+15*pow(q0,9)+150*pow(q0,13);
D0=(pow(10,0.17*wr)-1)/(pow(10,.17*wp)-1);
n=log10(16*D0)/log10(1/q);
}
if(dc==3 || dc == 4){
if(nbucle !=1){
wp=2*atan(wp/2/fz1);
wr=2*atan(wr/2/fz1);
}
else {
wr=wr/wp;wp=wp/wp;
}
n=(pow(10,.17*wr)-1)/(pow(10,.17*wp)-1);
n=.5*log10(n)/log10(wr/wp);
}
p15 = linea5;
x = x-24*Esp1;
y = 269*Escy;
borrar(4);
pprintf (&x, &y,2, "ORDEN DEL FILTRO (N > 2 .2f)",n);
ayuda_linea(74,6,1);
op5 = grread (&18*Esp1, 275*Escy, linea5,ind5, 4, up5,6,2);
ind5=1;
p15++;
p=strstr(p15,"\\x18");
if(*p == '\\x18')return 0;
p=strstr(p15,"C");
if(*p == 'C'){
p14--;
goto dato4;
}
nd = atof (p15);
n=nd;
p=strstr(p15,"D");
if((*p == 'D' && op5==0);;n==0)
mensaje(245*Escy,46,0,23, " E R R O R ! No existe dato para ORDEN DEL FILTRO ");
}while( op5 == 0 || n==0);
if(n>10){
mensaje(235*Escy,46,0,23, " E R R O R ! ORDEN del FILTRO debe ser menor que 10 ");
goto dato5;
}
switch(dc){
case 1:
fpeA=fp;freA=fr;hpeA=awp;hfreA=awr;hu=n; break;
case 2:
fpeB=fp;freB=fr;hpeB=awp;hfreB=awr;hu=n; break;
case 3:
fpeB=fp;freB=fr;hpeB=awp;hfreB=awr;hu=n; break;
case 4:
fpeB=fp;freB=fr;hpeB=awp;hfreB=awr;hu=n; break;
}
return 1;
}

```

/\*ENTRADA\_FILTRO:Entrada de datos para estructurar la entrada a los bucles

\* Parámetros: Ninguno

\* Retorno : Indicador (Datos acaptados? si o no)

#####

char entrada(void)

{

```

int prinx,exf,res,f,ind1=0,ind2=0,ind3=0,ind4=0;
char linea1[30],linea2[30],linea3[30],linea4[30];
int op1=0,op2=0,op3=0,op4=0,l,yy1,nu,*,espx1;
char *p11,*p12,*p13,*p14,*p;
    m=(MaxX-24*Esci)/2;
    espx1=lexwidth("M");
    prinx=m-30*espx1;
    exf=m+8*espx1;
    ch=1;
    for (l=0; l<=20; l++)
        in[l] = 0;

    borrar(l);
    selviewperl(0,0,MaxX,MaxY,l);
    selcolor(0);
    setlinestyle(SOLID_LINE,1,4);
    settextstyle(0,0,0);
    bar (m-36*espx1,290*Escy,m+36*espx1,305*Escy);
    line(m-36*espx1,292*Escy,m+36*espx1,292*Escy);
    outtextxy(m-35*espx1,296*Escy,"[Esc] SALIR    [F9] Dato ANTERIOR    [F10] o [ENTER] Dato POSTERIOR ");
    selfillstyle (SOLID_FILL, getmaxcolor());
    selcolor (getmaxcolor());
    leaa(200*Escy,1, "Formando la ONDA de ENTRADA al PUCLE");
    settxtjustify( LEFT_TEXT, TOP_TEXT );
    settxtstyle (DEFAULT_FONT, HORIZ_DIR, 1);
    rectangle(m-36*espx1, 195*Escy,m+36*espx1, MaxY-41*Escy);

dato1: do(    p11=linea1;
            ayuda_linea(74,10,1);
            outtextxy (prinx, 224*Escy, "BIT's A SIMULAR");
            op1 = grread (exf, 224*Escy, linea1,ind1, 20, op1,10,1);
            p11++;
            ind1=1;
            p=strstr(p11,"\x1B");
            if(*p == '\x1B')return 0;

            p=strstr(p11,"C");
            if(*p == 'C'){
                p11--;
                goto dato1;
            }

            p=strstr(p11,"D");
            if(*p == 'D' && op1==0 )
                mensaje(190*Escy,51,0,23, " E R R O R !    NO existe dato VALIDO para BIT's a SIMULAR");

        )while (op1 == 0);
        for (l=1; l<=op1; l++)
            in[l-1] = linea1[l] - 48;
        N2 = op1;

dato2:
p12 = linea2;
ayuda_linea(74,11,1);
outtextxy (prinx, 241*Escy, "RITMO de TRASMISION    (R)    [bps]");
op2 = grread (exf, 241*Escy, linea2,ind2, 10, op2,11,2);
p12++;
ind2=1;
p=strstr(p12,"\x1B");
if(*p == '\x1B')return 0;

ncb = atof (p12);
p=strstr(p12,"C");
if(*p == 'C'){

```

```

        p11 ;
        for (i=0; i<20; i++)
            in[i] = 0;
        goto dato1;
    }
    p=strcmp(p12,"D");
    if(*p == 'D' && op2==0){
        goto dato1;
    }
    if((*p == 'D' && op2==0) || !ch--){
        mensaje(200+Escy,55,0,23, " E R R O R !   NO existe dato VALIDO para RITMO de
TRANSMISION");
        goto dato2;
    }

    dato3:  p13 = linea3;
            ayuda_linea(74,12,1);
            outlexy (prin3, 258+Escy, "FRECUENCIA DE PORTADORA (fc) [Hz]");
            op3 = grread (exf, 258+Escy, linea3, ind3, 10, op3, 12, 2);
            ind3=1;
            p13++;
            p=strcmp(p13, "\x1B");
            if(*p == '\x1B') return 0;
            ff = atof (p13);
            p=strcmp(p13, "C");
            if(*p == 'C'){
                p12--;
                goto dato2;
            }
            p=strcmp(p13, "D");
            if(*p == 'D' && op3>0){
                goto dato4;
            }
            if((*p == 'D' && op3==0) || !f==0){
                mensaje(218+Escy,57,0,23, " E R R O R !   NO existe dato VALIDO para FRECUENCIA de
PORTADORA");
                goto dato3;
            }

    dato4:  do{
            p14 = linea4;
            ayuda_linea(74,13,1);
            outlexy (prin4, 275+Escy, "ANGULO DE DEFASAMIENTO (α)[grados]");
            op4 = grread (exf, 275+Escy, linea4, ind4, 10, op4, 13, 2);
            ind4=1;
            p14++;
            p=strcmp(p14, "\x1B");
            if(*p == '\x1B') return 0;
            p=strcmp(p14, "C");
            if(*p == 'C'){
                p13--;
                goto dato3;
            }
            p=strcmp(p14, "D");
            if(*p == 'D' && op4==0){
                mensaje(233+Escy,56,0,23, " E R R O R !   NO existe dato VALIDO para ANGULO de
DEFASAMIENTO");
            }
            }while( op4 == 0);
            DO1 = atof (p14);
            if(DO1>360){
                mensaje(235+Escy,56,0,23, " E R R O R !   ANGULO de DEFASAMIENTO debe ser menor que 360 grados");
                goto dato4;
            }

```

```

    }
    ncb=ff/ncb;
    DO1=001*PI/180;
    WO=2*PI*fff;
    nc=ncb*H2;
    if(nbuclé == 7)nc=ncb*H2/2;
        p_entrada=1;
    return 1;
}

/*BEEP: Control de sonido
* Parámetros: f.- ciclos de sonido
*             d.- tiempo de sonido
* Retorno : Ninguno
*/
void beep (int f, int d)
{
    sound (f);
    delay (d);
    nosound();
}

/*MOV: Movimiento del cursor en las ventanas de texto
* Parámetros: x.- coordenada X de inicio del carácter
*             yy.- coordenada Y de inicio del carácter
*             x2.- coordenada X de fin del carácter
*             lor.- Indicador de desplazamiento hacia atrás
* Retorno : Ninguno
*/
void mov (int x1, int yy, int x2, int lor)
{
    size_t ss;
    int fpp;

    ss = imagesize (x1, yy, x2, yy+7);
    fpp = _graphgetmode (ss);
    getimage (x1, yy, x2, yy+7, fpp);
    if (lor == 0)
        x1 = x1-8;
    else
    {
        x1 = x1+8;
        x2 = x1;
    }
    putimage (x1, yy, fpp, 0);
    bar (x2-8, yy, x2-1, yy+7);
    _graphfreemode (fpp, ss);
}

/*GREAD: Simula ventana de texto en modo gráfico
* Parámetros: x.- coordenada X de inicio del recuadro
*             y.- coordenada Y de inicio del recuadro
*             str.- mensaje
*             indicador.- Tipo de entrada que se acepta (números, letras, teclas de funciones, etc)
* Retorno : Indicador [D: datos aceptados? (S/N) o el código de la tecla de función]
*/
int gread (int x, int y, char str[30], int indicador, int l, int esc, int help, int numero)
{
    struct fillsettings_t lastfill;
    char s[30], *ss[1];
    int ch, w, espx, i, xp, ins, opc, done, nuevo, tip, cp, ipp, ind=0;
    size_t size;
    espx=textwidth("M");

```



```

    }
    else
    {
        cp = (xp-x) / 8 + 1;
        getimage (xp, y, xp+7, y+7, ip);
        sellinestyle(0,0,THICK_WIDTH);
        line (xp+2, y, xp+2, y+7);
        if (ins == 0)
            line (xp+4, y, xp+4, y+7);
        sellinestyle(0,0,THIN_WIDTH);
        do{
            }while(!kbhit());
            if(kbhit()){
                ch=tecla();
                if(ch==Fnl){
                    ayuda(a-34*espl,70*Escy,a+34*espl,270*Escy,help);
                    ch=tecla();
                    setfillstyle (SOLID_FILL, 0);
                    setcolor (getmaxcolor());
                }
            }
        }
        pulimage (xp, y, ip, 0);
    }
}
if(numero==1){
    if (!( (ch == 48) || (ch == 49) || (ch == 338) || (ch == 339) ||
        (ch == 327) || (ch == 328) || (ch == 329) || (ch == 336) ||
        (ch == 331) || (ch == 333) || (ch == 335) || (ch == 27) ||
        (ch == 13) || (ch == 32) || (ch == 9) || (ch == 8) ||
        (ch == 323) || (ch == 324) || (ch == 337))) ch = 0;
}
if(numero==2){
    if (!( (ch >= 48) && (ch <= 57) || (ch == 46) || (ch == 338) || (ch == 339) ||
        (ch == 327) || (ch == 328) || (ch == 329) || (ch == 336) ||
        (ch == 331) || (ch == 333) || (ch == 335) || (ch == 27) ||
        (ch == 13) || (ch == 32) || (ch == 9) || (ch == 8) ||
        (ch == 323) || (ch == 324) || (ch == 337))) ch = 0;
}
if(numero==3){
    if(ch == 78 || ch == 110 || ch == 83 || ch == 115){
        noc=1;
        str[0] = ch;
        return noc;
    }
    if (!( (ch == 78) || (ch == 110) || (ch == 83) || (ch == 115) ||
        (ch == 327) || (ch == 328) || (ch == 329) || (ch == 336) ||
        (ch == 331) || (ch == 333) || (ch == 335) || (ch == 27) ||
        (ch == 13) || (ch == 32) || (ch == 9) || (ch == 8) ||
        (ch == 338) || (ch == 339) || (ch == 337))) ch = 0;
}
if(numero==4){
    if (!( (ch >= 48) && (ch <= 57) || (ch == 338) || (ch == 339) ||
        (ch == 327) || (ch == 328) || (ch == 329) || (ch == 336) ||
        (ch == 331) || (ch == 333) || (ch == 335) || (ch == 27) ||
        (ch == 13) || (ch == 32) || (ch == 9) || (ch == 8) ||
        (ch == 323) || (ch == 324) || (ch == 337))) ch = 0;
}
if(numero==5){
    if (!( (ch == 66) || (ch == 69) || (ch == 98) || (ch == 101) ||
        (ch == 338) || (ch == 339) || (ch == 327) || (ch == 328) ||
        (ch == 329) || (ch == 336) || (ch == 331) || (ch == 333) ||
        (ch == 335) || (ch == 27) || (ch == 13) || (ch == 32) ||

```



```

        (ch == 9) || (ch == 8) || (ch == 337)))
        ch = 0;
    }
if(numero==6){
    if(ch >= 48 && ch <= 57){
        noc=1;
        str[0] = ch;
        return noc;
    }
    if (!((ch >= 48) && (ch <= 57) || (ch == 46) || (ch == 338) || (ch == 339) ||
        (ch == 327) || (ch == 328) || (ch == 329) || (ch == 336) ||
        [ch == 331] || (ch == 333) || (ch == 335) || (ch == 27) ||
        (ch == 13) || (ch == 32) || (ch == 9) || (ch == 8) ||
        (ch == 337))) ch = 0;
    }
    if (ch > 326 || ch == 32 || ch == 9)
    {
        if (nuevo == 0)
        {
            bar (x-1, y-1, x+18+4, y+8);
            outtextxy (x, y, s);
            nuevo = 1;
        }

        do{
        }while(!kbhit());
        if(kbhit()){
            ch = getch();
            if(ch==F0){
                ayuda(a-34*esp1,70*Escy,a+34*esp1,270*Escy,help);
                ch=getch();
                setfillstyle (SOLID_FILL, 0);
                setcolor (getmaxcolor());
            }
        }

        switch (ch)
        {
            case 338:
                if (ins == 0)
                    ins = 1;
                else
                    ins = 0;
                break;
            case 339:
                if (xp+8 <= x+noc+8)
                {
                    bar (xp, y, xp+7, y+7);
                    mov (xp+8, y, x+noc+8+7, 0);
                    if (noc > 0)
                        noc = noc-1;
                }
                break;
            case 327:
                xp = x;
                break;
            case 335:
            case 9:
                xp = x+noc+8;
                break;
            case 331:

```

```

        case 336:
        case 337:
            if (xp > x)
                xp = xp-8;
            break;
        case 333:
        case 328:
        case 329:
        case 32:
            if (xp-x < 148 && xp<x+noc*8)
                xp = xp+8;
            break;
    }
else
{
    if ((nuevo == 0) && (ch != 13) && indicador == 0)
    {
        noc = 0;
        cp = 1;
        s[0] = (char)0;
        har (x-1, y-1, x+148+6, y+8);
        outtextxy (x, y, s);
        nuevo = 1;
    }
    switch (ch)
    {
        case 13:
            if (nuevo == 0)
                har (x-1, y-1, x+148+6, y+8);
            outtextxy (x, y, s);
            done = 0;
            esc=1;
            break;
        case 27:
            esc = 0;
            done = 0;
            s[cp] =(char) ch;
            for (i=0;i<noc+1;i++)
                str[i] = s[i];
            return noc;
        case 8:
            if ((xp < x+noc*8+7) && (xp >= x) && (xp > x))
            {
                mov (xp, y, x+noc*8+7, 0);
                noc = noc - 1;
                xp = xp - 8;
            }
            break;
        default:
            if (ins == 0)
            {
                if (noc < 1)
                {
                    if (xp <= x+noc*8-1)
                        mov (xp, y, x+noc*8-1, 1);
                    if (ch>314 && ch<327)
                    {
                        str[0] = ch;
                        noc=1;
                        return noc;
                    }
                    ss[0] = (char) ch;
                }
            }
    }
}

```



```

/!ELFILTRO: Estructura el filtro pasabajos ELIPTICO
! Parámetros: resp.- Indica si se debe presentar o no la respuesta de frecuencia
!             dc.- Indica que filtro se estructura (existen 2 filtros)
! Retorno: Ninguno
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!/
void elfiltro(int resp,int dc)
{
int i,j,r,b,il,l,decision=0,deser,*;
float fr,fp,fsl,k1,k2,k0,q0,q,DD,bh,on,n,lan,s1=0,s2=0,s0,w;
float nd,ho,Adato,Bdato,Cdato,wp,wr,awp,oh,vi,bi,ol,o2,o3,u;
n=(MaxX-24*Escx)/2;
switch(dc){
case 1:fp = fpeA ; fr = freA ; awp = AfpeA ; n=An;break;
case 2:fp = fpeB ; fr = freB ; awp = AfpeB ; n=Bn;
}
fsl=3.5*fr;wp=2*P1*fr;wr=2*P1*fr;
if(nbucl != 10)
k0=tan(wp/2/fsl)/tan(wr/2/fsl);
else k0=wp/wr;
k1=sqrt(1-pow(k0,2));k2=sqrt(k1);
q0=.5*(1-k2)/(1+k2);q=q0+2*pow(q0,5)+15*pow(q0,9)+150*pow(q0,13);
lan=.5/n*log((pow(10,.05*awp)+1)/(pow(10,.05*awp)-1));
for(i=0;i<5;i++){
j=i*(i+1);s1=s1*pow(-1,i)*pow(q,j)*sinh((2*i+1)*lan);
}
for(j=1;j<7;j++){
l=pow(j,2);s2=s2+pow(-1,j)*pow(q,l)*cosh(2*j*lan);
}
s0=2*pow(q,.25)*s1/(1+2*s2);
s0= s0>0? s0:-1*s0;
w=sqrt((1+k0*pow(s0,2))*(1+pow(s0,2)/k0));
nn=n;
while (nn>0)nn=nn-2;
r=nn==0?n/2:(n-1)/2;
for(il=1;il<r+1;il++){
u=nn==0?il-.5:il;
ol=0;o2=0;
for(i=0;i<15;i++){
b=i*(i+1);o3=(2*i+1)*P1/n*u;ol=ol+pow(-1,i)*pow(q,b)*sin(o3);
}
for(i=1;i<17;i++){
b=pow(i,2);o2=o2+pow(-1,i)*pow(q,b)*cos(2*i*P1*u/n);
}
oh=2*pow(q,.25)*ol/(1+2*o2);
vi=sqrt(((1-k0*pow(oh,2))*(1-pow(oh,2)/k0));Adato=1/pow(oh,2);
bi=1+pow(s0*oh,2);Cdato=(pow(s0*vi,2)+pow(oh*w,2))/pow(bi,2);
Bdato=2*s0*vi/bi;
if(dc==1){
AAD[il]=Adato;AC[il]=Cdato;ABR[il]=Bdato;
}
else{
BAD[il]=Adato;BC[il]=Cdato;RBR[il]=Bdato;
}
}
bb=1;
if(dc==1){
for(i=1;i<r+1;i++){
bb=bb*AC[i]/AAD[i];
if(resp !=0 && nbucl !=10){
fsea=f/fsl*W/nc;
if(fsea>100)
fsea=100;
}
}
}
}

```

```

    }
else {
    for(i=1;i/r+1;i++)
        hh-hb*pr[i]/AA0[i];
    if(resp !=0 && nbucle !=10){
        fseb=f/fsl*H/nr;
        if(fseb>100)
            fseb=100;
    }
}
hn=nn==0?hb*prn(10,-,0.5*wp):hb*c0;
if (n==1)ho=s0;
if(dc==1){
    A0=s0;Ann=nn;Ahn=hn;
}
else {
    B0=s0;Bnn=nn;Bhn=hn;
}
if(resp ==0){
    decision=mensaje(277*Escy,50,3,7, " FRESFIAR RESPUESTA DE FRECUENCIA DEL FILTRO (S/N)");
    if(decision=='s' || decision=='S'){
        if(dc==1){
            fsea=fsl;
            ayuda(n-31*lexwidh("N"),70*Escy,n+31*lexwidh("N"),270*Escy,100);
        }
        else {
            fseb=fsl;
            ayuda(n-31*lexwidh("N"),70*Escy,n+31*lexwidh("N"),270*Escy,101);
        }
    }
}
}
if (nbucle==10){
    if(n>1)
        AC[0]=(f|n-1)/r;
    ho=pow(ho,AC[0]);
    for(i=1;i<r+1;i++){
        if(AA0[i]<0)AA0[i]=-AA0[i];
        BC[i]=1/sqrt(AA0[i]);
        BB[i]=(AA0[i]-AC[i])/(2*BC[i]*sqrt(AA0[i]));
        BA0[i]=1-(AB0[i]-2*AC[i]*BB[i])/(4*BC[i]*AC[i]);
        if(BB[i]<0)BB[i]=-BB[i];
        if(BA0[i]<0)BA0[i]=-BA0[i];
    }
}
}
}

```

```

/!ELFILTROB: Estructura el filtro pasabajos de BUTTERWORTH
# Parámetros: resp .- Indica si se debe presentar o no la respuesta de frecuencia
#             dc .- Indica que filtro se estructura (existen 2 filtros)
# Retorno : Ninguno
#####
void elfillrob(int resp,int dc)
{
    int i,p,decision=0,deser,mil;
    float fr,fp,fsl,pr,n,nn,n,wc,a,b,r,xx[5],yy[5],nd,ho,wp,wr,awp,m1,m2;
    mil=(MaxX-24*Escx)/2;
    switch(dc){
        case 1: p = fpbA ; fr = frbA ; awp = AfpbA ; n=DBn; break;
        case 2: p = fpbB ; fr = frbB ; awp = AfpbB ; n=DBn;
    }
    fsl=5*fr;wp=2*PI*fp;wr=2*PI*fr;
}

```

```

if(nbuclé !=1){
    wp=2*atan(wp/2/|s1);wr=2*atan(wr/2/|s1);
}
else {
    wr=wr/wp;wp=wp/wp;
}
wc=pow(10,.1*amp)-1;wc=wr*tpow(wc,-.5/n);nn=n;
while (nn>0)nn=nn-2;
if(dc==1)DAnn=nn;else DAnn=nn;
if(dc==1)DBn=n;else DBn=n;
r=nn==0?n/2:(n-1)/2;
i=0;
for(p=1;p<r+1;p++){
    a=wc*cos(PI*(2*p+n-1)/2/n);
    b=wc*sin(PI*(2*p+n-1)/n/2);
    if(a<0){
        xx[i]=-2*a;
        yy[i]=pow(a,2)+pow(b,2);
        ++i;
    }
}
if(nn != 0) pr=wc*cos(PI*(2*p+n-1)/2/n);
ho=pow(wc,n);
if(pr<0)pr=-1*pr;
if(dc==1) pra=pr;
else prb=pr;
if(dc==1) Dha=ho;
else Dhb=ho;
if(resp == 0)
{
    decision=mensaje(292*Escy,50,3,7, " PRESENTAR RESPUESTA DE FRECUENCIA DEL FILTRO (S/N)*");
    if(decision=='s'; decision=='S'){
        n=2;
        switch(dc){
            case 1:
                for(i=0;i<r;+i){
                    DA0[i]=pow(n,2)+xx[i]*yy[i];
                    DA1[i]=-pow(n,2)*2*yy[i];
                    DA2[i]=pow(n,2)+yy[i]-xx[i]*n;
                }
                fsba=fs1;break;
            case 2:
                for(i=0;i<r;+i){
                    DB0[i]=pow(n,2)+xx[i]*yy[i];
                    DB1[i]=-pow(n,2)*2*yy[i];
                    DB2[i]=pow(n,2)+yy[i]-xx[i]*n;
                }
                fsbb=fs1;
        }
        if(dc==1)
            ayuda(mit-31*textwidth("M"),70*Escy,mit+31*textwidth("M"),270*Escy,102);
        else
            ayuda(mit-31*textwidth("M"),70*Escy,mit+31*textwidth("M"),270*Escy,103);
    }
}
if(resp !=0 && nbuclé !=1){
    n=N/nc;
    switch(dc){
        case 1:
            for(i=0;i<r;+i){
                DA0[i]=pow(n,2)+xx[i]*yy[i];
                DA1[i]=-pow(n,2)*2*yy[i];
            }
    }
}

```

```

        DA2[i]=pow(a,2)*yy[i]*xx[i]*n;
    }
    fsha=nt/f/fsl;
    if(fsha>1000)
        fsha=1000;
    break;
}
case 2:
    for(i=0;i<r;+i){
        DA0[i]=pow(a,2)*xx[i]*yy[i];
        DB1[i]=pow(a,2)*2*yy[i];
        DB2[i]=pow(a,2)*yy[i]*xx[i]*n;
    }
    fshb=nt/f/fsl;
    if(fshb>1000)
        fshb=1000;
}
}
if(resp !=0 && nbucle ==1){
    ho=pow(ho,1/r);
    for(i=0;i<r;+i){
        DA0[i]=ho/yy[i];
        n1=xx[i]/(4*yy[i]);
        n2=sqrt(pow(xx[i],2)-8*yy[i]*(1-DA0[i]))/(4*yy[i]);
        DA1[i]=n1+n2;
        if(DA1[i]<0) DA1[i]=n1-n2;
        DA2[i]=1/(DA1[i]*yy[i]);
        if(DA2[i]<0) DA2[i]=-DA2[i];
    }
    pra=pra;
}
}

/*RESPUESTA_BANDA:Presenta la respuesta de frecuencia del filtro pasabajas estructurado
* Parámetros: x.- coordenada X base de la figura
*              y.- coordenada Y base de la figura
*              caso.- Indica de que filtro se trata(existen 4 filtros)
* Retorno : Ninguno
#####
void respuesta(int x,int y,int caso)
{
    float AO[6],B[6],DO[6],fp,fr,fw,a[6],b[6],d[6],R[6],I[6],Alw,Al1;
    float ii,fsl,H,PAR,IMPAR,ress0,resho,ka,kcont,kaa,X1,X2,Y1,Y2,i,iii;
    int r,reso,reso0,il,indp=0,indr=0,indw=0,indka=0,kc=1,cont=0;
    Alw=2;Al1=4;
    switch(case){
        case 1:r=Ann==0?An/2:(An-1)/2;
            resho=Aho;ress0=As0;resn=An;resno=Ann;
            for(iii=1;iii<r+1;+iii){
                a[iii]=(Al1+AO[iii]);b[iii]=2*(AO[iii]-Al1);d[iii]=Al1+AO[iii];
                AO[iii]=Al1+APB[iii]*Alw+AC[iii];
                R[iii]=2*(AC[iii]-Al1);DB[iii]=Al1-ABD[iii]*Alw+AC[iii];
            }
            break;
        case 2:r=Bnn==0?Bn/2:(Bn-1)/2;
            resho=Bho;ress0=Bs0;resn=Bn;resno=Bnn;
            for(iii=1;iii<r+1;+iii){
                a[iii]=(Al1+BA0[iii]);b[iii]=2*(BA0[iii]-Al1);d[iii]=Al1+BA0[iii];
                AO[iii]=Al1+BP[iii]*Alw+BC[iii];
                B[iii]=2*(BC[iii]-Al1);DB[iii]=Al1-BP[iii]*Alw+BC[iii];
            }
            break;
        case 3:resho=Dha;ress0=pra;resn=DA0;resno=DA0nn;

```

```

r=DBnn==0?DBn/2:(DBn-1)/2;
    for(iii=1;iii<r+1;+iii){
        a[iii]=1;b[iii]=2;d[iii]=1;AO[iii]=DA0[iii-1];
        B[iii]=DA1[iii-1];DD[iii]=DA2[iii-1];
    }
    break;
case 4:resho=DBb;ress0=prb;resn=DBn;resnn=DBnn;
r=DBnn==0?DBn/2:(DBn-1)/2;
for(iii=1;iii<r+1;+iii){
    a[iii]=1;b[iii]=2;d[iii]=1;AO[iii]=DB0[iii-1];
    B[iii]=DB1[iii-1];DD[iii]=DB2[iii-1];
}
break;
}
switch(caso){
    case 1:fp=fpeA/fsa;fr=freA/fea;fw=ff/fea;break;
    case 2:fp=fpeB/fsb;fr=freB/feb;fw=ff/feb;break;
    case 3:fp=fpeA/fsa;fr=freA/fsa;fw=ff/fsa;break;
    case 4:fp=fpeB/fsb;fr=freB/fsb;fw=ff/fsb;break;
}
kcont=0;ii=0;ka=2;kaa=0;ii=60*Escx;
settextstyle(SMALL_FONT,HORIZ_DIR,3);
outtextxy(x+ii-8*Escx,y+Escy*196,"2");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
line(ii+x,Escy*320+y,x+ii+355*Escx,Escy*320+y);
line(ii+x,Escy*200+y,x+ii,Escy*320+y);
outtextxy(ii+x-Escx*20,y+Escy*325,"0");
outtextxy(ii+x-Escx*20,y+Escy*210,"1");
outtextxy(x+ii+fp*355*Escx-8*Escx,y+Escy*322,"kfp");
if(fr*355*Escx>fp*355*Escx+24*Escx)
    outtextxy(x+ii+fr*355*Escx-8*Escx,y+Escy*322,"kfr");
else
    outtextxy(x+ii+fr*355*Escx-8*Escx,y+Escy*330,"kfr");
outtextxy(x+ii-Escx*65,y+Escy*200,"IHa(s)l");
outtextxy(x+ii+260*Escx,y+Escy*322,"frecuencia");
setlinestyle(0,0,THICK_WIDTH);
setcolor(9);
while(i<PI){
    X1=cos(i);X2=cos(2*i);Y1=sin(i);Y2=sin(2*i);
    if(resn!=0){
        AO[0]=1+X1;
        B[0]=-Y1;
        d[0]=(Aln+ress0)+(ress0-Aln)*X1;
        DD[0]=-(ress0-Aln)*Y1;
        R[0]=(AO[0]*d[0]+B[0]*DD[0])/(pow(d[0],2)+pow(DD[0],2));
        I[0]=(B[0]*d[0]-AO[0]*DD[0])/(pow(d[0],2)+pow(DD[0],2));
        R[7]=R[0];I[7]=I[0];
        if(resn==1)goto fin;
    }
    for(it=1;it<r+1;it++){
        AO[it]=a[it]+b[it]*X1+d[it]*X2;B[it]=-(b[it]*Y1+d[it]*Y2);d[it]=(AO[it]+B[it])*X1+DD[it]*X2;
        DD[it]=-(B[it]*Y1+DD[it]*Y2);
        R[it]=(AO[it]*d[it]+B[it]*DD[it])/(pow(d[it],2)+pow(DD[it],2));
        I[it]=(B[it]*d[it]-AO[it]*DD[it])/(pow(d[it],2)+pow(DD[it],2));
        if(resn==0){
            if(resn==2){
                R[6]=R[it];I[6]=I[it];goto fin;
            }
            else {
                if(it<2){
                    R[6]=R[it];I[6]=I[it];
                }
            }
        }
    }
}

```





```

else
    outltxy(60*Escx+x-Escx*46,Escy*(320-H*100)+y-5*Escy,"(Afr)");
    indr=1;
}
if(indw==0 && ii>fw+355*Escx+60*Escx){
    line(x+ii,320*Escy+y,x+ii,Escy*250+y);
    indw=1;
}
i+=PI/71042;ii+=Escx;
line(ii+z,Escy*(320-H*100)+y,ii+z,Escy*(320-H*100)+y);
}
}

```

```

/ELFILTROBANDA: Estructura el filtro pasabanda ELIPTICO
# Parámetros: resp .- Indica si se debe presentar o no la respuesta de frecuencia
#             dc .- Indica que filtro se estructura (existen 2 filtros)
# Retorno : Ninguno
#####/
void elfiltrobanda(int resp,int dc)
{
    int i,j,r,b,il,l,decision=0,mil;
    float fr,fp,w1,w2,fs1,k1,k2,k0,q0,q,DD,awp,fa,alpa,k,y,yy,B,d,a,g,u,nd,ho;
    float bb,nn,n,lan,s1=0,s2=0,s0,w,oh,vi,bi,o1,o2,o3,Anda[6],Canda[6],Banda[6];
    mil=(MaxX-24*Escx)/2;
    switch(dc){
        case 1:fo=Afo;B=AW;fr=frbanA;awp=AfpbanA;n=BAN;break;
        case 2:fo=Bfo;B=BP;fr=frbanB;awp=AfpbanB;n=BBn;
    }
    fp=B/2;fs1=2*fo;k0=tan(PI*fp/fs1)/tan(PI*fr/fs1);
    B=1.5*B*PI/fo;yy=1.3*PI;k1=sqrt(1-pow(k0,2));k2=sqrt(k1);
    q0=.5*(1-k2)/(1+k2);q=q0+2*pow(q0,5)+15*pow(q0,9)+150*pow(q0,13);
    lan=.5/n*log((pow(10,.05*awp)+1)/(pow(10,.05*awp)-1));
    for(i=0;i<5;i++){
        j=i*(i+1);s1=s1+pow(-1,i)*pow(q,j)*sinh((2*i+1)*lan);
    }
    for(j=1;j<7;j++){
        l=pow(j,2);s2=s2+pow(-1,j)*pow(q,l)*cosh(2*j*lan);
    }
    s0=2*pow(q,.25)*s1/(1+2*s2);
    s0=s0>0? s0:-1*s0;
    w=sqrt((1+k0*pow(s0,2))*(1+pow(s0,2)/k0));
    nn=n;
    while (nn>0)nn=nn-2;
    if(dc==1)BAN=nn;else BBn=nn;
    if(dc==1)BAN=n;else BBn=n;
    r=nn==0?n/2:(n-1)/2;
    for(it=1;it<r+1;it++){
        u=nn==0?it-.5:it;
        o1=0;o2=0;
        for(i=0;i<15;i++){
            b=i*(i+1);o3=(2*i+1)*PI/n*u;o1=o1+pow(-1,i)*pow(q,b)*sin(o3);
        }
        for(i=1;i<17;i++){
            b=pow(i,2);o2=n2*pow(-1,i)*pow(q,b)*cos(2*i*PI*u/n);
        }
        oh=2*pow(q,.25)*o1/(1+2*n2);
        vi=sqrt((1-k0*pow(oh,2))*(1-pow(oh,2)/k0));Anda[i]=1/pow(oh,2);
        bi=1+pow(s0*foh,2);Canda[i]=(pow(s0*vi,2)+pow(oh*w,2))/pow(bi,2);
        Banda[i]=2*s0*vi/bi;
    }
    bb=1;
    for(i=1;i<r+1;i++)

```



```

AH4[i]=pow(a,4)-Banda[i]*Btpow(a,3)+g-Banda[i]*Btyy*tpow(yy,2);
AH3[i]=-4*tpow(a,4)+24*Banda[i]*Btpow(a,3)-24*Banda[i]*Btyy*tpow(yy,2);
AH2[i]=6*tpow(a,4)-24*tpow(a,3)+6*tpow(yy,2);
AH1[i]=-4*tpow(a,4)-24*Banda[i]*Btpow(a,3)+24*Banda[i]*Btyy*tpow(yy,2);
AH0[i]=pow(a,4)+Banda[i]*Btpow(a,3)+g+Banda[i]*Btyy*tpow(yy,2);
}
else {
  BH4[i]=pow(a,4)+d+tpow(yy,2);
  BH3[i]=4*tpow(yy,2)-4*tpow(a,4);
  BH2[i]=6*tpow(a,4)-24*tpow(yy,2);
  BH1[i]=BH3[i];
  BH0[i]=BH4[i];
  BH4[i]=pow(a,4)-Banda[i]*Btpow(a,3)+g-Banda[i]*Btyy*tpow(yy,2);
  BH3[i]=-4*tpow(a,4)+24*Banda[i]*Btpow(a,3)-24*Banda[i]*Btyy*tpow(yy,2);
  BH2[i]=6*tpow(a,4)-24*tpow(yy,2);
  BH1[i]=-4*tpow(a,4)-24*Banda[i]*Btpow(a,3)+24*Banda[i]*Btyy*tpow(yy,2);
  BH0[i]=pow(a,4)+Banda[i]*Btpow(a,3)+g+Banda[i]*Btyy*tpow(yy,2);
}
}
if(dc==1){
  AJ2=a*B;AJ22=pow(a,2)-s0*B*tyy;AJ1=-pow(a,2)+2*tyy;AJ0=pow(a,2)+s0*B*tyy;
}
else {
  BJ2=a*B;BJ22=pow(a,2)-s0*B*tyy;BJ1=-pow(a,2)+2*tyy;BJ0=pow(a,2)+s0*B*tyy;
}
}
}

```

/\*RESPUESTA\_BANDA:Presenta la respuesta de frecuencia del filtro pasabanda estructurado

\* Paraelros: x ,- coordenada X base de la figura

\* y ,- coordenada Y base de la figura

\* caso ,- Indica de que filtro se trata(existen 2 filtros)

\* Retorno : Ninguno

#####

void respuesta\_banda(int x,int y,int caso)

{

float a,b,d,c,R[8],l[8],x1,x2,y1,y2,x3,x4,y3,y4,H,PAR,IMPAR,ii,resho,i,fo,fw,HN;

int r,reshn,resn,il,indo=0,indw=0,indw2=0;

HN=187;ii=60\*Escx;

switch(caso){

case 1: r=BA\*nn==0?BA\*2:(BA\*nn-1)/2;  
resho=BA\*ho;resn=BA\*nn;resnn=BA\*nn;  
fo=Afo/fsbana;fw=ff/fsbana;  
break;

case 2: r=BB\*nn==0?BB\*2:(BB\*nn-1)/2;  
resho=BB\*ho;resn=BB\*nn;resnn=BB\*nn;  
fo=Bfo/fsbana;fw=ff/fsbana;  
break;

case 3: r=1;  
resho=.3526;resn=2;resnn=0;fo=.5;  
AH0[1]=160.50;AH1[1]=3.61;AH2[1]=26.27;AH3[1]=1.82;AH4[1]=74.50;  
AH0[1]=197.36;AH1[1]=2.71;AH2[1]=-133.28;AH3[1]=2.71;AH4[1]=197.36;

settextstyle(1,1,HORN\_WIDTH);

line(ii\*x,Escy+220\*y,x+ii+130\*Escx,Escy+220\*y);

line(ii\*x,Escy+241\*y,x+ii+55\*Escx,Escy+241\*y);

line(ii\*x+Escx+55,Escy+241\*y,x+ii+Escx+55,Escy+320\*y);

outtextxy(ii\*x-Escx+78,y+Escy+235,"H (Afp)");

outtextxy(ii\*x-Escx+78,y+Escy+302,"H (Afr)");

outtextxy(ii\*x+Escx+33,y+Escy+323,"fr");

outtextxy(ii\*x+Escx+63,y+Escy+323,"fr");

settextstyle(SMALL\_FONT,HORIZ\_DIR,4);

outtextxy(x+ii-67\*Escx,y+Escy+235,"(fp)");

```

        outltxy(x+ii-67*Escx,y+Escy*302,"(fr)");
        outltxy(x+ii+35*Escx-lexwidth("N"),y+Escy*321,"X");
        outltxy(x+ii+63*Escx-lexwidth("N"),y+Escy*321,"K");
        line(ii+x,Escy*305*y,x+ii+46*Escx,Escy*305*y);
        line(ii+x+Escx*46,Escy*305*y,x+ii+Escx*46,Escy*320*y);
        indw=1;indw2=1;caso=1;break;
    }

    i=0;
    setltxstyle(SMALL_FONT,HORIZ_DIR,3);
    outltxy(x+ii-8*Escx,y+Escy*172,"2");
    setltxstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setlinesyle(0,0,HORN_WIDTH);
    line(ii+x,Escy*320*y,x+ii+280*Escx,Escy*320*y);
    line(ii+x,Escy*200*y,x+ii,Escy*320*y);
    outltxy(ii+x+277*Escx,y+Escy*317,">");
    outltxy(ii+x-3,y+Escy*200,"^");
    outltxy(ii+x-Escx*20,y+Escy*315,"0");
    outltxy(ii+x-Escx*20,y+Escy*213,"1");
    outltxy(x+ii-Escx*75,y+Escy*196," Iln(s)I");
    outltxy(x+ii+190*Escx,y+Escy*323,"frecuencia");
    setlinesyle(0,0,THICK_WIDTH);
    setcolor(7);
    while(i<P1){
        x1=cos(i);x2=cos(2*i);y1=-sin(i);y2=-sin(2*i);
        x3=cos(3*i);x4=cos(4*i);y3=-sin(3*i);y4=-sin(4*i);
        if(resn!=0){
            if(caso == 1){
                a=Aj2*(1-x2);b=-Aj2*y2;c=Aj2*x2+Aj1*x1+Aj0;d=Aj2*y2+Aj1*y1;
            }
            if(caso == 2){
                a=Bj2*(1-x2);b=-Bj2*y2;c=Bj2*x2+Bj1*x1+Bj0;d=Bj2*y2+Bj1*y1;
            }
            R[0]=(a+c+b*d)/(pow(c,2)+pow(d,2));
            I[0]=(c+b-a*d)/(pow(c,2)+pow(d,2));
            R[7]=R[0];I[7]=I[0];
            if(resn==1)goto fin;
        }
        for(il=1;il<r+1;il++){
            if(caso == 1){
                a=Ah4[il]*x4+Ah3[il]*x3+Ah2[il]*x2+Ah1[il]*x1+Ah0[il];
                b=Ah4[il]*y4+Ah3[il]*y3+Ah2[il]*y2+Ah1[il]*y1;
                c=Ah4[il]*x4+Ah3[il]*x3+Ah2[il]*x2+Ah1[il]*x1+Ah0[il];
                d=Ah4[il]*y4+Ah3[il]*y3+Ah2[il]*y2+Ah1[il]*y1;
            }
            if(caso == 2){
                a=Bh4[il]*x4+Bh3[il]*x3+Bh2[il]*x2+Bh1[il]*x1+Bh0[il];
                b=Bh4[il]*y4+Bh3[il]*y3+Bh2[il]*y2+Bh1[il]*y1;
                c=Bh4[il]*x4+Bh3[il]*x3+Bh2[il]*x2+Bh1[il]*x1+Bh0[il];
                d=Bh4[il]*y4+Bh3[il]*y3+Bh2[il]*y2+Bh1[il]*y1;
            }
            R[il]=(a+c+b*d)/(pow(c,2)+pow(d,2));
            I[il]=(c+b-a*d)/(pow(c,2)+pow(d,2));
            if(resn==0){
                if(resn==2){
                    R[6]=R[il];I[6]=I[il];goto fin;
                }
                else {
                    if(il<2){
                        R[6]=R[il];I[6]=I[il];
                    }
                    if(il>1){
                        PAR=R[6];
                    }
                }
            }
        }
    }

```

```

R[6]=R[6]*R[i1]-I[6]*I[i1];
I[6]=PAR*I[i1]+R[i1]*I[6];
}
}
else {
    IMPAR=R[7];
    R[7]=R[7]*R[i1]-I[7]*I[i1];
    I[7]=IMPAR*I[i1]+R[i1]*I[7];
}
}
fin:if(resna==0){
    H=pow(resho,2)*(pow(R[6],2)+pow(I[6],2));
}
else {
    H=pow(resho,2)*(pow(R[7],2)+pow(I[7],2));
}
if(indo==0 && ii>f0*HH*Escx+60*Escx){
    sellinesyle(1,1,HORM_WIDTH);
    line(x+ii,320*Escy+y,x+ii,Escy*(320-H*100)+y);
    outtextxy(x+ii,320*Escy+y,"fo");
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    outtextxy(x+ii-tet*widh("M"),y+Escy*321,"K");
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    sellinesyle(0,0,INICK_WIDTH);
    indo=1;
}
if(indw==0 && ii>fw*HH*Escx+60*Escx){
    line(x+ii,320*Escy+y,x+ii,Escy*250+y);
    indw=1;
}
if(indw2==0 && ii>2*fw*HH*Escx+60*Escx){
    line(x+ii,320*Escy+y,x+ii,Escy*250+y);
    indw2=1;
}
i+=PI/HH;ii+=Escx;
line(ii*x,Escy*(320-H*100)+y,ii*x,Escy*(320-H*100)+y);
}
}

```

```

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
PANTALLA.C
Archivo de funciones auxiliares
*/

```

```

#include "dos.h"
#include "math.h"
#include "conio.h"
#include "stdlib.h"
#include "stdio.h"
#include "graphics.h"
#include "proyfil.h"
#include "bios.h"
int GraphDriver,GraphMode,MaxColors,ErrorCode;;
float MaxX, MaxY,Escx,Escy,R10,R11;
extern char far tbuf;
extern int ch,nbucle;

```

```

/*INITIALIZE: Consigue el modo grafico
* Parámetros :Ninguno
* Retorno : Ninguno

```

```

#####/
void Initialize(void)
{
    GraphDriver = DETECT;
    initgraph( &GraphDriver, &GraphMode, "" );
    MaxColors = getmaxcolor() + 1;
    MaxX = getmaxx();
    MaxY = getmaxy();
    Escx =(float) getmaxx()/719;
    Escy =(float) getmaxy()/347;
}

/*LEMA: Pone un titulo en pantalla
# Parámetros : yl.- posición horizontal del titulo
#               lamano.- lamano de los caracteres
#               titulo.- mensaje
# Retorno : Ninguno
#####/
void lea(int yl,int lamano ,char *titulo )
{
    settextjustify(CENTER_TEXT, TOP_TEXT);
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, lamano);
    outtextxy(MaxX/2, yl, titulo);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    settextjustify(LEFT_TEXT, TOP_TEXT);
}

/*STATUSLINE: Pone un mensaje al final de la pantalla
# Parámetros : caso.- Indicador de la longitud del mensaje
#               msg.- mensaje
# Retorno : Ninguno
#####/
void StatusLine( int caso, char *msg )
{
    int height;
    setviewport( 0, 0, MaxX, MaxY, 1 );
    setcolor( MaxColors - 1 );
    settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    settextjustify( CENTER_TEXT, TOP_TEXT );
    setlinestyle( SOLID_LINE, 0, HORN_WIDTH );
    setfillstyle( EMPTY_FILL, 0 );
    height = textheight( "N" );
    switch(caso){
        case 1: bar( 10*Escx, MaxY-(height*4), MaxX-32*Escx, MaxY );
                rectangle(10*Escx, MaxY-(height*4), MaxX-32*Escx, MaxY );
                break;
        case 2: bar( 100*Escx, MaxY-(height*4), MaxX-132*Escx, MaxY );
                rectangle(100*Escx, MaxY-(height*4), MaxX-132*Escx, MaxY );
                break;
    }
    outtextxy( MaxX/2-10*Escx, MaxY-(height*2), msg );
}

/*GPRINTF: Semejante a PRINTF pero en modo grafico
# Parámetros :xloc.- coordenada X del punto de pantalla donde inicia el mensaje
#               yloc.- coordenada Y del punto de pantalla donde inicia el mensaje
#               separ.- valor de la separación entre lineas
#               *fal.- Mensaje a escribirse
# Retorno : Valor de la coordenada Y que queda para la proxima linea
#####/
int gprintf( int xloc, int yloc, int separ, char *fal, ... )
{

```

```

va_list argptr;
char str[140];
int cnt;

if((separ==2) ?yloc += textheight( "H" ) ;
sellinesize( 0, 0, NORM_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
if(separ !=3)setcolor(15);
va_start( argptr, int );
cnt = vsprintf( str, fmt, argptr );
outtextxy( xloc, yloc, str );
if(separ==2) ?yloc += textheight( "H" )/4;
if(separ==1 ||separ==3) ?yloc += textheight( "H" )/2+Escy;
va_end( argptr );
return( cnt );
}

/*MAINWINDOW:despliega en pantalla un recuadro
# Parámetros : x1 .- coordenada X de la esquina superior izquierda del recuadro
#              y1 .- coordenada Y de la esquina superior izquierda del recuadro
#              x2 .- coordenada X de la esquina inferior derecha del recuadro
#              y2 .- coordenada Y de la esquina inferior derecha del recuadro
#              estilo.- indicador de tipo de relleno del borde del recuadro
#              mensaje.- Indica si se debe poner mensaje de ayuda y que caso
#              #titulo.- TITULO que se pone al recuadro
#Retorno :      Ninguno
#####/
void MainWindow(int x1,int y1,int x2,int y2,int estilo,int mensaje,char #titulo )
{
int hl,w,xp;
char buffer[40];
if(mensaje==0){
setcolor( MaxColors - 1 );
selviewpor( x1+7*Escx,y1+7*Escy,x2-7*Escx,y2-7*Escy, 1 );
DrawBorderg();
selviewpor(x1,y1,x2,y2, 1 );
DrawBorderg();
setfillstyle(estilo,MaxColors-1);
floodfill(5*Escx,5*Escy,MaxColors-1);
}
w=x2-x1;hl=y2-y1;xp=(w/2-120)*Escx;
if(mensaje==0)outtextxy(xp,7*Escy+textheight("H")/2-1,titulo);
if(mensaje==2 || mensaje==3 ||mensaje==1 || mensaje==4 || mensaje==5 ||mensaje==6 )
selviewpor(x1,y1,x2,y2,1);
if(mensaje==1)outtextxy(xp-80*Escx,ht-17*Escy,"[ESC] Salir de AYUDA [PgDn] Abajo ");
if(mensaje==2)outtextxy(xp-80*Escx,ht-17*Escy,"[ESC] Salir de AYUDA [PgUp]Arriba [PgDn] Abajo ");
if(mensaje==3)outtextxy(xp-80*Escx,ht-17*Escy,"[ESC] Salir de AYUDA [PgUp]Arriba ");
if(mensaje==4)outtextxy(xp-80*Escx,ht-17*Escy," [ ] Pulse cualquier tecla para salir de AYUDA ");
if(mensaje==5)outtextxy(xp-80*Escx,ht-17*Escy," Plantilla de un filtro de tercer orden ");
if(mensaje==6)outtextxy(xp-80*Escx,ht-17*Escy," [ E S C ] para salir de AYUDA ");
selviewpor((x1+9*Escx,y1+7*Escy+textheight("H")+3,x2-9*Escx,y2-20*Escy,1);
DrawBorderg();
selviewpor(x1,y1,x2,y2, 1 );
}

/*MAINWINDOW_LINEA:despliega en pantalla un recuadro
# Parámetros : x1 .- coordenada X de la esquina superior izquierda del recuadro
#              y1 .- coordenada Y de la esquina superior izquierda del recuadro
#              x2 .- coordenada X de la esquina inferior derecha del recuadro
#              y2 .- coordenada Y de la esquina inferior derecha del recuadro
#              estilo.- indicador de tipo de relleno del borde del recuadro
# Retorno :      Ninguno
#####/

```



```

void MainWindow_linea(int x1,int y1,int x2,int y2,int estilo)
{
    setcolor( MaxColors - 1 );
    setviewport( x1+3*Escx,y1+3*Escy,x2-3*Escx,y2-3*Escy, 1 );
    DrawBorderg();
    setviewport(x1,y1,x2,y2, 1 );
    DrawBorderg();
    setfillstyle(estilo,MaxColors-1);
    floodfill(2*Escx,2*Escy,MaxColors-1);
    setviewport(x1,y1,x2,y2, 1 );
}

//TECLA: Entrada de caracteres desde el teclado
* Parámetros : Ninguno
* Retorno : Código del caracter
#####/
short lecla(void)
{
    union REGS r;
    r.ax=0;
    int86(0x16,&r,&r);
    return((r.h.al==0)?(short) r.h.ah:(short) r.h.al);
}

//DRAWBORDERG: Dibuja un cuadro sobre en los límites de la ventana grafica actual
* Parámetros :Ninguno
* Retorno : Ninguno
#####/
void DrawBorderg(void)
{
    struct viewporttype vp;
    setcolor( MaxColors - 1 );
    setlinestyle( SOLID_LINE, 0, NORM_WIDTH );
    getviewsettings( &vp );
    rectangle( 0, 0, vp.right-vp.left, vp.bottom-vp.top );
}

//MENSAJE:Despliega un mensaje de acuerdo al caso
* Parámetros : diy1 .- coordenada Y de la esquina superior izquierda del recuadro
* caracteres .- cantidad de caracteres del mensaje
* tipo .- indicador de que caso de mensaje se ha solicitado
* help .- indicador de que caso de ayuda tiene el mensaje
* msg .- mensaje
* Retorno : elección dada al mensaje
#####/
int mensaje (int diy1,int caracteres,int tipo,int help,char msg)
{
    int decision,col,loc,dix2,diy2,dix1,pp,xi,yi,op,x,y;
    char line[80];
    FILE *p;
    size_t ss;
    va_list argptr;
    char str[140],*pl;
    struct viewporttype vp;
    loc = textwidth( "M" ) + 2;
    getviewsettings( &vp );
    dix1=( (vp.right-vp.left)-2*Escx-loc*caracteres)/2;
    dix2=dix1+loc*caracteres;
    diy2=diy1+textheight("M")+12;
    if(dix1<5)dix1=5;
    if(dix2>MaxX-5)dix2=MaxX-5;
    ss = imagesize (dix1, diy1, dix2,diy2);
    pp = _graphgetch( ss);
}

```

```

getlineage (dix1, diy1, dix2, diy2, pp);
setlinestyle (0, 0, NORM_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
setfillstyle (SOLID_FILL, 0);
bar (dix1, diy1, dix2, diy2);
setcolor (getmaxcolor());
rectangle (dix1+1, diy1+1, dix2-1, diy2-1);
setfillstyle (SOLID_FILL, getmaxcolor());
bar (dix1+4, diy1+4, dix2-4, diy2-4);
setcolor (0);
outtextxy(dix1+10,diy1+7, msg);
if(tipo !=10 && tipo !=11)xi=(int)(dix2-50*Escx);
else xi=(int)(dix2-120*Escx);
yi=diy1+6;pl = linea;
switch(tipo){
    case 0: getch();break;
    case 2: pl=linea;
            op = gread (xi,yi, linea,0, 1, op,help,tipo);
            ++pl;
            decision = linea();break;
    case 3: pl=linea;
            op = gread (xi,yi, linea,0, 1, op,help,tipo);
            decision = linea(0);break;
    case 4: pl=linea;
            op = gread (xi,yi, linea,0, 4, op,help,tipo);
            pl++;
            decision = atoi(pl);break;
    case 5: decision=tecla();
            if(decision<58 && decision>47)decision=decision-48;break;
    case 6: pl=linea;
            op = gread (xi,yi, linea,0, 1, op,help,tipo);
            decision = linea(0);
            decision=decision-48;break;
    case 10: pl=linea;
            op = gread (xi,yi, linea,0, 10, op,help,2);
            ++pl;
            R10 = atof(pl);break;
    case 11: pl=linea;
            op = gread (xi,yi, linea,0, 10, op,help,2);
            ++pl;
            R11 = atof(pl);break;
    case 15: do{
                decision=tecla();
            }while(decision !=323 && decision !=324);
            break;
    case 20: x=loc+dix1;y=diy1+6;
            gprintf(&x,&y,3,"Frecuencias de OSCILACION: INFERIOR: % .2f SUPERIOR % .2f",R10,R11);
            getch();break;
}
putimage (dix1, diy1, pp, COPY_PUT);
_graphfree mem (pp, ss);
setlinestyle (0, 0, NORM_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
setfillstyle (SOLID_FILL, getmaxcolor());
setcolor (getmaxcolor());
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
return decision;
}

```

/40GRPLL:Despliega texto en pantalla

\* Parámetros : x .- coordenada X de inicio del texto

```

#           y .- coordenada Y de inicio del texto
#           des .- indicador de longitud del texto
# Retorno : Ninguno
#####
void BGRFILL(int x,int y,int des)
{
    setfillstyle (SOLID_FILL,1);
    settextjustify(LEFT_TEXT,TOP_TEXT);
    setlinestyle (0, 0, NORM_STYLE);
    if(des ==1)
        settextstyle (TRIPLEX_FONT, HORIZ_DIR, 1);
    if(nbucle == 0 ;; nbucle == 1 ;; nbucle == 3 ;; nbucle == 4 )
        gprintf(%x,%y,2,"FB) Comparador de Fase");
    gprintf(%x,%y,2,"<FB) Filtro Pasabajos ");
    if(nbucle != 0 && des == 1 )
        gprintf(%x,%y,2,"<X) Multiplicador Analógico ");
    if(nbucle != 0 && des == 0 )
        gprintf(%x,%y,2,"<X) Multiplicador ");
    if( des == 1 )
        gprintf(%x,%y,2,"<VCO) Oscilador Controlado por Voltaje");
    if( des == 0 )
        gprintf(%x,%y,2,"<VCO) Osc. Cont. Voltaje");
    if(nbucle == 7 )
        gprintf(%x,%y,2,"<+) Sumador");
    if(nbucle != 0 )
        gprintf(%x,%y,2,"<ADC) Ecuilizador");
    if(nbucle == 2 ;; nbucle == 5 ;; nbucle == 6 ;; nbucle == 7 )
        gprintf(%x,%y,2,"<90) Defasador de 90");
    if(nbucle == 1 && des ==-1)
        gprintf(%x,%y,2,"<X2) Duplicador de Frecuencia ");
    if(nbucle == 1 && des ==0)
        gprintf(%x,%y,2,"<X2) Duplicador de Frec.");
    settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
}

```

/\*BARRA: Presenta una barra en pantalla

```

# Parámetros :x1 .- coordenada X de la esquina superior izquierda de la barra
#           y1 .- coordenada Y de la esquina superior izquierda de la barra
#           x2 .- coordenada X de la esquina inferior derecha de la barra
#           y2 .- coordenada Y de la esquina inferior derecha de la barra
# Retorno : Ninguno
#####

```

```

void barra(int x1,int y1,int x2,int y2)
{
    int yy1;
    setcolor(15);
    setfillstyle(1,0);
    yy1=getcolor();
    bar3d(x1,y1,x2,y2,10,1);
    setfillstyle(1,3);
    floodfill(x1+5,y1-2,yy1);
    setfillstyle(1,3);
    floodfill(x2+2,y2-10,yy1);
}

```

/\*BARRA: Presenta una barra en pantalla

```

# Parámetros : x1 .- coordenada X de la esquina superior izquierda
#           y1 .- coordenada Y de la esquina superior izquierda
#           x2 .- coordenada X de la esquina inferior derecha
#           y2 .- coordenada Y de la esquina inferior derecha
#           espesor .- espesor de la barra
#           modo.- relleno de la cara lateral

```

```

f          relleno .- relleno de la cara superior
f Retorno :   Ninguno
#####/
void barra(int x1,int y1,int x2,int y2,int espesor,int modo,int relleno)
{
int yy1;
    setbcolor(7);
    setfillstyle(1,0);
    setcolor(15);
    yy1=getcolor();
    bar3d(x1,y1,x2,y2,espesor,1);
    rectangle(x1+5,y1+3,x2-5,y2-3);
    setfillstyle(1,15);
    floodfill(x1+5,y1-2,yy1);
    setfillstyle(1,15);
    floodfill(x2+2,y2-10,yy1);
    setfillstyle(mod0,YELLOW);
    floodfill(x1+2,y1+2,yy1);
}

/!CIRCULO: Presenta un CIRCULO sombreado en pantalla
f Parámetros :  x1 .- coordenada X del centro
f              y1 .- coordenada Y del centro
f              r .- radio
f Retorno :     Ninguno
#####/
void circulo(int x1,int y1,int r)
{
float yy1,alfa,x11,y11,x22,y22,x2,y2;
    setfillstyle(0,BLUE);
    yy1=getcolor();
    circle(x1,y1,r);
    x2=x1+r+4;y2=y1-r;
    arc(x2,y2,-80,170,r);
    alfa=PI/2-atan((y1-y2)/(x2-x1));
    x11=x1-(r*cos(alfa)-2);
    y11=y1-(r*sin(alfa)-2);
    x22=x2-(r*cos(alfa)-2);
    y22=y2-(r*sin(alfa)-2);
    line(x11,y11,x22,y22);
    x11=x1+(r*cos(alfa)-1);
    y11=y1+(r*sin(alfa)-1);
    x22=x2+(r*cos(alfa)-1);
    y22=y2+(r*sin(alfa)-1);
    line(x11,y11,x22,y22);
    setfillstyle(1,BLUE);
    setcolor(WHITE);
    floodfill(x2,y2,yy1);
}

/!FORMABUC2: Presenta el bucle de cos(α) general en pantalla
f Parámetros :  x .- coordenada X base
f              y .- coordenada Y base
f Retorno :     Ninguno
#####/
void formaabuc2(float x, float y)
{
    settextjustify(CENTER_TEXT, TOP_TEXT );
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setcolor(9);
    barra(x+Escx*280,-y+Escy*190,x+Escx*330,-y+Escy*110);
    barra(x+Escx*480,-y+Escy*190,x+Escx*530,-y+Escy*110);
}

```

```

barra(x+Escx#120,-y+Escy#200,x+Escx#130,-y+Escy#220);
barra(x+Escx#190,-y+Escy#125,x+Escx#240,-y+Escy#145);
barra(x+Escx#195,-y+Escy#160,x+Escx#235,-y+Escy#180);
circulo(x+Escx#215,-y+Escy#100,Escx#10);
circulo(x+Escx#215,-y+Escy#210,Escx#10);
circulo(x+Escx#380,-y+Escy#135,Escx#10);
sellinesstyle(0,1,THICK_WIDTH);
setcolor(10);
line(x+Escx#130,-y+Escy#100,x+Escx#202,-y+Escy#100);
line(x+Escx#335,-y+Escy#100,x+Escx#367,-y+Escy#100);
line(x+Escx#227,-y+Escy#100,x+Escx#277,-y+Escy#100);
line(x+Escx#335,-y+Escy#100,x+Escx#477,-y+Escy#100);
line(x+Escx#75,-y+Escy#135,x+Escx#127,-y+Escy#135);
line(x+Escx#130,-y+Escy#210,x+Escx#202,-y+Escy#210);
line(x+Escx#227,-y+Escy#210,x+Escx#277,-y+Escy#210);
line(x+Escx#380,-y+Escy#210,x+Escx#380,-y+Escy#133+10*Escx);
line(x+Escx#380,-y+Escy#100,x+Escx#380,-y+Escy#137+10*Escx);
line(x+Escx#370,-y+Escy#135,x+Escx#245,-y+Escy#135);
line(x+Escx#215,-y+Escy#145,x+Escx#215,-y+Escy#155);
line(x+Escx#215,-y+Escy#100,x+Escx#215,-y+Escy#212+10*Escx);
line(x+Escx#130,-y+Escy#100,x+Escx#130,-y+Escy#210);
line(x+Escx#215,-y+Escy#70+10*Escx,x+Escx#215,-y+Escy#120);
line(x+Escx#335,-y+Escy#210,x+Escx#380,-y+Escy#210);
setcolor(12);
outtextxy(x+Escx#305,-y+Escy#97,"FPR1");
outtextxy(x+Escx#305,-y+Escy#207,"FPR2");
outtextxy(x+Escx#215,-y+Escy#132,"VCU");
outtextxy(x+Escx#210,-y+Escy#168," 90");
outtextxy(x+Escx#305,-y+Escy#96,"AOC");
outtextxy(x+Escx#215,-y+Escy#97,"X");
outtextxy(x+Escx#215,-y+Escy#207,"X");
outtextxy(x+Escx#380,-y+Escy#132,"X");
outtextxy(x+Escx#202,-y+Escy#97,">");
outtextxy(x+Escx#202,-y+Escy#207,">");
outtextxy(x+Escx#362,-y+Escy#97,">");
outtextxy(x+Escx#277,-y+Escy#97,">");
outtextxy(x+Escx#477,-y+Escy#97,">");
outtextxy(x+Escx#215,-y+Escy#107,"^");
outtextxy(x+Escx#254,-y+Escy#132,"^");
outtextxy(x+Escx#380,-y+Escy#120,"v");
outtextxy(x+Escx#380,-y+Escy#142,"^");
outtextxy(x+Escx#277,-y+Escy#207,">");
outtextxy(x+Escx#215,-y+Escy#195,"v");
outtextxy(x+Escx#215,-y+Escy#147,"v");
outtextxy(x+Escx#126,-y+Escy#132,">");
selltextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx#95,-y+Escy#120,"[1]");
outtextxy(x+Escx#187,-y+Escy#106,"[2]");
outtextxy(x+Escx#185,-y+Escy#187,"[8]");
outtextxy(x+Escx#305,-y+Escy#120,"[7]");
outtextxy(x+Escx#262,-y+Escy#85,"[3]");
outtextxy(x+Escx#365,-y+Escy#85,"[5]");
outtextxy(x+Escx#363,-y+Escy#195,"[9]");
outtextxy(x+Escx#260,-y+Escy#195,"[6]");
outtextxy(x+Escx#370,-y+Escy#85,"[4]");
sellinesstyle(0,0,HORN_WIDTH);
settextjustify(LEFT_TEXT, TOP_TEXT);
selltextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(15);
}

```

```

/4GrPrint:Imprime la pantalla actual del monitor
# Parámetros : Ninguno

```

```

* Retorno : Ninguno
#####/
void GrPrint (void)
{
int h,v,dec,ia,c,f,i,j,k,m,w,xi,yi;
FILE *p;
selviewport(0,0,MaxX,MaxY,1);
h=getmaxx();v=getmaxy();xi=textwidth("M");yi=textheight("H");
ia=biosprint(2,0,0);
if(ia == 144){
dec=mensaje(205*Escy,40,15,7, "Imprimir Pantalla? [F9]Imprimir [F10]Cancelar");
if(dec==323)
ia=biosprint(2,0,0);
if(dec==324)return;
}
if(ia != 144){
no:if(ia==24)dec=mensaje(105*Escy,49,15,7, " ERROR ! IMPRESORA no LISTA [F9] Reintentar [F10] Cancelar");
if(ia==200)dec=mensaje(155*Escy,49,15,7, " ERROR ! impresora apagada [F9] Reintentar [F10] Cancelar");
if(dec==323){
ia=biosprint(2,0,0);
if(ia !=144)goto no;
}
if(dec==324)return;
}
p = fopen ("prn", "w");
fprintf (p, "\x1B\x40\xD\xA\x1B\x41\x1B");
i=0;
while (i <= v)
{
fprintf (p, "\x1B\x4C\xD0\x12");
for (u=0;u<=h;u++)
{
a=0;j=120;
for (k=0;k<i;k++)
{
f=i+k;w = getpixel (c,f);
if ( w != 0)
a = a + j;
j = j / 2;
}
putc ((char)a,p);
}
fprintf (p, "\x0A\x0D");
i = i + yi;
}
fprintf (p, "\x1B\x40\xD\x0A");
fclose (p);
}

```

/4

Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.

BUCLES.C

Archivo que presentar los diagramas de bloques de los bucles

```

*/
#include <math.h>
#include <graphics.h>
#include "proyfil.h"
extern float Escx,Escy,MaxX,MaxY;

```

/4FORMABUC5: Presenta el bucle de costas con delección dentro de un brazo

\* Parámetros : x .- coordenada X base

```

}          y .- coordenada Y base
} Retorno : Ninguno
#####//
void formabuc5(float x,float y)
{
  sellerljustify( CENTER_TEXT, TOP_TEXT );
  selltextstyle(DEFAULT_FONT,HORIZ_DIR,1);
  barra(x+Escx#230,100+Escy-y,x+Escx#280,120+Escy-y);
  barra(x+Escx#230,200+Escy-y,x+Escx#280,220+Escy-y);
  barra(x+Escx#330,100+Escy-y,x+Escx#380,120+Escy-y);
  barra(x+Escx#145,135+Escy-y,x+Escx#195,155+Escy-y);
  barra(x+Escx#155,170+Escy-y,x+Escx#185,190+Escy-y);
  circulo(x+Escx#170,110+Escy-y,12+Escx);
  circulo(x+Escx#440,145+Escy-y,12+Escx);
  circulo(x+Escx#170,210+Escy-y,12+Escx);
  setlinestyle(0,1,THICK_WIDTH);
  line(x+Escx#90,110+Escy-y,x+Escx#90,210+Escy-y);
  line(x+Escx#70,160+Escy-y,x+Escx#90,160+Escy-y);
  line(x+Escx#170,212+Escy-y-12+Escx,x+Escx#170,190+Escy-y);
  line(x+Escx#170,155+Escy-y,x+Escx#170,167+Escy-y);
  line(x+Escx#200,145+Escy-y,x+Escx#428,145+Escy-y);
  line(x+Escx#385,110+Escy-y,x+Escx#440,110+Escy-y);
  line(x+Escx#440,110+Escy-y,x+Escx#440,138+Escy-y);
  line(x+Escx#440,143+Escy-y+12+Escx,x+Escx#440,210+Escy-y);
  line(x+Escx#440,210+Escy-y,x+Escx#285,210+Escy-y);
  line(x+Escx#227,210+Escy-y,x+Escx#182,210+Escy-y);
  line(x+Escx#90,210+Escy-y,x+Escx#155,210+Escy-y);
  line(x+Escx#170,108+Escy-y+12+Escx,x+Escx#170,132+Escy-y);
  line(x+Escx#90,110+Escy-y,x+Escx#155,110+Escy-y);
  line(x+Escx#182,110+Escy-y,x+Escx#227,110+Escy-y);
  line(x+Escx#285,110+Escy-y,x+Escx#328,110+Escy-y);
  outtextxy(x+Escx#155,107+Escy-y,">");
  outtextxy(x+Escx#227,107+Escy-y,">");
  outtextxy(x+Escx#440,129+Escy-y,"v");
  outtextxy(x+Escx#170,157+Escy-y,"v");
  outtextxy(x+Escx#210,145+Escy-y-lextheight("n")/2+1,"<");
  outtextxy(x+Escx#155,207+Escy-y,">");
  outtextxy(x+Escx#170,194+Escy-y,"v");
  outtextxy(x+Escx#170,108+Escy-y+12+Escx,"^");
  outtextxy(x+Escx#227,207+Escy-y,">");
  outtextxy(x+Escx#440,143+Escy-y+12+Escx,"^");
  outtextxy(x+Escx#327,107+Escy-y,">");
  outtextxy(x+Escx#253,107+Escy-y,"FPB1");
  outtextxy(x+Escx#170,142+Escy-y,"VCO");
  outtextxy(x+Escx#255,207+Escy-y,"FBP2");
  outtextxy(x+Escx#355,107+Escy-y,"ADC");
  outtextxy(x+Escx#170,107+Escy-y,"X");
  outtextxy(x+Escx#440,142+Escy-y,"X");
  outtextxy(x+Escx#170,207+Escy-y,"X");
  outtextxy(x+Escx#170,177+Escy-y,"90");
  selltextstyle(SMALL_FONT,HORIZ_DIR,4);
  outtextxy(x+Escx#70,145+Escy-y,"[1]");
  outtextxy(x+Escx#140,118+Escy-y,"[2]");
  outtextxy(x+Escx#210,192+Escy-y,"[6]");
  outtextxy(x+Escx#380,192+Escy-y,"[9]");
  outtextxy(x+Escx#310,95+Escy-y,"[4]");
  outtextxy(x+Escx#213,95+Escy-y,"[3]");
  outtextxy(x+Escx#142,192+Escy-y,"[8]");
  outtextxy(x+Escx#415,95+Escy-y,"[5]");
  outtextxy(x+Escx#310,131+Escy-y,"[7]");
  setlinestyle(0,0,NORM_WIDTH);
  selltextjustify( LEFT_TEXT, TOP_TEXT );
}

```

```

settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}
/*FORMABUC6: Presenta el bucle de casillas NUMERICO
* Parámetros : x.- coordenada X base
*              y.- coordenada Y base
* Retorno : Ninguno
#####*/
void formaBuch6(float x,float y)
{
settextjustify(CENTER_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
barra(x+Escx*230,100fEscy-y,x+Escx*290,120fEscy-y);
barra(x+Escx*330,100fEscy-y,x+Escx*390,120fEscy-y);
barra(x+Escx*145,135fEscy-y,x+Escx*195,155fEscy-y);
barra(x+Escx*155,170fEscy-y,x+Escx*185,190fEscy-y);
barra(x+Escx*230,200fEscy-y,x+Escx*280,220fEscy-y);
barra(x+Escx*330,200fEscy-y,x+Escx*390,220fEscy-y);
barra(x+Escx*352,135fEscy-y,x+Escx*402,155fEscy-y);
circulo(x+Escx*170,110fEscy-y,12fEscx);
circulo(x+Escx*170,210fEscy-y,12fEscx);
setlinewidth(0,0,THICK_WIDTH);
line(x+Escx*90,110fEscy-y,155fEscx,x,110fEscy-y);
line(x+Escx*182,110fEscy-y,x+Escx*227,110fEscy-y);
line(x+Escx*285,110fEscy-y,x+Escx*327,110fEscy-y);
line(x+Escx*385,110fEscy-y,x+Escx*440,110fEscy-y);
line(x+Escx*440,110fEscy-y,x+Escx*440,140fEscy-y);
line(x+Escx*440,140fEscy-y,x+Escx*407,140fEscy-y);
line(x+Escx*407,150fEscy-y,x+Escx*440,150fEscy-y);
line(x+Escx*352,145fEscy-y,x+Escx*200,145fEscy-y);
line(x+Escx*170,155fEscy-y,x+Escx*170,167fEscy-y);
line(x+Escx*170,190fEscy-y,x+Escx*170,212fEscy-y-12fEscx);
line(x+Escx*170,108fEscy-y+12fEscx,x+Escx*170,132fEscy-y);
line(x+Escx*90,210fEscy-y,x+Escx*155,210fEscy-y);
line(x+Escx*90,110fEscy-y,x+Escx*90,210fEscy-y);
line(x+Escx*170,160fEscy-y,x+Escx*90,160fEscy-y);
line(x+Escx*181,210fEscy-y,x+Escx*227,210fEscy-y);
line(x+Escx*285,210fEscy-y,x+Escx*327,210fEscy-y);
line(x+Escx*385,210fEscy-y,x+Escx*440,210fEscy-y);
line(x+Escx*440,150fEscy-y,x+Escx*440,210fEscy-y);
outtextxy(x+Escx*155,107fEscy-y,">");
outtextxy(x+Escx*227,107fEscy-y,">");
outtextxy(x+Escx*327,107fEscy-y,">");
outtextxy(x+Escx*209,145fEscy-y-12fEscx,"N")+1,"<");
outtextxy(x+Escx*416,137fEscy-y,"<");
outtextxy(x+Escx*416,147fEscy-y,"<");
outtextxy(x+Escx*155,207fEscy-y,">");
outtextxy(x+Escx*227,207fEscy-y,">");
outtextxy(x+Escx*327,207fEscy-y,">");
outtextxy(x+Escx*170,108fEscy-y+12fEscx,"^");
outtextxy(x+Escx*170,157fEscy-y,"v");
outtextxy(x+Escx*255,107fEscy-y,"FB1");
outtextxy(x+Escx*170,142fEscy-y,"VCO");
outtextxy(x+Escx*255,207fEscy-y,"FBP2");
outtextxy(x+Escx*355,107fEscy-y,"ADC");
outtextxy(x+Escx*355,207fEscy-y,"ADC");
outtextxy(x+Escx*377,142fEscy-y,"ANO");
outtextxy(x+Escx*170,194fEscy-y,"v");
outtextxy(x+Escx*170,107fEscy-y,"X");
outtextxy(x+Escx*170,207fEscy-y,"X");
outtextxy(x+Escx*170,177fEscy-y,"90");
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx*72,145fEscy-y,"[1]");

```



```

outtextxy(x+Escx#147,115#Escy-y,"[2]");
outtextxy(x+Escx#212,195#Escy-y,"[6]");
outtextxy(x+Escx#310,195#Escy-y,"[9]");
outtextxy(x+Escx#310,95#Escy-y,"[4]");
outtextxy(x+Escx#210,95#Escy-y,"[3]");
outtextxy(x+Escx#143,192#Escy-y,"[8]");
outtextxy(x+Escx#415,95#Escy-y,"[5]");
outtextxy(x+Escx#310,130#Escy-y,"[7]");
outtextxy(x+Escx#415,195#Escy-y,"[0]");
setlinesyle(0,0,HORR_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

```

/##FORMABUCI: Presenta el bucle CON MULTIPLICACION

```

```

# Parametros : x .- coordenada X base
#              y .- coordenada Y base
# Retorno : Ninguno
#####
void formabucl(float x, float y)
{
settextjustify( CENTER_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
barra(x+Escx#350,90#Escy-y,x+Escx#400,110#Escy-y);
barra(x+Escx#450,90#Escy-y,x+Escx#500,110#Escy-y);
barra(x+Escx#230,122#Escy-y,x+Escx#280,142#Escy-y);
barra(x+Escx#230,154#Escy-y,x+Escx#280,174#Escy-y);
barra(x+Escx#375,154#Escy-y,x+Escx#425,174#Escy-y);
barra(x+Escx#230,190#Escy-y,x+Escx#280,210#Escy-y);
barra(x+Escx#130,190#Escy-y,x+Escx#180,210#Escy-y);
circulo(x+Escx#255,100#Escy-y,12#Escx);
setlinesyle(0,1,THICK_WIDTH);
line(x+Escx#95,100#Escy-y,x+Escx#240,100#Escy-y);
line(x+Escx#267,100#Escy-y,x+Escx#347,100#Escy-y);
line(x+Escx#405,100#Escy-y,x+Escx#447,100#Escy-y);
line(x+Escx#505,100#Escy-y,x+Escx#550,100#Escy-y);
line(x+Escx#110,200#Escy-y,x+Escx#127,200#Escy-y);
line(x+Escx#185,200#Escy-y,x+Escx#227,200#Escy-y);
line(x+Escx#285,200#Escy-y,x+Escx#400,200#Escy-y);
line(x+Escx#110,100#Escy-y,x+Escx#110,200#Escy-y);
line(x+Escx#255,98#Escy-y+12#Escx,x+Escx#255,118#Escy-y);
line(x+Escx#255,142#Escy-y,x+Escx#255,150#Escy-y);
line(x+Escx#255,174#Escy-y,x+Escx#255,186#Escy-y);
line(x+Escx#285,132#Escy-y,x+Escx#400,132#Escy-y);
line(x+Escx#400,132#Escy-y,x+Escx#400,150#Escy-y);
line(x+Escx#400,176#Escy-y,x+Escx#400,200#Escy-y);
outtextxy(x+Escx#400,174#Escy-y,"^");
outtextxy(x+Escx#240,97#Escy-y,">");
outtextxy(x+Escx#347,97#Escy-y,">");
outtextxy(x+Escx#550,97#Escy-y,">");
outtextxy(x+Escx#127,197#Escy-y,">");
outtextxy(x+Escx#447,97#Escy-y,">");
outtextxy(x+Escx#227,197#Escy-y,">");
outtextxy(x+Escx#255,177#Escy-y,"v");
outtextxy(x+Escx#255,98#Escy-y+12#Escx,"^");
outtextxy(x+Escx#295,132#Escy-y-tertheight("H")/2+1,"<");
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx#130,85#Escy-y,"[1]");
outtextxy(x+Escx#304,85#Escy-y,"[3]");
outtextxy(x+Escx#428,85#Escy-y,"[4]");
outtextxy(x+Escx#527,85#Escy-y,"[5]");
outtextxy(x+Escx#320,186#Escy-y,"[6]");

```

```

outtextxy(x+Escr#210,186+Escy-y,"[7]");
outtextxy(x+Escr#330,118+Escy-y,"[9]");
outtextxy(x+Escr#227,105+Escy-y,"[2]");
outtextxy(x+Escr#227,175+Escy-y,"[R]");
outtextxy(x+Escr#375,97+Escy-y,"[P2]");
outtextxy(x+Escr#400,161+Escy-y,"[FR1]");
outtextxy(x+Escr#255,129+Escy-y,"VCO");
outtextxy(x+Escr#255,161+Escy-y,"X2");
outtextxy(x+Escr#255,98+Escy-y,"X");
outtextxy(x+Escr#475,97+Escy-y,"ADC");
outtextxy(x+Escr#255,197+Escy-y,"FD");
outtextxy(x+Escr#155,198+Escy-y,"X2");
setlinestyle(0,0,HDRH_WIDTH);
settextjustify(LEFT_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

\*/#FORMABUC3: Presenta el bucle CON REMODULACION POR EL VCO

\* Parámetros : x .- coordenada X base

\* y .- coordenada Y base

\* Retorno : Ninguno

#####

void formabuc3(float x,float y)

```

{
settextjustify(CENTER_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
barra(x+Escr#230,110+Escy-y,x+Escr#280,130+Escy-y);
barra(x+Escr#330,110+Escy-y,x+Escr#300,130+Escy-y);
barra(x+Escr#145,160+Escy-y,x+Escr#175,180+Escy-y);
barra(x+Escr#155,200+Escy-y,x+Escr#185,220+Escy-y);
circulo(x+Escr#170,120+Escy-y,12+Escr);
circulo(x+Escr#440,120+Escy-y,12+Escr);
setlinestyle(0,1,THICK_WIDTH);
line(x+Escr#80,120+Escy-y,x+Escr#155,120+Escy-y);
line(x+Escr#182,120+Escy-y,x+Escr#227,120+Escy-y);
line(x+Escr#285,120+Escy-y,x+Escr#327,120+Escy-y);
line(x+Escr#385,120+Escy-y,x+Escr#425,120+Escy-y);
line(x+Escr#452,120+Escy-y,x+Escr#500,120+Escy-y);
line(x+Escr#170,120+Escy-y,x+Escr#170,210+Escy-y);
line(x+Escr#170,210+Escy-y,x+Escr#152,210+Escy-y);
line(x+Escr#170,118+Escy-y+12+Escr,x+Escr#170,155+Escy-y);
line(x+Escr#170,195+Escy-y,x+Escr#170,182+Escy-y);
line(x+Escr#170,210+Escy-y,x+Escr#500,210+Escy-y);
line(x+Escr#500,120+Escy-y,x+Escr#500,210+Escy-y);
line(x+Escr#200,170+Escy-y,x+Escr#440,170+Escy-y);
line(x+Escr#440,170+Escy-y,x+Escr#440,118+Escy-y+12+Escr);
outtextxy(x+Escr#255,117+Escy-y,"FPB1");
outtextxy(x+Escr#170,167+Escy-y,"VCO");
outtextxy(x+Escr#170,207+Escy-y,"FD");
outtextxy(x+Escr#355,117+Escy-y,"ADC");
outtextxy(x+Escr#170,117+Escy-y,"X");
outtextxy(x+Escr#440,117+Escy-y,"X");
outtextxy(x+Escr#155,117+Escy-y,">");
outtextxy(x+Escr#227,117+Escy-y,">");
outtextxy(x+Escr#327,117+Escy-y,">");
outtextxy(x+Escr#425,117+Escy-y,">");
outtextxy(x+Escr#152,207+Escy-y,">");
outtextxy(x+Escr#201,207+Escy-y,"<");
outtextxy(x+Escr#170,180+Escy-y,"^");
outtextxy(x+Escr#170,118+Escy-y+12+Escr,"^");
outtextxy(x+Escr#440,118+Escy-y+12+Escr,"^");
settextstyle(SMALL_FONT,HORIZ_DIR,4);
}

```

```

outtextxy(x+Escx#110,105#Escy-y,"[1]");
outtextxy(x+Escx#195,135#Escy-y,"[2]");
outtextxy(x+Escx#205,192#Escy-y,"[6]");
outtextxy(x+Escx#310,105#Escy-y,"[4]");
outtextxy(x+Escx#212,105#Escy-y,"[3]");
outtextxy(x+Escx#380,195#Escy-y,"[7]");
outtextxy(x+Escx#415,105#Escy-y,"[5]");
setlinestyle(0,0,HORN_WIDTH);
settextjustify(LEFT_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

```

/!FORNABUC4: Presenta el bucle CON REMODULACION POR LA ONDA RECIBIDA
! Parámetros : x .- coordenada X base
!              y .- coordenada Y base
! Retorno : Ninguno
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!/
void fornabuc4(float x,float y)
{
settextjustify(CENTER_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
barra(x+Escx#230,110#Escy-y,x+Escx#280,130#Escy-y);
barra(x+Escx#330,110#Escy-y,x+Escx#380,130#Escy-y);
barra(x+Escx#145,160#Escy-y,x+Escx#195,180#Escy-y);
barra(x+Escx#300,160#Escy-y,x+Escx#350,180#Escy-y);
circulo(x+Escx#170,120#Escy-y,12#Escx);
circulo(x+Escx#440,120#Escy-y,12#Escx);
setlinestyle(0,1,THICK_WIDTH);
line(x+Escx#78,120#Escy-y,x+Escx#155,120#Escy-y);
line(x+Escx#182,120#Escy-y,x+Escx#227,120#Escy-y);
line(x+Escx#285,120#Escy-y,x+Escx#327,120#Escy-y);
line(x+Escx#385,120#Escy-y,x+Escx#425,120#Escy-y);
line(x+Escx#452,120#Escy-y,x+Escx#498,120#Escy-y);
line(x+Escx#170,118#Escy-y+12#Escx,x+Escx#170,156#Escy-y);
line(x+Escx#170,210#Escy-y,x+Escx#170,190#Escy-y);
line(x+Escx#80,120#Escy-y,x+Escx#80,85#Escy-y);
line(x+Escx#80,85#Escy-y,x+Escx#440,85#Escy-y);
line(x+Escx#440,85#Escy-y,x+Escx#440,122#Escy-y-12#Escx);
line(x+Escx#200,170#Escy-y,x+Escx#297,170#Escy-y);
line(x+Escx#355,170#Escy-y,x+Escx#500,170#Escy-y);
line(x+Escx#170,210#Escy-y,x+Escx#325,210#Escy-y);
line(x+Escx#325,210#Escy-y,x+Escx#325,180#Escy-y);
line(x+Escx#80,120#Escy-y,x+Escx#170,120#Escy-y);
line(x+Escx#500,120#Escy-y,x+Escx#500,170#Escy-y);
outtextxy(x+Escx#440,104#Escy-y,"v");
outtextxy(x+Escx#364,167#Escy-y,"<");
outtextxy(x+Escx#297,167#Escy-y,">");
outtextxy(x+Escx#170,180#Escy-y,"^");
outtextxy(x+Escx#170,118#Escy-y+12#Escx,"^^");
outtextxy(x+Escx#227,117#Escy-y,">");
outtextxy(x+Escx#327,117#Escy-y,">");
outtextxy(x+Escx#425,117#Escy-y,">");
outtextxy(x+Escx#155,117#Escy-y,">");
outtextxy(x+Escx#255,117#Escy-y,"FPB1");
outtextxy(x+Escx#170,167#Escy-y,"VCO");
outtextxy(x+Escx#325,167#Escy-y,"PD");
outtextxy(x+Escx#355,117#Escy-y,"ABC");
outtextxy(x+Escx#170,117#Escy-y,"X");
outtextxy(x+Escx#440,117#Escy-y,"X");
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx#62,105#Escy-y,"[1]");
outtextxy(x+Escx#195,130#Escy-y,"[2]");

```

```

outtextxy(x+Escx#245,195#Escy-y,"[6]");
outtextxy(x+Escx#310,105#Escy-y,"[4]");
outtextxy(x+Escx#213,105#Escy-y,"[3]");
outtextxy(x+Escx#460,150#Escy-y,"[7]");
outtextxy(x+Escx#415,105#Escy-y,"[5]");
setlinesstyle(0,0,HORN_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

/*FORMA: Presenta el diagrama de bloques del PLL
# Parámetros : x.- coordenada X base
#               y.- coordenada Y base
# Retorno : Ninguno
#####/
void forma(float x,float y)
{
settextjustify( CENTER_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(7);
barra(x+Escx#200,-y+Escy#100,x+Escx#300,-y+Escy#120);
barra(x+Escx#400,-y+Escy#140,x+Escx#500,-y+Escy#160);
barra(x+Escx#200,-y+Escy#200,x+Escx#300,-y+Escy#220);
setlinesstyle(0,1,THICK_WIDTH);
setcolor(10);
line(x+Escx#170,-y+Escy#110,x+Escx#198,-y+Escy#110);
line(x+Escx#305,-y+Escy#110,x+Escx#450,-y+Escy#110);
line(x+Escx#450,-y+Escy#110,x+Escx#450,-y+Escy#137);
line(x+Escx#450,-y+Escy#161,x+Escx#450,-y+Escy#210);
line(x+Escx#450,-y+Escy#210,x+Escx#305,-y+Escy#210);
line(x+Escx#250,-y+Escy#121,x+Escx#250,-y+Escy#197);
outtextxy(x+Escx#250,-y+Escy#122,"^");
outtextxy(x+Escx#197,-y+Escy#107,">");
outtextxy(x+Escx#314,-y+Escy#207,"<");
outtextxy(x+Escx#450,-y+Escy#126,"v");
setcolor(12);
outtextxy(x+Escx#250,-y+Escy#107,"P D");
outtextxy(x+Escx#450,-y+Escy#147,"F P B");
outtextxy(x+Escx#250,-y+Escy#207,"V C D");
setlinesstyle(0,0,HORN_WIDTH);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx#165,-y+Escy#95,"[1]");
outtextxy(x+Escx#350,-y+Escy#95,"[6]");
outtextxy(x+Escx#415,-y+Escy#175,"[7]");
outtextxy(x+Escx#220,-y+Escy#145,"[2]");
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(15);
}

/*FORMABUC7: Presenta el bucle de costas para QPSK
# Parámetros : x.- coordenada X base
#               y.- coordenada Y base
# Retorno : Ninguno
#####/
void formabuc7(float x,float y)
{
settextjustify( CENTER_TEXT, TOP_TEXT );
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
barra(x+Escx#230,95#Escy-y,x+Escx#280,115#Escy-y);
barra(x+Escx#330,95#Escy-y,x+Escx#380,115#Escy-y);
barra(x+Escx#145,140#Escy-y,x+Escx#195,160#Escy-y);

```

```

barra(x+Escx#155,170#Escy-y,x+Escx#185,170#Escy-y);
barra(x+Escx#230,200#Escy-y,x+Escx#280,220#Escy-y);
barra(x+Escx#330,200#Escy-y,x+Escx#380,220#Escy-y);
barra(x+Escx#460,140#Escy-y,x+Escx#500,140#Escy-y);
circulo(x+Escx#170,105#Escy-y,12#Escx);
circulo(x+Escx#170,210#Escy-y,12#Escx);
circulo(x+Escx#440,125#Escy-y,12#Escx);
circulo(x+Escx#440,190#Escy-y,12#Escx);
sellineslyle(0,1,THICK_WIDTH);
line(x+Escx#190,105#Escy-y,x+Escx#190,210#Escy-y);
line(x+Escx#190,160#Escy-y,x+Escx#170,160#Escy-y);
line(x+Escx#190,210#Escy-y,x+Escx#155,210#Escy-y);
line(x+Escx#170,190#Escy-y,x+Escx#170,212#Escy-y-12#Escx);
line(x+Escx#190,105#Escy-y,x+Escx#155,105#Escy-y);
line(x+Escx#182,105#Escy-y,x+Escx#227,105#Escy-y);
line(x+Escx#285,105#Escy-y,x+Escx#327,105#Escy-y);
line(x+Escx#385,105#Escy-y,x+Escx#440,105#Escy-y);
line(x+Escx#440,105#Escy-y,x+Escx#440,115#Escy-y);
line(x+Escx#480,125#Escy-y,x+Escx#480,137#Escy-y);
line(x+Escx#480,160#Escy-y,x+Escx#480,188#Escy-y);
line(x+Escx#480,125#Escy-y,x+Escx#452,125#Escy-y);
line(x+Escx#452,190#Escy-y,x+Escx#480,190#Escy-y);
line(x+Escx#440,188#Escy-y+12#Escx,x+Escx#440,210#Escy-y);
line(x+Escx#440,210#Escy-y,x+Escx#385,210#Escy-y);
line(x+Escx#327,210#Escy-y,x+Escx#285,210#Escy-y);
line(x+Escx#227,210#Escy-y,x+Escx#182,210#Escy-y);
line(x+Escx#305,208#Escy-y,x+Escx#305,190#Escy-y);
line(x+Escx#305,140#Escy-y,x+Escx#440,140#Escy-y);
line(x+Escx#460,150#Escy-y,x+Escx#198,150#Escy-y);
line(x+Escx#440,180#Escy-y,x+Escx#440,165#Escy-y);
line(x+Escx#440,123#Escy-y+12#Escx,x+Escx#440,140#Escy-y);
line(x+Escx#170,160#Escy-y,x+Escx#170,166#Escy-y);
line(x+Escx#440,165#Escy-y,x+Escx#300,165#Escy-y);
line(x+Escx#300,165#Escy-y,x+Escx#300,105#Escy-y);
line(x+Escx#170,114#Escy-y,x+Escx#170,136#Escy-y);
outtextxy(x+Escx#170,103#Escy-y+12#Escx,"^");
outtextxy(x+Escx#170,173#Escy-y,"v");
outtextxy(x+Escx#480,160#Escy-y,"^");
outtextxy(x+Escx#480,126#Escy-y,"v");
outtextxy(x+Escx#155,105#Escy-y-lextheight("H")/2+1,">");
outtextxy(x+Escx#227,105#Escy-y-lextheight("H")/2+1,">");
outtextxy(x+Escx#327,105#Escy-y-lextheight("H")/2+1,">");
outtextxy(x+Escx#440,109#Escy-y,"v");
outtextxy(x+Escx#440,123#Escy-y+12#Escx,"^");
outtextxy(x+Escx#440,188#Escy-y+12#Escx,"^");
outtextxy(x+Escx#440,173#Escy-y,"v");
outtextxy(x+Escx#327,207#Escy-y,">");
outtextxy(x+Escx#227,207#Escy-y,">");
outtextxy(x+Escx#155,207#Escy-y,">");
outtextxy(x+Escx#209,147#Escy-y,"<");
outtextxy(x+Escx#255,102#Escy-y,"FBI");
outtextxy(x+Escx#170,147#Escy-y,"VCD");
outtextxy(x+Escx#255,207#Escy-y,"FBP2");
outtextxy(x+Escx#355,102#Escy-y,"ADC");
outtextxy(x+Escx#355,207#Escy-y,"ADC");
outtextxy(x+Escx#170,101#Escy-y,"X");
outtextxy(x+Escx#170,207#Escy-y,"X");
outtextxy(x+Escx#440,187#Escy-y,"X");
outtextxy(x+Escx#440,122#Escy-y,"X");
outtextxy(x+Escx#170,177#Escy-y,"90");
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
outtextxy(x+Escx#480,139#Escy-y,"+");

```

```

settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x+Escx*175,145*Escy-y,"[1]");
outtextxy(x+Escx*195,115*Escy-y,"[2]");
outtextxy(x+Escx*212,195*Escy-y,"[6]");
outtextxy(x+Escx*315,195*Escy-y,"[9]");
outtextxy(x+Escx*310,90*Escy-y,"[4]");
outtextxy(x+Escx*210,90*Escy-y,"[3]");
outtextxy(x+Escx*145,172*Escy-y,"[8]");
outtextxy(x+Escx*415,90*Escy-y,"[5]");
outtextxy(x+Escx*270,135*Escy-y,"[7]");
outtextxy(x+Escx*500,113*Escy-y,"[b]");
outtextxy(x+Escx*500,170*Escy-y,"[a]");
outtextxy(x+Escx*495,123*Escy-y,"+");
outtextxy(x+Escx*495,160*Escy-y,"-");
outtextxy(x+Escx*410,195*Escy-y,"[0]");
setlinestyle(0,0,NORM_WIDTH);
settextjustify(LEFT_TEXT, TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

```
/*
```

```

Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
ENAYUDA.C

```

```

Archivo que despliega en pantalla ayudas si se las ha solicitado
desde ENTRADA.EXE

```

```
*/
```

```

#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include "proyfil.h"
#include <graphics.h>
#include <alloc.h>
#include <stdio.h>
extern float MaxX,MaxY,Escx,Escy;
char far *buf,*buf_imagen;
extern int nbucle;
void punto(int x,int y);
void resistencia(int x,int y,int esp1,int esp2,int hv);
void capacitor(int x,int y,int esp1,int esp2,int hv);
void amplificador(int x,int y,int esp1,int esp2);
void filtro_el(int x,int y,int esp1,int esp2);
void filtro_b2(int x,int y,int esp1,int esp2);
void escribir(int x,int y,int esp2,int tipo,int i);
void borrar_pagina(int dix1,int diy1,int dix2,int diy2);
/****
*AYUDA:Despliega en pantalla ayuda de acuerdo al caso ( ACCESO POR EL USUARIO)
*Parámetros: dix1 .- coordenada X de la esquina superior izquierda del recuadro
*             diy1 .- coordenada Y de la esquina superior izquierda del recuadro
*             dix2 .- coordenada X de la esquina inferior derecha del recuadro
*             diy2 .- coordenada Y de la esquina inferior derecha del recuadro
*             caso .- indicador de que caso de ayuda se ha solicitado
*Retorno : Ninguno
*****/
void ayuda(int dix1,int diy1,int dix2,int diy2, int caso){
size_t la;
int x,y,memoria,dec,a,esp1;
if(caso == 7)return;
x=175*Escx;y=5*Escy;
setviewport(0,0,MaxX,MaxY,2);
la=imagesize(dix1-1,diy1-1,dix2+1,diy2+1);
memoria=coreleft();

```

```

buf=graphgetarea(la);
getimage(dix1-1,diy1-1,dix2+1,diy2+1,buf);
setviewport(dix1,diy1,dix2,diy2,1);
clearviewport();
settextjustify(LEFT_TEXT, TOP_TEXT);
setviewport(0,0,MaxX,MaxY,?);
if(caso<100)
    MainWindow(dix1,diy1,dix2,diy2,1,0," A y u d a ");
if(caso>99 && caso<200)
    MainWindow(dix1,diy1,dix2,diy2,1,0,"RESPUESTA DE FRECUENCIA DEL FILTRO");
if(caso>499)
    MainWindow(dix1,diy1,dix2,diy2,1,0," DIAGRAMA DE BLOQUES DEL BUCLE ");
setviewport(dix1+11*Escx,diy1+22*Escy,dix2-11*Escx,diy2-22*Escy,1);
clearviewport();
settextjustify(LEFT_TEXT, TOP_TEXT);
setcolor(15);
switch(caso){
case 1:
    x=10*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"                ESTRUCTURACION DEL FILTRO PASABAJOS ");
    gprintf(&x,&y,2,"                Para la simulación del filtro pasabajos se ha escogido dos");
    gprintf(&x,&y,2,"aproximaciones ealesalicas.");
    gprintf(&x,&y,2,"                FILTRO ELIPTICO.- Permiten obtener un corte extremadamente");
    gprintf(&x,&y,2,"abrupto y disponer de ceros de transación, es decir se puede");
    gprintf(&x,&y,2,"eliminar frecuencias no deseadas ya que a la frecuencia donde se");
    gprintf(&x,&y,2,"da el cero de transación la atenuación tiene un valor INFINITO.");
    gprintf(&x,&y,2,"Pero presenta deformaciones muy grandes en regíen transitorio.");
    gprintf(&x,&y,2,"                FILTRO DE BUTTERWORTH.- tienen un corte poco abrupto. Pero");
    gprintf(&x,&y,2,"presenta deformaaciones muy pequenas en regíen transitorio.");
    break;
case 6:
    x=10*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"                ORDEN DEL FILTRO PASABAJOS ");
    gprintf(&x,&y,2,"                Dado las FRECUENCIA CRITICAS (wp) y (wr) con sus respectivas");
    gprintf(&x,&y,2,"ATENUACIONES (Awp) y (Awr). Se han hecho los calculos necesarios");
    gprintf(&x,&y,2,"para determinar el ORDEN del filtro que se requiere para satis_");
    gprintf(&x,&y,2,"facer las condiciones de frecuencia y atenuación impuestas.");
    gprintf(&x,&y,2,"                Si el valor es menor al recomendado se acepta,pero el filtro");
    gprintf(&x,&y,2,"estructurado no tiene las características de frecuencia dadas.");
    gprintf(&x,&y,2,"                Su VALOR debe ser ENTERO y MENOR que 12.");
    gprintf(&x,&y,2,"Se puede estructurar un filtro de orden INFINITO, Para el pre_");
    gprintf(&x,&y,2,"sente trabajo es suficiente cumplir con la restricción impuesta.");
    break;
case 10:
    x=10*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"                ENTRADA DE BIT'S A SIMULAR                ");
    gprintf(&x,&y,2,"                Se solicita el ingreso del tren de bits ha ser modulos con ");
    gprintf(&x,&y,2,"la tecnica PSK(Phase Shift Key), que constiluye la senal de en_");
    gprintf(&x,&y,2,"trada al bucle que se desea analizar el comportamiento.                ");
    gprintf(&x,&y,2,"                El tren de bit's pertenecen a un sistema BINARIO,es decir,");
    gprintf(&x,&y,2,"ser acepta como dato valido UNICAMENTE un valor 0 y un valor 1,");
    gprintf(&x,&y,2,"que corresponden a los valores logicos 0 y 1 respectivamente.                ");
    gprintf(&x,&y,2,"                ");
    gprintf(&x,&y,2,"                Por visualización de resultados considerar :");
    gprintf(&x,&y,2,"Ciclos en pantalla = FREQ. de Portadora * # bits / RITMO                ");
    gprintf(&x,&y,2,"                Por tanto, ingresar NUMERO DE DATOS < 10.");
    break;
case 11:
    x=25*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"                ");
    gprintf(&x,&y,2,"                ENTRADA DEL RITMO DE TRANSMISION                ");
    gprintf(&x,&y,2,"                Se solicita el ingreso de la velocidad binaria del tren");

```

```

gprintf(&x,&y,2,"de bits que se ha ingresado anteriormente. ");
gprintf(&x,&y,2," El Ritmo se inliende como el numero de bits que se ');
gprintf(&x,&y,2,"ransmite ,recibe , procesa , etc por segundo. ");
gprintf(&x,&y,2," Por visualización de resultados, es recomendable que la ');
gprintf(&x,&y,2,"relación FRECUENCIA DE PORTADORA/RITMO sea 1,2,.. . ");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2,"#ciclos en pantalla = FREC. de Portadora * # bits / RITMO ");
break;
case 12:x=10*Escx;y=-2*Escy;
gprintf(&x,&y,2," ENTRADA DE LA FRECUENCIA DE PORTADORA');
gprintf(&x,&y,2," La frecuencia se inliende como el numero ciclos de una senal');
gprintf(&x,&y,2,"analoga que se transmite recibe, procesa, etc por segundo.");
gprintf(&x,&y,2," Con esta senal analoga se modula el tren de bits que tiene");
gprintf(&x,&y,2,"el RITMO ya ingresados anteriormente. ");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," #ciclos en pantalla = FREC. de Portadora * # bits / RITMO ");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Por visualización de resultados se recomienda que la relación");
gprintf(&x,&y,2,"FRECUENCIA DE PORTADORA/RITMO sea aceptable, asi:1, 1.5, 2 ,etc.");
break;
case 13:
x=10*Escx;y=-2*Escy;
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," ENTRADA DEL ERROR DE FASE INICIAL QUE TIENE EL BUCLE");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Se solicita el ingreso del angulo de error inicial que exis_");
gprintf(&x,&y,2,"te entre las senales modulada recibida por bucle y la del Osci_");
gprintf(&x,&y,2,"lador Controlado por Voltaje creada en el mismo.");
gprintf(&x,&y,2," Se necesita este dato para poder observar la operación de ");
gprintf(&x,&y,2,"sincronisao del BUCLE. ");
gprintf(&x,&y,2," El valor del dato debe estar entre 0 y 360 ");
break;
case 51:
x=10*Escx;y=-2*Escy;
gprintf(&x,&y,2," Frec. CENTRAL del OSCILADOR CONTROLADO por VOLTAJE.");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Se utiliza un VCO de respuesta lineal de frecuencia con");
gprintf(&x,&y,2,"respecto al voltaje de control (Vc). El voltaje de ERROR de fase");
gprintf(&x,&y,2,"(Ve) esta normalizado, asi: -1 < Ve < 1. ");
gprintf(&x,&y,2,"Vc = 1 + Ve , en consecuencia 0 < Vc < 2.");
gprintf(&x,&y,2," La frecuencia CENTRAL de oscilación debe ser la misma que");
gprintf(&x,&y,2,"la utilizada para modular la onda de entrada a los bucles, para");
gprintf(&x,&y,2,"que el BUCLE responda adecuadamente.");
gprintf(&x,&y,2," Su valor esta dado en [Hertz].");
break;
case 52:
x=10*Escx;y=-2*Escy;
gprintf(&x,&y,2," SENSIBILIDAD del OSCILADOR CONTROLADO por VOLTAJE.");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Es el cambio de frecuencia por la variación del error de");
gprintf(&x,&y,2,"fase en un grado.");
gprintf(&x,&y,2," Conocidas fo y S estan definidas las frecuencias INFERIOR ");
gprintf(&x,&y,2,"y SUPERIOR de oscilación.");
gprintf(&x,&y,2," EJEMPLO : frecuencia CENTRAL (fo) = 1800 [Hz]");
gprintf(&x,&y,2," ESTABILIDAD = 20 [Hz/grado]");
gprintf(&x,&y,2," Cuando Vc=2, la frec. de oscilación es 3600[Hz]");
gprintf(&x,&y,2," Cuando Vc=0 el VCO no oscila");
gprintf(&x,&y,2,"Es decir, F.INFERIOR = 0 [Hz] y F. SUPERIOR = 3600 [Hz]");
break;
case 53:
x=10*Escx;y=-2*Escy;
gprintf(&x,&y,2," TOLERANCIA del OSCILADOR CONTROLADO por VOLTAJE.");

```



```

gprintf(&x,&y,2," ");
gprintf(&x,&y,2," Su valor es la dado en porcentaje : 0 < TOLERANCIA <100 % ");
gprintf(&x,&y,2," ");
gprintf(&x,&y,2," El V.C.O. produce cambio de frecuencia en su salida uni_");
gprintf(&x,&y,2,"cambie si el voltaje de ERROR de fase (Ve),supera el valor que");
gprintf(&x,&y,2,"se ingresa en porcentaje.");
gprintf(&x,&y,2," El ERROR maximo (100 %) se da cuando la diferencia de ");
gprintf(&x,&y,2,"fase es de 90 grados. Ya que (Ve) es proporcional a la función ");
gprintf(&x,&y,2,"trigonometrica seno del ERROR de IASF.");
break;
case 100:respuesta(25*Escx,-190*Escy,1);break;
case 101:respuesta(25*Escx,-190*Escy,2);break;
case 102:respuesta(25*Escx,-190*Escy,3);break;
case 103:respuesta(25*Escx,-190*Escy,4);break;
case 104:respuesta_banda(25*Escx,-190*Escy,1);break;
case 105:respuesta_banda(25*Escx,-190*Escy,2);break;
case 90:
a1:  borrar_pagina( dix1,diy1,dix2,diy2);
      x=60*Escx;y=-9*Escy;
      gprintf(&x,&y,2,"  ELEMENTOS BASICOS DEL FILTRO PASABAJOS ");
      x=70*Escx;y=-170*Escy;
      gprintf(&x,&y,2,"ELEMENTOS 1                                ELEMENTO 2");
      filtro_el(Escx*35,50*Escy,Escx*8,Escy*4);
      filtro_b2(Escx*36,58*Escy,Escx*8,Escy*4);
      MainWindow(dix1,diy1,dix2,diy2,1,1," ");
      do{
          dec=lecla();
      }while(dec!=27 && dec!=337 );
      if(dec==27)break;
case 91:
      borrar_pagina( dix1,diy1,dix2,diy2);
      x=10*Escx;y=-2*Escy;
      gprintf(&x,&y,2,"                                FILTRO PASABAJOS ");
      gprintf(&x,&y,2," Un filtro pasabajos de cualquier orden puede ser construido,");
      gprintf(&x,&y,2,"poniendo los elementos presentados en cascada. ");
      gprintf(&x,&y,2," Para el elemento 2,se recomienda R0 = 10.000 Ohmios, pues,se");
      gprintf(&x,&y,2,"precisa que R0 sea mucho menor que la impedancia de entrada del");
      gprintf(&x,&y,2,"amplificador y mucho mas elevada que su impedancia de salida. De");
      gprintf(&x,&y,2,"esto modo el amplificador funciona en buenas condiciones.");
      gprintf(&x,&y,2," Las resistencias R y r estan dados por el valor de k,=o asue");
      gprintf(&x,&y,2,"R=10.000 Ohmios con el mismo criterio que R0. r generalmente es ");
      gprintf(&x,&y,2,"un potenciómetro de ajuste de ganancia.");
      MainWindow(dix1,diy1,dix2,diy2,1,3," A Y U D O ");
      do{
          dec=lecla();
      }while(dec!=27 && dec!=329);
      if(dec==27)break;
      if(dec==329)goto a1;
}
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
if(caso != 90){
    MainWindow(dix1,diy1,dix2,diy2,1,4," ");
    getch();
}
clearviewport();
setviewport(0,0,MaxX,MaxY,1);
putimage((dix1-1),(diy1-1),buf,COPY_PUT);
_graphfreezen (buf,ta);
setfillstyle(SOLID_FILL, getmaxcolor());
setcolor (getmaxcolor());
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);

```

```

}

/*AYUDA_LINEA:Despliega en pantalla ayuda de acuerdo al caso ( SIN ACCESO AL USUARIO)
#Parámetros: num .- cantidad de caracteres a desplegarse
# pos .- posición del mensaje
# caso .- indicador de que caso de ayuda se ha solicitado
#Retorno : Ninguno
#####/
void ayuda_linea(int num,int caso,int pos){
int x,y,ail,espxl,prin;
float a;
    if(pos==1)
        a=10*Escy;
    else
        a=0;
    ail=(MaxX-24*Escx)/2;espxl=textwidth("M");
    x=15*Escx;y=MaxY-30*Escy;
    setviewport(0,0,MaxX,MaxY,1);
    setviewport(5*Escx,MaxY-39*Escy,MaxX-30*Escx,MaxY,1);
    clearviewport();
    setfillstyle(SOLID_FILL,0);
    MainWindow_linea(ail-(num+2)*espxl/2,MaxY-28*Escy-a,ail+(num+2)*espxl/2,MaxY-a,1);
    settextjustify(LEFT_TEXT, TOP_TEXT);
    setcolor(15);
    switch(caso){
case 1:  x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1,"Para la simulación del filtro pasabajos se ha escogido dos aproximaciones");
          gprintf(&x,&y,1,"matematicas: ELIPTICO y BUTTERWORTH ");
          break;
case 2:  x=10*Escx;y=12*Escy;
          gprintf(&x,&y,1," INGRESAR : Frecuencia de CORTE (fp)  0<fp<INFINITO  [Hz]");
          break;
case 3:  x=10*Escx;y=11*Escy;
          gprintf(&x,&y,1," INGRESAR : Frecuencia de SUPRESION  (fr)  fp<fr<INFINITO  [Hz]");
          break;
case 4:  x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1," INGRESAR : ATENUACION en la FRECUENCIA de CORTE  (Afp).");
          gprintf(&x,&y,1," Su valor : 0 < Afp < 20  dado en [dB] ");
          break;
case 5:  x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1," INGRESAR : ATENUACION en la FRECUENCIA de SUPRESION (Afr).");
          gprintf(&x,&y,1," Su valor : Afp < Afr < 100  dado en [dB] ");
          break;
case 6:  x=10*Escx;y=11*Escy;
          gprintf(&x,&y,1," INGRESAR : ORDEN del FILTRO  (N)  VALOR RECOMENDADO < N < 11.");
          break;
case 10: x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1," INGRESAR : TREN de BITS a ser modulados. Su valor : 0 o 1");
          gprintf(&x,&y,1," No Ingresar mas de 10 números por visualización de resultados");
          break;
case 11: x=10*Escx;y=11*Escy;
          gprintf(&x,&y,1," INGRESAR : RITMO de TRANSMISION  (R).  0<R<INFINITO  [bits/segundo]");
          break;
case 12: x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1," INGRESAR : FRECUENCIA de PORTADORA (fw)  0<fw<INFINITO  [Hz]");
          gprintf(&x,&y,1," Se recomienda que fw sea igual o multiplo entero de R");
          break;
case 13: x=10*Escx;y=7*Escy;
          gprintf(&x,&y,1," INGRESAR : ERROR INICIAL de FASE (e) que existe entre la onda modulada");
          gprintf(&x,&y,1," recibida y la creada por el VCO. Su valor : 0<e<360  [grados]");
          break;
case 32: x=10*Escx;y=7*Escy;

```

```

gprintf(&x,&y,1," INGRESAR : Frecuencia INFERIOR del FILTRO PASABANDA (f1).");
gprintf(&x,&y,1," Su valor : 0 < f1 < INFINITO [Hz]");
break;
case 33: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : Frecuencia SUPERIOR del FILTRO PASABANDA (f2).");
gprintf(&x,&y,1," Su valor : f1 < f2 < INFINITO [Hz]");
break;
case 34: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : ERROR de FASE MAXIMO que produce el DEFASADOR (EPS).");
gprintf(&x,&y,1," Su valor : 0 < EPS < 45 [grados]");
break;
case 35: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : ORDEN del FILTRO (N) VALOR RECOMENDADO < N < 6.");
break;
case 42: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : FRECUENCIA CENTRAL (fo) 0<fo<INFINITO [Hz]");
gprintf(&x,&y,1," Se recomienda fo=2fw. Donde fw es la frecuencia de portadora");
break;
case 43: x=10*Escx;y=11*Escy;
gprintf(&x,&y,1," INGRESAR : Frecuencia de CORTE inferior (fp) 0 < fp < fo [Hz]");
break;
case 44: x=10*Escx;y=11*Escy;
gprintf(&x,&y,1," INGRESAR : Frecuencia de SUPRESION inferior (fr) 0 < fr < fp [Hz]");
break;
case 45: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : ATENUACION en la FRECUENCIA de CORTE (Afp).");
gprintf(&x,&y,1," Su valor : 0 < Afp < 20 dado en [dB] ");
break;
case 46: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : ATENUACION en la FRECUENCIA de SUPRESION (Afr).");
gprintf(&x,&y,1," Su valor : Afr < Afr < 100 dado en [dB] ");
break;
case 47: x=10*Escx;y=11*Escy;
gprintf(&x,&y,1," INGRESAR : ORDEN del FILTRO (N) VALOR RECOMENDADO < N < 11.");
break;
case 51: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR: FRECUENCIA CENTRAL de OSCILACION (fo). 0<fo<INFINITO [Hz]");
gprintf(&x,&y,1," Se recomienda fo = fw. Donde fw es la frecuencia de portadora");
break;
case 52: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : SENSIBILIDAD (S) 0<S< INFINITO [Hz/grado]");
gprintf(&x,&y,1," SENSIBILIDAD : Cuanto cambia la frecuencia por GRADO de error de fase.");
break;
case 53: x=10*Escx;y=7*Escy;
gprintf(&x,&y,1," INGRESAR : TOLERANCIA 0 < TOLERANCIA < 100 [%]");
gprintf(&x,&y,1," TOLERANCIA: Desde que valor de error de fase cambia la frecuencia");
break;
}
setTextStyle(DEFAULT_FONT,HORIZ_DIR,1);
setviewport(0,0,MaxX,MaxY,1);
if(pos==1){
rectangle(ceil((num+2)*espx/2-12*textwidth("N"),MaxY-textheight("N")-4,ceil((num+2)*espx/2,MaxY);
outtextxy(ceil((num+2)*espx/2-11*textwidth("N"),MaxY-textheight("N")/2-4,"[F1] AYUDA");
}
}

/*RESPUESTA_ACOMODADA:Respuesta de frecuencia prototipo del filtro pasabajos ELIPTICO
#Parámetros : x.- coordenada X de la esquina superior izquierda de la figura
# y.- coordenada Y de la esquina superior izquierda de la figura
#Retorno : Ninguno
#####/
void respuesta_acomodada(int x,int y)

```

```

(
float i,A01,R01,R11,S0,H0,H,R[5],R[5],FR,X1,X2,Y1,Y2,Cf,Vf,ii;
A01=1.9690915486;R01=-.6513074073;R11=-.309851248077;
H0=.1531423400042;S0=-.46270213;O[1]=(2*A01-8)/(A01+4);
O[2]=1;R[3]=(S0-2)/(S0+2);O[4]=0;O[3]=1;O[4]=0;
R[1]=(2*R01-8)/(4+2*R11+R01);
R[2]=(4-2*R11+R01)/(4+2*R11+R01);
KR=H0*(4+A01)/(4+2*R11+R01)/(S0+2);
setlinesyle(0,0,HORN_WIDTH);
line(x+Escx*160,y+Escy*180,x+Escx*160,y+Escy*292);
line(x+Escx*150,y+Escy*250,x+Escx*270,y+Escy*290);
setlinesyle(1,0,HORN_WIDTH);
line(x+Escx*109,y+Escy*210,x+Escx*109,y+Escy*292);
line(x+Escx*117,y+Escy*190,x+Escx*117,y+Escy*292);
line(x+Escx*60,y+Escy*190,x+Escx*118,y+Escy*190);
line(x+Escx*60,y+Escy*267,x+Escx*260,y+Escy*267);
line(x+Escx*60,y+Escy*210,x+Escx*110,y+Escy*210);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(x-Escx*15,y+Escy*205,lextheight("H")/2,"(fp)");
outtextxy(x-Escx*15,y+Escy*262,lextheight("H")/2,"(fr)");
outtextxy(x+Escx*90-lexwidth("H"),y+Escy*293,"K");
outtextxy(x+Escx*112,y+Escy*293,"L");
settextstyle(SMALL_FONT,HORIZ_DIR,3);
outtextxy(x+48*Escx,y+Escy*171,"2");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(x+Escx*157,y+Escy*180,"^");
outtextxy(x+Escx*267,y+Escy*287,">");
outtextxy(x+Escx*50,y+Escy*280,"0");
outtextxy(x-Escx*25,y+Escy*205,"H (fp)");
outtextxy(x-Escx*25,y+Escy*262,"H (fr)");
outtextxy(x+Escx*50,y+Escy*190,"l");
outtextxy(x+Escx*90,y+Escy*295,"p");
outtextxy(x+Escx*115,y+Escy*295,"fr");
outtextxy(x+Escx*190,y+Escy*295,"frecuencia");
outtextxy(x-Escx*17,y+Escy*177,"|Ha(s)|");
i=0;ii=60*Escx;
setlinesyle(0,2,THICK_WIDTH);
setcolor(9);
while(i<Pl){
    i=PI/710*3.5;ii=Escx;
    X1=cos(i);X2=cos(2*i);Y1=sin(i);Y2=sin(2*i);
    Vf=pow(KR,2)*(pow(X1,2)+pow(Y1,2))*(pow(1+R[1]*X1+X2,2)+pow(O[1]*Y1+Y2,2));
    Cf=(pow(1+R[3]*X1,2)+pow(R[3]*Y1,2))*(pow(1+R[1]*X1+R[2]*X2,2)+pow(R[1]*Y1+R[2]*Y2,2));
    H=(Vf/Cf);
    if(i>.3754*Pl)H=H*100;
    line(ii*x,y+Escy*(290-H*100),ii*x,y+Escy*(290-H*100));
}
setlinesyle(1,1,HORN_WIDTH);
}

/###
#RESPUESTA_BUTLERWORTH: Respuesta de frecuencia prototipo del filtro pasabajos de BUTLERWORTH
#Parámetros : x.- coordenada X de la esquina superior izquierda de la figura
# y.- coordenada Y de la esquina superior izquierda de la figura
#Retorno : Ninguno
#####
void respuesta_butterworth(int x,int y)
(
float A0,R,DD,A00,B0,DD0,ii,H,IMPAR,ress0,resho;
float a,b,d,d0,R,1,R0,R7,I0,I7,A!e,8(1,X1,X2,Y1,Y2,i;
A!e=2;resho=10;ress0=2.179724;a=1;b=2;d=1;A0=13.05548;B=1.424088;DD=4.370609;
setlinesyle(0,0,HORN_WIDTH);

```



```

selviewport(0,0,MaxX,MaxY,1);
la_imagen=imagesize((dix1-1),(diy1-1),(dix2+1),(diy2+1));
switch(paso){
  case 1:
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    if(nbucle !=10 && nbucle !=11){
      buf_imagen=_graphgetwch(la_imagen);
      getimage((dix1-1),(diy1-1),(dix2+1),(diy2+1),buf_imagen);
    }
    selviewport(dix1,diy1,dix2,diy2,1);
    clearviewport();
    settextjustify(LEFT_TEXT, TOP_TEXT);
    selviewport(0,0,MaxX,MaxY,2);
    MainWindow(dix1,diy1,dix2,diy2,1,0,"RESPUESTA DE FRECUENCIA DEL FILTRO");
    switch(caso){
      case 2:x=dix2-54!textwidth("M");y=18!Escy;
        printf(&x,&y,2,"ESTRUCTURACION DEL FILTRO PASABAJOS ELIPTICO");
        x=dix2-54!textwidth("M");y=34!Escy;
        printf(&x,&y,2,"fp :frecuencia de corte      0<fp<INFINITO [Hz]");
        printf(&x,&y,2,"fr :frecuencia de supresión  fp<fr<INFINITO [Hz]");
        printf(&x,&y,2,"Afp:atenuación en fp      0<Afp<20 [dB]");
        printf(&x,&y,2,"Afr:atenuación en fr      0<Afr<100 [dB]");
        printf(&x,&y,2,"s=j2nf con k=2n");
        respuesta_acomodada(45!Escx,-145!Escy);
        break;
      case 3:x=dix2-58!textwidth("M");y=18!Escy;
        printf(&x,&y,2,"ESTRUCTURACION DEL FILTRO PASABAJOS DE BUTTERWORTH");
        x=dix2-54!textwidth("M");y=34!Escy;
        printf(&x,&y,2,"fp :frecuencia de corte      0<fp<INFINITO [Hz]");
        printf(&x,&y,2,"fr :frecuencia de supresión  fp<fr<INFINITO [Hz]");
        printf(&x,&y,2,"Afp:atenuación en fp      0<Afp<20 [dB]");
        printf(&x,&y,2,"Afr:atenuación en fr      0<Afr<100 [dB]");
        printf(&x,&y,2,"");
        printf(&x,&y,2,"s=j2nf con k=2n");
        respuesta_butterworth(45!Escx,-145!Escy);
        break;
      case 4:x=dix2-52!textwidth("M");y=17!Escy;
        printf(&x,&y,2,"ESTRUCTURACION DEL FILTRO PASABANDA ELIPTICO ");
        x=dix2-52!textwidth("M");y=31!Escy;
        printf(&x,&y,2,"fo :frecuencia central      0<fo<INFINITO [Hz]");
        printf(&x,&y,2,"fp :frecuencia de corte      0<pf<fo [Hz]");
        printf(&x,&y,2,"fr :frecuencia de supresión  0<fr<fp [Hz]");
        printf(&x,&y,2,"Afp:atenuación en fp      0<amp<20 [dB]");
        printf(&x,&y,2,"Afr:atenuación en fr      0<awr<100 [dB]");
        printf(&x,&y,2,"s=j2nf con k=2n");
        respuesta_banda(37!Escx,-172!Escy,3);break;
    }
    MainWindow(dix1,diy1,dix2,diy2,1,5,"RESPUESTA DE FRECUENCIA DEL FILTRO");
    setfillstyle(SOLID_FILL, getmaxcolor());
    setcolor(getmaxcolor());
    settextjustify(LEFT_TEXT, TOP_TEXT);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    break;
  case 2:
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    selviewport(0,0,MaxX,MaxY,1);
    if(nbucle !=10 && nbucle !=11){
      putimage((dix1-1),(diy1-1),buf_imagen,COPY_PUT);
      _graphfreech(buf_imagen,la_imagen);
    }
    setfillstyle(SOLID_FILL, getmaxcolor());
    setcolor(getmaxcolor());

```

```

        settextjustify( LEFT_TEXT, TOP_TEXT );
        settextstyle( DEFAULT_FONT, HORIZ_DIR, 1);
        break;
    case 3: setviewport(0,0,MaxX,MaxY,1);
            _graphfreearea (buf_imagen,la_imagen);
            setfillstyle (SOLID_FILL, getmaxcolor());
            setcolor (getmaxcolor());
            settextjustify( LEFT_TEXT, TOP_TEXT );
            settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        }
    }
}
/****
#BORRAR_PAGINA:Despliega en pantalla ayuda de acuerdo al caso ( ACCESO POR EL USUARIO)
#Parámetros :      dix1 .- coordenada X de la esquina superior izquierda de la PAGINA
#                  diy1 .- coordenada Y de la esquina superior izquierda de la PAGINA
#                  dix2 .- coordenada X de la esquina inferior derecha de la PAGINA
#                  diy2 .- coordenada Y de la esquina inferior derecha de la PAGINA
#Retorno :         Ninguno
#####/
void borrar_pagina(int dix1,int diy1,int dix2,int diy2)
{
    setviewport(0,0,MaxX,MaxY,1);
    setviewport(dix1+11*Escx,diy1+22*Escy,dix2-11*Escx,diy2-10*Escy,1);
    clearviewport();
}

```

/f

Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.  
ELEMENOS.C  
Contiene modelos de simulación

1/

```

#include <io.h>
#include <dir.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include "perfil.h"
#include <graphics.h>
#include <stdio.h>
extern float nc, Bpend, Dini, Dfo, ncb, rd[4], D01, MaxX, MaxY, EstX, Escy, Dfp, Dfr, EPS;
extern double CCCO, T0, XCCO0, l, A1;;
extern int indiceg, ncanales, ycanal, pcr, N2, Dnr, N, qg, nbucle, Dn;
extern int cein, ced, ceo, ceosc, cel, ce2, ce3, ce4, ce5, ces, ceosc2, cein2;
extern float A, B0, ff, nc, e2, e3, e4, e5, ein[N], el, eosc, ed, eo, ein2, eosc2, es, W0;
float ddX, dy0[2], dy1[2], CER0[5], POL0[5], Dy0[2], Dy1[2], Dy2[2], Dy3[2], Dy4[2], DDx[2];
int *ppcuadro, npl1, ch, inicio1, inicio;

/!ELDEFASAR: Estructura el Defasador de 90 grados
#Parámetros : Ninguno
#Retorno : Indicador de éxito(S/N)
#####/
char eldefasar(void)
{
char decision;
int i, b, it;
float fr, fp, wp, wr, fs1, DK, DKP, DK1, k2, AK1, AK2, AK3, AK4, q0, q;
float n, nd, o1=0, o2=0, o3=0, o0=0, p, z, zzz, pl;
fp=Dfp; fr=Dfr; n=Dnr;
fs1=3*fr; fr=fr/fs1*100; fp=fp/fs1*100; o1=ff/fs1*100;
fs1=3*ff/fs1*W/nc;
if(fp==0) fp=.1;
if(fr>o1 && o1>fp)
fs1=PI*fp/.022336/nc;
wp=PI*fp/fs1; wr=PI*fr/fs1;
DK=tan(wp); DK=DK;
DK1=1/tan(wr); DK=DK*DK1;
if(DK>1 || DK<=-1){
decision=mensaje(1921Escy, 60, 3, 7, "PELIGRO: El Defasador estructurado puede dar errores. CONTINUAR? [S/N]*");
if(decision=='s' || decision=='S') DK=.99;
else return 0;
}
DK1=(1-tan(EPS/2))/(1+tan(EPS/2)); DK1=pow(DK1, 2);
DKP=sqrt(1-pow(DK, 2));
AK1=AK(DK); AK3=AK(DKP); k2=sqrt(DKP);
q0=.5*(1-k2)/(1+k2); q=q0+2*pow(q0, 5)+15*pow(q0, 9)+150*pow(q0, 13);
q=PI*AK3/AK1; if(q<0) q=q*-1; q=pow(2.71828182, -q);
for(it=0; it<n; it++){
zzz=(4*it+1)*AK3/7/n; o2=0;
for(i=0; i<6; i++){
b=i*(i+1); o3=(2*i+1)*PI*zzz/2/AK1;
if(o3>0) while(o3>2*PI) o3=o3-PI*2;
if(o3<0) while(o3<-PI*2) o3=o3+PI*2;
o2=o2+pow(q, b)*cos(o3);
}
o2=2*pow(q, .25)*o2; o1=0;
for(i=0; i<6; i++){
b=i*(i+1); o3=(2*i+1)*PI*zzz/2/AK1;
if(o3>0) while(o3>PI*2) o3=o3-PI*2;
}
}

```



```

        if(a3<=0)while(a3<-PI*2)a3=a3+PI*2;
        o1=o1*pow(-1,i)*pow(q,b)*sin(a3);
    }
    o1=2*pow(q,.25)*o1;o0=0;
    for(i=1;i<6;i++){
        b=i*(i);o3=(2*i)*PI*zz/2/AK1;
        if(a3>0)while(a3>PI*2)a3=a3-PI*2;
        if(a3<=0)while(a3<-PI*2)a3=a3+PI*2;
        o0=o0+pow(-1,i)*pow(q,b)*cos(a3);
    }
    a0=1+2*o0;o1=o1;o2=o2;
    p=-1/k2*o1/o2*ln(wp);
    if(p<0)z=(1+p)/(1-p);
    else z=(1-p)/(1+p);
    CERD[i1]=z;Dn=n;
}

return i;
}

/* AK: Integral ELIPTICO completo de primer orden
* Parametros: Y.- Constante del integral
* Retorno : Valor del integral
#####/
float AK(float Y)
{
float X, res2;
    X=pow(Y,2);
    res2=PI*(.5+.125*X+.0703125*pow(X,2)+.48828125*pow(X,3)+.037384*pow(X,4));
    return res2;
}

/*DEFASAR : Simula el defasador de 90 grados
* Parametros: derr.- onda a defasar
* Retorno : Valor de la onda defasada
#####/
float defasar(float derr)
{
float y;
    if(inicio==0){
        Dy0[1]=0;Dy0[0]=0;
        Dy1[1]=0;Dy1[0]=0;
        Dy2[1]=0;Dy2[0]=0;
        Dy3[1]=0;Dy3[0]=0;
        Dy4[1]=0;Dy4[0]=0;
        DDx[0]=0;inicio=1;
    }
    DDx[1]=derr;
    Dy0[1]=DDx[0]+CERD[0]*(-DDx[1]+Dy0[0]);
    if(Dn==1){
        y=Dy0[1];goto Dfin0;
    }
    Dy1[1]=Dy0[0]+CERD[1]*(-Dy0[1]+Dy1[0]);
    if(Dn==2){
        y=Dy1[1];goto Dfin1;
    }
    Dy2[1]=Dy1[0]+CERD[2]*(-Dy1[1]+Dy2[0]);
    if(Dn==3){
        y=Dy2[1];goto Dfin2;
    }
    Dy3[1]=Dy2[0]+CERD[3]*(-Dy2[1]+Dy3[0]);
    if(Dn==4){
        y=Dy3[1];goto Dfin3;
    }
}

```

```

    }
    Dy4[1]=Dy3[0]+CERD[4]*(-Dy3[1]+Dy3[0]);
    if(Dn==5){
        y=Dy4[1];
    }
    Dfin4: Dy4[0]=Dy4[1];
    Dfin3: Dy3[0]=Dy3[1];
    Dfin2: Dy2[0]=Dy2[1];
    Dfin1: Dy1[0]=Dy1[1];
    Dfin0: Dy0[0]=Dy0[1];
    DDY[0]=DDY[1];
    return y;
}

```

```

//VCO : Simula el oscilador controlado por voltaje
# Parametros: derr.- onda de control del oscilador
# Retorno : Valor de la onda de oscilacion
#####/
float VCO(float oid)
{

```

```

    double XVC01,XVC02,CCC1,T1,CCC2,K1,OA1;
    int nxx;
    XVC01=1;
    if(oid>0ini)
        XVC01=1+Opendfoid;
    if(oid<-0ini)
        XVC01=1+Opendfoid;
    K1=2*Ofo12;
    if(CCC0==1){
        if(T0>=V2)
            CCC1=-1;
        else
            CCC1=1;
    }
    else {
        if(T0<=(-V2))
            CCC1=1;
        else
            CCC1=-1;
    }
    XVC01=XVC01+CCC1;
    T1=K1*(XVC00+XVC01)*A1/2+T0;
    CCC0=CCC1;T0=T1;XVC00=XVC01;
    XVC02=sin(Pi*T1/2);
    if(CCC1>0) CCC2=1;
    else CCC2=0;
    return XVC02;
}

```

```

//APD : Simula el comparador de fase
# Parametros: einn.- onda 1
# yeosc.- onda 2
# Retorno : Valor de la respuesta del comparador
#####/
float PD(float einn,float yeosc)
{
    float y,eea,eeb,h,pr,ea,eb,m,CE[2];
    pr=2.835638;CE[0]=.8406166;CE[1]=.02460552;
    h=pr*2;m=4*N/hc;
    if(inicial==0){
        dy0[1]=0;dy0[0]=0;dy1[1]=0;dy1[0]=0;
        ddX=0;pd[0]=0;pd[1]=0;pd[2]=0;pd[3]=0;
    }

```

```

        inicial=1;
    }
    dy0[1]=ddX+CE[0]*(-yeoscndy0[0]);
    dy1[1]=dy0[0]+CE[1]*(-dy0[1]+dy1[0]);
    y=dy1[1];
    dy1[0]=dy1[1];
    dy0[0]=dy0[1];
    ddX=yeoscnd;
    ea=.5*einny;
    eb=-.5*einny;
    if(ea<0)ea=0;
    if(eb<0)eb=0;
    eea=1/(n+pr)*((n-pr)*pd[2]+h*(pd[0]+ea));
    pd[0]=ea;pd[2]=eeea;
    eeb=1/(n+pr)*((n-pr)*pd[3]+h*(pd[1]+eb));
    pd[1]=eb;pd[3]=eeeb;
    y=(eea-eeb);
    return y;
}

/!BORRAR : Borra las ventanas graficas de presentacion de resultados
* Parametros: einn .- onda 1
* yeoscnd .- onda 2
* Retorno : Valor de la respuesta del comparador
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!/
void borrar(int caso)
{
    int i,j;
    switch(caso){
        setviewport(0,0,MaxX,MaxY,1);
        case 1: setviewport(1*Escx, 193*Escy, MaxX, MaxY,1);
                clearviewport();
                inil();indiceq=0;
                break;
        case 2: setviewport(146*Escx,199*Escy,649*Escx,306*Escy,1);
                clearviewport();
                break;
        case 3: setviewport( 180*Escx,1.680*Escx,(MaxY-30)*Escy,1);
                clearviewport();
                settlxtstyle( CENTER_TEXT, TOP_TEXT );
                setviewport(0,0,MaxX,MaxY,1);
                rectangle( 180*Escx,0,680*Escx,(MaxY-30)*Escy);
                setlinestyle(1,0,HIGH_WIDTH);
                for( i=0 ; i<ncanales+2-1; ++i ){
                    line(180*Escx,(i+1)*ycanal+2*Escy,680*Escx,(i+1)*ycanal+2*Escy);
                    ++i;
                }
                settlxtstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
                setlinestyle(1,0,HIGH_WIDTH);
                for( j=1 ; j<nc; ++j )
                    line(100*Escx+j*pxc,2*Escy,180*Escx+j*pxc,MaxY-30*Escy );
                settlxtstyle( CENTER_TEXT, TOP_TEXT );
                settlxtstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
                break;
        case 4: setviewport(100*Escx,267*Escy,610*Escx,288*Escy,1);
                clearviewport();
                break;
        case 5: setviewport(480*Escx,267*Escy,681*Escx,288*Escy,1);
                clearviewport();
                break;
    }
    setviewport(0,0,MaxX,MaxY,1);
}

```

```

    setfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    settextjustify( LEFT_TEXT, TOP_TEXT );
    settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
    setviewport(0,0,MaxX,MaxY,1);
}

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
CONTROL.C
Archivo central de comandos en la simulación de los bucles
*/

#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include "proyfil.h"
#include <graphics.h>
#include <stdio.h>
extern int p2,p_entrada,p_fpb,p_yco,p_def, p_fpban,cfl,cf2,nbucle,np11,nbb;
extern float A,00,ff,e2,e3,e4,e5,ein[H],el,eosc,ed,eo,ein2,eosc2,es;
extern float MaxX,MaxY,Estx,Escy,nc,ncb,W0;
extern int indiceg,gg,M,ch,pa4,Dn,ppcuadro,L[13],H2,npantall,ppbit;
extern int cein,ced,ceo,ceosc,cel,ce2,ce3,ce4,ce5,ces,ceosc2,cein2;
extern double l,At;
int tba[13],gg,M2r,Dnr;
float ncr,ncbr;

/*MAIN:Comandos principales en el comportamiento de SIMULAR.EYE
*Parametros : Ninguno
*Retorno : Ninguno
*/
void main(void)
{
FILE *fp;
Initialize();
cleardevice();
fp=fopen("a","rb");
rewind(fp);
fread(&nbucle,sizeof(int),1,fp);
fread(&np11,sizeof(int),1,fp);
fclose(fp);
CALRUC2();

}/*CALRUC2:Comandos principales en la simulación
* Parametros : Ninguno
* Retorno : Ninguno
*/
void CALRUC2(void)
{
FILE *fp;
char decision;
float co,eoscn,eosc2n;
size_t sscuadro;
size_t ssba;
int esp,espy,espyb,espxb,eleccion,r,lt,i,nciclo=1,op;
settextstyle(SMALL_FONT,HORIZ_DIR,4);
espx=4*textwidth("H")+8;
espy=textheight("H")+10;
espxb=5*textwidth("H")+4;
espyb=2*textheight("H")+6;

```

```

settextstyle(DEFAULT_FONT,HWR17,01R,1);
sscuardro = imagesize(0,0,espx,espy);
ppcuadro = _graphgetkee(sscuadro);
ssha = imagesize(0,0,espxb,espyb);
for(r=1;r<13;+r)
    ha[r] = _graphgetkee(ssha);
fp=fopen("d","rb");
rewind(fp);
fread(&ch,sizeof(int),1,fp);
fclose(fp);
leer();
Dnr=0n;ncr=H2*ncb;
if(nbucle==7 || npl==4)ncr=H2*nrb/2;
H2r=H2;ncbr=ncb;
RE12:cleardevice();i=0;
switch(nbucle){
    case 2: barral(Escx*5,Escy*5,Escx*700,Escy*185,5*Escx,11,1);
            lema(10*Escy,1,"BLOQUE DE COSTAS GENERAL ");
            forma Buc2(-30*Escx,50*Escy);
            QGRPLL(Escx*420,Escy*80,0);
            break;
    case 5: barral(Escx*5,Escy*5,Escx*700,Escy*185,5*Escx,10,1);
            lema(10*Escy,1,"BLOQUE DE COSTAS CON DECISION DENTRO DE UN BRAZO");
            forma Buc5(-40*Escx,60*Escy);
            QGRPLL(Escx*440,Escy*60,0);
            break;
    case 6: barral(Escx*5,Escy*5,Escx*700,Escy*185,5*Escx,7,1);
            lema(10*Escy,1,"BLOQUE DE COSTAS NUMERICO");
            forma Buc6(-40*Escx,60*Escy);
            QGRPLL(Escx*440,Escy*70,0);
            break;
    case 0: barral(Escx*10,Escy*5,Escx*700,Escy*185,5,11,1);
            lema(10*Escy,1,"BLOQUE IGUALADOR DE FASE [ PLL ]");
            forma(-115*Escx,57*Escy);
            QGRPLL(405*Escx,48*Escy,0);
            break;
    case 3: barral(Escx*10,Escy*5,Escx*700,Escy*185,5,11,1);
            lema(10*Escy,1,"BUCLE CON REMODULACION POR EL OSCILADOR LOCAL ");
            forma Buc3(-30*Escx,50*Escy);
            QGRPLL(Escx*470,Escy*70,0);
            break;
    case 4: barral(Escx*10,Escy*5,Escx*700,Escy*185,5,11,1);
            lema(10*Escy,1,"BUCLE CON REMODULACION POR LA SENAL RECIBIDA ");
            forma Buc4(-35*Escx,50*Escy);
            QGRPLL(Escx*480,Escy*70,0);
            break;
    case 1: barral(Escx*10,Escy*5,Escx*700,Escy*185,5,11,1);
            lema(10*Escy,1,"BLOQUE CON MULTIPLICACION DE FRECUENCIA");
            forma Buc1(-45*Escx,50*Escy);
            QGRPLL(Escx*440,Escy*70,0);
            break;
    case 7: barral(Escx*10,Escy*5,Escx*700,Escy*185,5,11,1);
            lema(10*Escy,1,"BLOQUE DE COSTAS CON REMODULACION PARA QPSK");
            forma Buc7(-43*Escx,53*Escy);
            QGRPLL(Escx*480,Escy*70,0);
            break;
}
do {
    RE13:nc=ncr;H2=H2r;ncb=ncbr;indiceg=0;
    do {
        eleccion=mensaje(282*Escy,60,5,7, "BRAFICAR ONDAS:[F2]UNA a UNA [F3]VARIOS EJES [F5]MISMO
        EJE [F4]SALIR");
    }
}

```

```

}while((eleccion != 316) && (eleccion != 317) && (eleccion != 319) && (eleccion != 318));
if(eleccion != 318)npantalla=numero_ciclos();
init(); initc();
if(nbucle == 2 !! nbucle==5 !! nbucle == 6 !! nbucle == 7)
    decision=eldefasar();
if(decision==0){
    ch=323;
    fp=fopen("d", "wb+");
    fwrite(&ch, sizeof(int), 1, fp);
    fclose(fp); goto SALL;
}
if(eleccion == 314)
{
    do
    {
        RE14:
        StatusLine (1, "[F9]MS [F10]Salir [ESC]Borrar [CURSOR]Elegir [ENTER]Graficar [F7]Imprimir");
        init(); initc(); eos = cos(W0*t-00); gg = 2; nbb=0; ppbit=0;
        switch(nbucle){
            case 2: op = ondaread_BUC2 (-30*Escx, 50*Escy); break;
            case 5: op = ondaread_BUC2 (-40*Escx, 60*Escy); break;
            case 6: op = ondaread_BUC2 (-40*Escx, 60*Escy); break;
            case 0: op = ondaread_BUC2 (-115*Escx, 57*Escy); break;
            case 3: op = ondaread_BUC2 (-50*Escx, 50*Escy); break;
            case 4: op = ondaread_BUC2 (-35*Escx, 50*Escy); break;
            case 1: op = ondaread_BUC2 (-45*Escx, 50*Escy); break;
            case 7: op = ondaread_BUC2 (-43*Escx, 53*Escy); break;
        }
        if(op==318)goto SALL;
        if(op==ESC){
            borrar(1); goto RE14;
        }
        if(op==324){
            borrar(1); goto RE13;
        }
        if(op==323){
            fp=fopen("d", "wb+");
            fwrite(&op, sizeof(int), 1, fp);
            fclose(fp); goto SALL;
        }
        co=Escy*10;
        switch (op)
        {
            case 1: cein = co; break;
            case 2: ceosc = co; break;
            case 3: ce1 = co; break;
            case 4: ce2 = co; break;
            case 5: ces = co; break;
            case 6: ced = co; break;
            case 7: cein2 = co; break;
            case 8: ceosc2 = co; break;
            case 9: ceo = co; break;
            case 10: ce3 = co; break;
            case 11: ce4 = co; break;
            case 12: ce5 = co; break;
        }
        StatusLine(1, "[F4]Menu P. [F9]Menu S. [F10]Salir [ESC]Borrar [P]Parar [ENTER]Continuar");
        if(indiceq==0) barral(140*Escx, 205*Escy, 650*Escx, 320*Escy, 3, 1, 1);
        if (op != 0){
            if(nbucle ==7 !! npl==4) mod4psk();
            else
                mod2psk();
            do{

```



```

    } while (op != 0);
}

if(eleccion == 319)
{
    borrar(1);
    do
    {
        initc();eosc=cos(M_PI-0.01);
        RE16:nbb=0;ppbil=0;
        Statusline(1,"[F10]Salir [F9]Salir [ENTER]Seleccionar [F3]Graficar [F7]Reparar");
        init();gg = 3;
        elegir_grafico(1,1);
        if(ch== 324){
            nbb=0;ppbil=0;nciclo=1;p2=0;
            for(i=0;i<13;i++)
                L[i]=0;
            goto RE12;
        }
        Statusline(1,"[F4]Menu Prin. [F9]Menu Sec. [F10]Regraf. [ESC]Borrar [F7]Carar [ENTER]Continuar ");
        rectangle(130*Escr,204*Escy,652*Escx,321*Escy);
        if (op != 0){
            if(nbucle ==7 ;; op1==4) mod4psk();
            else
                mod2psk();
            do{
                for(N=0;N<N1;N++){
                    switch(nbucle){
                        case 2:
                            CBUC2();break;
                        case 5:
                            CBUC5();break;
                        case 6:
                            CBUC6();break;
                        case 0:
                            CBUC0();break;
                        case 3:
                            CBUC3();break;
                        case 4:
                            CBUC4();break;
                        case 1:
                            CBUC1();break;
                        case 7:
                            CBUC7();break;
                    }
                    t=t+1;
                }
                graficar(nciclo);
                if(ch== 323){
                    nbb=0;ppbil=0;nciclo=1;p2=0;
                    fp=fopen("d","wb+");
                    fwrite(&ch,sizeof(int),1,fp);
                    fclose(fp);goto SAL1;
                }
                if(ch== 324){
                    nbb=0;ppbil=0;nciclo=1;p2=0;
                    for(i=0;i<13;i++)
                        L[i]=n;
                    goto RE12;
                }
                if(ch==ESC){
                    borrar(1);nciclo=1;goto RE16;
                }
            } while (ch!=310){

```



```

                                nciclo--; goto SAL;
                                }
                                }
                                nciclo++;
                                if(npantalla!=1){
                                    if(nbucle ==7 || np1==4) mod4psk();
                                    else mod2psk();
                                    if(nciclo<=npantalla){
                                        setviewport(0,0,MaxX,MaxY,1);
                                        do{
                                            decision=mensaje(172+Escy,50,5,4, " [F5] Graficar siguientes bits [F6]
                                            Graficar otras Onda");
                                        }while(decision !=63 && decision != 64);
                                        if(decision==63 ){
                                            borrar(12);
                                            setviewport(0,205+Escy,MaxX,319+Escy,1);
                                            line(140+Escx,60+Escy,430+Escx,Escy+60);
                                        }
                                        if(decision==64){
                                            nciclo=npantalla+1;borrar(1);
                                        }
                                    }
                                }while(nciclo<=npantalla);
                                }
                                init(); initc();nciclo=1;nbb=0;ppbil=0;
                                } while (op != 0);
                                }

                                if(eleccion==317)
                                {
                                    gg = 1;nbb=0;ppbil=0;borrar(1);
                                    setlinesyle (0, 0, HORB_WIDTH);
                                    setcolor (getmaxcolor());
                                    setfillstyle (SOLID_FILL, getmaxcolor());
                                    if(nbucle ==7 || np1==4) mod4psk();
                                    else mod2psk();
                                    indiceg = 0;
                                    do{
                                        esc=rao*(40+1-00);
                                        for(N=0;N<H;N++){
                                            if(N%2>0)nciclo++;
                                            if(npantalla>1 && N%2>0 ){
                                                if(nbucle ==7 || np1==4) mod4psk();
                                                else mod2psk();
                                                if(nciclo<=npantalla){
                                                    do{
                                                        decision=mensaje(172+Escy,50,5,4, " [F5] Graficar siguientes
                                                        bits [F6] Graficar otras Onda");
                                                    }while(decision !=63 && decision != 64);
                                                    if(decision==63 ){
                                                        borrar(3);
                                                        setlinesyle(0,0,THICK_WIDTH);
                                                        setviewport(0,0,MaxX,MaxY-31+Escy,1);
                                                    }
                                                    if(decision==64){
                                                        nciclo=npantalla+1;
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                                switch(nbucle){
                                    case 2: CBUC2();break;
                                }

```

```

        case 5: CBUC5();break;
        case 4: CBUC4();break;
        case 0: CBUC0();break;
        case 3: CBUC3();break;
        case 9: CBUC9();break;
        case 1: CBUC1();break;
        case 7: CBUC7();break;
        }

        li=A1;
        graficar(nciclo);
        if(ch== 323){
            nbb=0;ppbil=0;nciclo=1;p2=0;
            fp=fopen("d","wb");
            fwrite(&ch,sizeof(int),1,fp);
            fclose(fp);goto SAL1;
        }
        if(ch== 324){
            nbb=0;ppbil=0;nciclo=1;p2=0;
            for(i=0;i<13;i++)
                L[i]=0;
            goto DE12;
        }
        if(ch== 318){
            nbb=0;ppbil=0;nciclo=1;goto SAL1;
        }
    }
    }while(nciclo<=npantalla);
}
} while (eleccion != 318);
SAL1:graphfreeeee (ppcuadro, sscuadro);
for(r=1;r<13;+r)
    _graphfreeeee (ba[r], ssha);
}

/*CBUC2:Bucl e de costas general
* Parametros : Ninguno
* Retorno : Ninguno
*/
void CBUC2 (void)
{
    float eoscn,eosc2n;
    eoscn=eosc/E2;
    eosc2n=defasar(eoscn); eosc2=E2*eosc2n;
    ed=ein[N]*eosc2;
    e1=ein[N]*eosc;
    if (cf2=='e';cf2=='E')eo=filtro(ed,2);
    else eo=filtro(ed,2);
    if (cf1=='e';cf1=='E')e2=filtro(e1,1);
    else e2=filtro(e1,1);
    ein2=e2*eo;
    eosc=VCQ(-ein2);
    es=e2>0? 1:-1;
}

/*CBUC5:Bucl e de costas con deteccion dentro de un brazo
* Parametros : Ninguno
* Retorno : Ninguno
*/
void CBUC5 (void)
{
    float eoscn;
    eoscn=eosc/E2;eosc2=defasar(eoscn);
    ed=ein[N]*eosc2;

```

```

        el=ein[M]*eosc;
        if (cf2=='e';cf2=='F')eo=filtro(ed,2);
        else eo=filtrob(ed,2);
        if (cf1=='e';cf1=='E')e2=filtro(e1,1);
        else e2=filtrob(e1,1);
        es=e2>0? 1:-1;
        ein2=eotes;
        eosc=VCO(-ein2);
    }
/ICRUC6:Bucla de costas NUMERICO
#Parametros : Ninguno
#Retorno : Ninguno
#####/
void CRUC6 (void)
{
    float eosc;
        eosc=eosc/E2;eosc2=defasar (eosc);
        ed=ein[M]*eosc2;
        el=ein[M]*eosc;
        if (cf2=='e';cf2=='F')eo=filtro(ed,2);
        else eo=filtrob(ed,2);
        if (cf1=='e';cf1=='F')e2=filtro(e1,1);
        else e2=filtrob(e1,1);
        es=e2>0? 1:-1;
        e3=e2>0.33? 1:-1;
        if(es==1 && e3==1)ein2=1;
        else ein2=0;
        eosc=VCO(-ein2);
    }
/ICRUC3:Bucla CON REMODULACION POR FL VCO
# Parametros : Ninguno
# Retorno : Ninguno
#####/
void CRUC3 (void)
{
    float einn,ein2n;
        el=ein[M]*eosc;
        if (cf1=='e';cf1=='E')
            e2=filtro(e1,1);
        else
            e2=filtrob(e1,1);
        es=e2>0?1:-1;
        ein2=esteosc;
        ed=PD(ein[M],ein2);
        eosc=VCO(-ed);
    }
/ICRUC4:Bucla CON REMODULACION POR LA ONDA RECIBIDA
# Parametros : Ninguno
# Retorno : Ninguno
#####/
void CRUC4 (void)
{
        el=ein[M]*eosc;
        if (cf1=='e';cf1=='E')
            e2=filtro(e1,1);
        else
            e2=filtrob(e1,1);
        es=e2>0? 1:-1;
        ein2=ein[M]*es;
        ed=PD(eosc,ein2);
        eosc=VCO(ed);
    }
/ICRUCO:PLL
# Parametros : Ninguno

```

```

# Retorno : Ninguno
#####/
void CBUC0 (void)
{
    ed=PD(ein[M],eosc);
    if (cf1=='e';cf1=='E')
        eo=filtro(ed,1);
    else eo=fillrob(ed,1);
    eosc=VCO(eo);
}
/*CBUC1:Bucle CON MULTIPLICACION
# Parametros : Ninguno
# Retorno : Ninguno
#####/
void CBUC1 (void)
{
    float einn;
    einn=ein[M];
    ein2=doblador(einn,2);
    eosc2=doblador(eosc,1);
    ed=PD(ein2,eosc2);
    e1=eosc4ein[N];
    if (cf2=='e';cf2=='E')eo=filtro(ed,2);
    else eo=fillrob(ed,2);
    if (cf1=='e';cf1=='E')e2=filtro(e1,1);
    else e2=fillrob(e1,1);
    es=e2>0? 1:-1;
    eosc=VCO(-eo);
}
/*CBUC7:Bucle de costas para QPSK
# Parametros : Ninguno
# Retorno : Ninguno
#####/
void CBUC7 (void)
{
    eosc2=defasar(eosc);
    ed=ein[N]*eosc2;e1=ein[M]*eosc;
    if (cf2=='e';cf2=='E')eo=filtro(ed,2);
    else eo=fillrob(ed,2);
    if (cf1=='e';cf1=='E')e2=filtro(e1,1);
    else e2=fillrob(e1,1);
    es=e2>0?1:-1;
    e3=eo>0? 1:-1;
    e4=e3es;e5=e2e3;
    ein2=-e4e5;eosc=VCO(ein2);
}

/*
Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
GRAFICAR.C
Archivo de comandos para graficar resultados
*/
#include "math.h"
#include "stdio.h"
#include "conio.h"
#include "proyfil.h"
#include "stdlib.h"
#include "dos.h"
#include "graphics.h"
extern int L[13],indr,des,Ann,Bnn,An,Bn,inicio,iniciol,ch,gg,p_entrada,np11,in[20],*ppcuadro,H2,M2r,indiceg;
extern float ff,indice,nc,ncr,Escx,Escy,MaxY,MaxX,OU1,ncb,ncbr;

```

```

extern int iniciobanda0, iniciobanda0, inicioa, inicioeb, inicioha, iniciohb;
float W0, K, ed, 00, KK, eo, ein[N], ein2, eosr2, eosr, es, e1, e2, e3, e4, e5, pd[8], X[3];;
int A, N, a, HP, KK0, KK1, KK2, KK3, KK4, XY5, YFA, XY7, XY8, XY9, KK13, F14, KK15, pce4, pce5, pceosc2;
int pun[10][12], phit, pphit, ycanal, ncanalcs, nbh, nbucle, npanlalla, pcein2, pces, pcsb, pce3;
int dc, Adalo, conl, c, MX, MX3, MX4, MX5, MX6, MX7, MX8, MX9, csb, ce3, ce4, ce5, pcc, pce0, pcel, pce2;
int cein, ceosc, ced1, ced2, ced, ceol, cen2, cen3, ceo, cel, ce2, ceosc2, cein2, ces, pcein, pceosc, pced;
double XVC00, T0, CCC0, t, At;
long taag;
char far tbufq;
void inilc(void);
void mov (int x1, int yy, int x2, int lnr);
void beep (int f, int d);

```

/\*GRAFICAR: Controla las pantallas de resultados

\* Parametros :nciclo.- Indica el # de la pantalla de resultados actual

\* Retorno : Ninguno

#####

void graficar(int nciclo)

```

{
int xstep,ystep,i,j,hhh,vvv,chl,decision,decisionl;
int #pp,xi,yi,eycanal,c,f,k,a,m,z,op,y,MAXX,oc,nuu;
float aa,co;
char linea[80];
FILE *p;
size_t ss;

if(N<2 && ((gg==2 && nciclo==1);!gg==3) ){
selviewport(0,0,MAXX,MAXY,1);
if(cein!=0){
setcolor(14);
outtextxy(100,210*Escy,"[1]");
}
if (cein2 !=0){
setcolor(11);
outtextxy(100,270*Escy,"[7]");
}
if (ceosc2 !=0){
setcolor(15);
outtextxy(100,260*Escy,"[6]");
}
if (ceo !=0){
setcolor(9);
outtextxy(100,290*Escy,"[9]");
}
if (ced !=0){
setcolor(4);
outtextxy(100,280*Escy,"[8]");
}
if (cel !=0){
setcolor(6);
outtextxy(100,230*Escy,"[3]");
}
if (ce2 !=0){
setcolor(12);
outtextxy(100,240*Escy,"[4]");
}
if (ces !=0){
setcolor(10);
outtextxy(100,250*Escy,"[5]");
}
if (ceosc!=0){
setcolor(13);
outtextxy(100,220*Escy,"[2]");
}
}

```

```

    }
    if (ce3 !=0){
        setcolor(12);
        outtextxy(100,300*Escy,'[0]');
    }
    if (ce4 !=0){
        setcolor(10);
        outtextxy(100,310*Escy,'[a]');
    }
    if (ce5!=0){
        setcolor(13);
        outtextxy(100,320*Escy,'[b]');
    }
    setviewport(0,205*Escy,MaxX,319*Escy,1);
}
if(indiceg==0){
    switch(gg){
        case 1:ncanales=mensaje(200*Escy,55,6,20, " CUANTOS EJES en la pantalla? (Numero ENTERO (1..9)");
            ycanal=(MaxY-30)*Escy/2/ncanales;
            rectangle(MaxX/2-194*Escx,194*Escy,MaxX/2+135*Escx,328*Escy);
            settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
            outtextxy(MaxX/2-170*Escx,306*Escy," Distribucion de la PANTALLA");
            outtextxy(MaxX/2-170*Escx,316*Escy," de presentacion de resultados");
            sellinesstyle(0,0,THICK_WIDTH);
            rectangle(MaxX/2-80*Escx,197*Escy,MaxX/2+120*Escx,297*Escy);
            eycanal=100*Escy/2/ncanales;
            nue=0;
            for( i=0 ; i<(ncanales*2-1); ++i ){
                x=MaxX/2-175*Escx;y=(i+1)*eycanal+197*Escy;
                sellinesstyle(1,1,HORN_WIDTH);
                line(MaxX/2-80*Escx,y+4*Escy,MaxX/2+120*Escx,y+4*Escy);
                nue++;
                gprintf (x, y,1, "EJE [%2d]",nue);
                ++i;
            }
            sellinesstyle(0,0,HORN_WIDTH);
            for(i=0;i<13;i++)
                !i]=0;
            elegir_grafico(ycanal,ncanales);
            if(ch==318)return;
            if(ch==324)return;
            for(i=1;i<(ncanales+1);++i)
                {
                    if(pcein==i)ubicar(i,ncanales,ycanal,' 1 ');
                    if(pceosc==i)ubicar(i,ncanales,ycanal,' 2 ');
                    if(pce1==i)ubicar(i,ncanales,ycanal,' 3 ');
                    if(pce2==i)ubicar(i,ncanales,ycanal,' 4 ');
                    if(pce3==i)ubicar(i,ncanales,ycanal,' 5 ');
                    if(pce4==i)ubicar(i,ncanales,ycanal,' 6 ');
                    if(pcein2==i)ubicar(i,ncanales,ycanal,' 7 ');
                    if(pceosc2==i)ubicar(i,ncanales,ycanal,' 8 ');
                    if(pceo==i)ubicar(i,ncanales,ycanal,' 9 ');
                    if(pce3==i)ubicar(i,ncanales,ycanal,' 0 ');
                    if(pce4==i)ubicar(i,ncanales,ycanal,' a ');
                    if(pce5==i)ubicar(i,ncanales,ycanal,' b ');
                }
            StatusLine(1,"[F4]Menu Prin. [F9]Menu Sec. [F10]Salir [P]parar [ENTER]Continuar");
            settextjustify(CENTER_TEXT, TOP_TEXT );
            rectangle( 180*Escx,0,680*Escx,(MaxY-30)*Escy);
            sellinesstyle(1,0,HORN_WIDTH);
            for( i=0 ; i<(ncanales*2-1); ++i ){
                line(180*Escx,(i+1)*ycanal+6*Escy,680*Escx,(i+1)*ycanal+6*Escy);
            }
        }
    }
}

```

```

        ++i;
    }
    settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    setlinestyle(1,0,HORN_WIDTH);
    for( j=1 ; j<=2; ++j )
        line(180*Escx+j*pcct*Escx,2*Escy,180*Escx+j*pcct*Escx,MaxY-31*Escy );
    settextjustify( CENTER_TEXT, TOP_TEXT );
    settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    setlinestyle(1,0,HORN_WIDTH);
    setlinestyle(0,0,THICK_WIDTH);
    setviewport(0,0,MaxX,MaxY-31*Escy,1);
    break;
case 2:
    setviewport(0,205*Escy,MaxX,319*Escy,1);
    line(140*Escx,60*Escy,650*Escx,Escy*60);
    KK1=40*Escy;KK2=40*Escy;KK3=40*Escy;KK4=40*Escy;
    KK6=40*Escy;KK7=40*Escy;KK8=40*Escy;KK9=40*Escy;
    KK13=40*Escy;KK14=40*Escy;KK15=40*Escy;KK5=40*Escy;
    break;
case 3:
    setviewport(0,205*Escy,MaxX,319*Escy,1);
    line(140*Escx,60*Escy,650*Escx,Escy*60);
    taq=imagesize(139*Escx,205*Escy,650*Escx,320*Escy);
}
indiceq=1;
}
if (gg==1) mc=M*Escx+180*Escx;
if (gg==2) mc=M*Escx+150*Escx;
if (gg==3){
    settextjustify(CENTER_TEXT,TOP_TEXT);
    setlinestyle(0,0,HORN_WIDTH);
    bufq=malloc(taq);
    if(!bufq){
        mensaje(222*Escy,35,0,23, " E R R O R !  MEMORIA RAM INSUFICIENTE");
        exit (1);
    }
    mc=141*Escx;
}
if(gg==3)
    line(140*Escx,60*Escy,650*Escx,Escy*60);
if(cein!=0){
    setcolor(14);
    line(mc,-KK1*ein[M]*cein,mc,-KK1*ein[M]*cein);
}
if (cein2 !=0){
    setcolor(11);
    line (mc,cein2-KK3*ein2,mc,cein2-KK3*ein2);
}
if (ceosc2 !=0){
    setcolor(15);
    line (mc,ceosc2-KK4*eosc2,mc,ceosc2-KK4*eosc2);
}
if (csb !=0){
    setcolor(2);
    line (mc,-KK0*(tdato)+csb+2,mc,-KK0*(tdato)+csb+2);
}
if (ceo !=0){
    setcolor(9);
    line (mc,ceo-KK6*eo,mc,ceo-KK6*eo);
}
if (ced !=0){
    setcolor(4); line (mc,ced-KK5*ed,mc,ced-KK5*ed);
}

```

```

if (ce1 !=0){
    setcolor(6); line (ac,ce1-KK7*te1,ac,ce1-KK7*te1);
}
if (ce2 !=0){
    setcolor(12); line (ac,ce2-KK8*te2,ac,ce2-KK8*te2);
}
if (ces !=0){
    setcolor(10); line(ac,ces-KK9*tes,ac,ces-KK9*tes);
}
if (ceosc!=0){
    setcolor(13); line (ac,-KK2*teosc+ceosc,ac,-KK2*teosc+ceosc);
}
if (ce3 !=0){
    setcolor(12); line (ac,ce3-KK13*te3,ac,ce3-KK13*te3);
}
if (ce4 !=0){
    setcolor(10); line(ac,ce4-KK14*te4,ac,ce4-KK14*te4);
}
if (ce5!=0){
    setcolor(13); line (ac,-KK15*te5+ce5,ac,-KK15*te5+ce5);
}
if(qg==3){
    settextjustify(CENTER_TEXT, TOP_TEXT);
    setlinestyle(0,0,NORM_WIDTH);
    getimage(139*Escx,(205-205)*Escy,650*Escx,(320-205)*Escy,bufg);
    putimage(190*Escx,(205-205)*Escy,bufg,COPY_PUT);
    free(bufg);
}
if(N+2>N && qg==1 &&(ncirclo>npenalla)){
    setviewport(0,0,MaxX,MaxY,1);
    setviewport(0,MaxY-25*Escy,MaxX,MaxY,1);
    clearviewport();
    setviewport(0,0,MaxX,MaxY,1);
    do{
        decision=aensaje(320*Escy,65,5,21, "[F4]Menu P. [F9]Menu S. [F10]Salir [F1]Diagrama de
        bloques [F7]Imprimir");
        if(decision==315)
            {
                settextstyle( DEFAULT_FONT, NORM12_DIR, 1 );
                setlinestyle(0,0,NORM_WIDTH);
                switch(nucleo){
                    case 0:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,500);break;
                    case 1:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,501);break;
                    case 2:
                        ayuda(105*Escx,5*Escy,614*Escx,190*Escy,502);break;
                    case 3:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,503);break;
                    case 4:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,504);break;
                    case 5:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,505);break;
                    case 6:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,506);break;
                    case 7:
                        ayuda(115*Escx,5*Escy,604*Escx,190*Escy,507);break;
                }
            }
        if(decision == 321) GrPrint();
    }while( decision != 318 && decision != 323 &&decision != 324);
}

```



```

if((kbbil() && M>5) || M<2*N){
    if(M+2<N){
        sound(500);delay(30);sound(800);
        delay(50);sound(210);delay(20);
        nosound();ch=lecla();
    }
    if(M+2>N)
        ch=decision;
    if( ch == 323 ){
        setviewport (0, 0, MaxX, MaxY, 1);
        clearviewport();
        sellinesyle (0, 0, HORN_WIDTH);
        setcolor (getmaxcolor());
        setfillstyle (SOLID_FILL, getmaxcolor());
    }
    if( ch == 324 && (gg==1 || gg==3)){
        setviewport (0, 0, MaxX, MaxY, 1);
        clearviewport();
        sellinesyle (0, 0, HORN_WIDTH);
        setcolor (getmaxcolor());
        setfillstyle (SOLID_FILL, getmaxcolor());
    }
    if(ch=='p' || ch=='P'){
        do{
            co=getch();
        }while(co != '\r');
    }
}
}

/*UBICAR: Controla la presentacion del # de onda que se esta graficando en el eje
# Parametros :i.- Indicador de eje
# ncanales.- # de ejes en la pantalla de resultados
# ycanal.- posicion vertical del eje
# msg.- mensaje(contiene el # de onda graficando)
# Retorno : Ninguno
#####/
void ubicar(int i,int ncanales,int ycanal,char msg)
{
    int j;
    for(j=0;j<12;++){
        if(pun[i][j]==0)
        {
            poner(i,j,ycanal,msg);
            pun[i][j]=1;
            return;
        }
    }
}

/*PONER: Presenta # de onda que se esta graficando en el eje
# Parametros :i.- Indicador de eje
# j.-Posicion horizontal del mensaje
# ycanal.- posicion vertical del eje
# msg.- mensaje(contiene el # de onda graficando)
# Retorno : Ninguno
#####/
void poner(int i,int j,int ycanal,char msg)
{
    int dix1,dix2,diy1,diy2,loc;
    if(j<6){
        switch(i){

```

```

        case 1:diy1=ycanal-64Escy;break;
        case 2:diy1=31ycanal-64Escy;break;
        case 3:diy1=51ycanal-64Escy;break;
        case 4:diy1=71ycanal-64Escy;break;
        case 5:diy1=91ycanal-64Escy;break;
        case 6:diy1=111ycanal-64Escy;break;
        case 7:diy1=131ycanal-64Escy;break;
        case 8:diy1=151ycanal-64Escy;break;
        case 9:diy1=171ycanal-64Escy;break;
    }
}
if(j)>5 && j<12{
    switch(i){
        case 1:diy1=ycanal+12Escy;break;
        case 2:diy1=31ycanal+12Escy;break;
        case 3:diy1=51ycanal+12Escy;break;
        case 4:diy1=71ycanal+12Escy;break;
        case 5:diy1=91ycanal+12Escy;break;
        case 6:diy1=111ycanal+12Escy;break;
        case 7:diy1=131ycanal+12Escy;break;
        case 8:diy1=151ycanal+12Escy;break;
        case 9:diy1=171ycanal+12Escy;break;
    }
    j-=6;
}
loc = 3*textwidth( "M" ); ;
dix1=135Escx-j*loc;dix2=dix1+loc-1;
sellinestyle (0, 0, NORM_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
selfillstyle (SOLID_FILL, 0);
bar (dix1Escx, diy1Escy, dix2Escx,(diy1+16)Escy);
setcolor (getmaxcolor());
rectangle ((dix1+1)Escx, (diy1+1)Escy, (dix2-1)Escx,(diy1+15)Escy);
selfillstyle (SOLID_FILL, getmaxcolor());
bar ((dix1+3)Escx, (diy1+3)Escy, (dix2-3)Escx,(diy1+13)Escy);
setcolor (0);
outtextxy( (dix1+1)Escx,(diy1+5)Escy, msg);
}

/*ELEGIR_GRAFICO: Controla la eleccion de la onda se desea graficar
* Parametros :ycanal.- posicion vertical del eje
* ncanales.-# de ejes en la pantalla de resultados
* Retorno : Ninguno
#####
void elegir_grafico(int ycanal,int ncanales)
{
int co,graf=0,a,op,x,y;
char linea[80],*p;
a=(MaxX-24Escx)/2;des=0;
if((gg==1)StatusLine (1, "[CURSOR]Elegir FONTO  [ENTER]Seleccionar  [F3]Graficar  [F10]Salir");
do{
    switch(abucle){
        case 0:
            op = endread_BUC2 (-115Escx,57Escy);break;
        case 1:
            op = endread_BUC2 (-45Escx,50Escy);break;
        case 2:
            op = endread_BUC2 (-30Escx,50Escy);break;
        case 3:
            op = endread_BUC2 (-50Escx,50Escy);break;
        case 4:

```

```

        op = ondaread_BUC2 (-354Escx,504Escy);break;
case 5:
        op = ondaread_BUC2 (-404Escx,604Escy);break;
case 6:
        op = ondaread_BUC2 (-404Escx,604Escy);break;
case 7:
        op = ondaread_BUC2 (-434Escx,534Escy);break;
    }
    if(op==324){
        ch=324;return;
    }
switch(qg){
case 1:
    if(L[indr] !=0)
    do {
        if(op != 317 && op != 318)des=mensaje(3254Escy,50,5,23, " EL GRAFICO se
UBICA en el EJE ? (NUMERO: 1... 8 EJES )");
        if(des==317 !! op == 317)break;
        if(des==318 !! op == 318){
            ch=318;return;
        }
        if(des==315)
            ayuda(m-3444Escx,704Escy,m+3444Escx,2004Escy,23);
        if(des>ncanales && des != 315)
            mensaje(2224Escy,35,0,23, " E R R O R ! NO existe este canal");
    }while(des>ncanales );
    switch(des){
        case 1:co=y canal+64Escx;break;
        case 2:co=34Escx+y canal+64Escx;break;
        case 3:co=54Escx+y canal+64Escx;break;
        case 4:co=74Escx+y canal+64Escx;break;
        case 5:co=94Escx+y canal+64Escx;break;
        case 6:co=114Escx+y canal+64Escx;break;
        case 7:co=134Escx+y canal+64Escx;break;
        case 8:co=154Escx+y canal+64Escx;break;
        case 9:co=174Escx+y canal+64Escx;break;
    }
    break;
case 3:
    co=604Escx;x=40;des=1;
    }
switch (op){
case 1: if(L[indr] !=0){
        cein = co;pcein = des;
    }
    else {
        cein =0;pcein = 0;
    }
    break;
case 2: if(L[indr] !=0){
        ceosc = co; pceosc = des;
    }
    else {
        ceosc = 0; pceosc = 0;
    }
    break;
case 3: if(L[indr] !=0){
        cel = co; pcel = des ;
    }
    else {
        cel = 0; pcel = 0 ;
    }
    break;
case 4: if(L[indr] !=0){
        ce2 = co; pce2 = des;
    }

```

```

        else { ce2 = 0; pce2 = 0;
              }
        break;
    case 5: if(L[indr] != 0){
            ces = co; pces = des;
        }
        else { ces = 0; pces = 0;
              }
        break;
    case 6: if(L[indr] != 0){
            ced = co; pced = des;
        }
        else { ced = 0; pced = 0;
              }
        break;
    case 7: if(L[indr] != 0){
            cein2 = co; pcein2 = des;
        }
        else { cein2 = 0; pcein2 = 0;
              }
        break;
    case 8: if(L[indr] != 0){
            ceosc2 = co; pceosc2 = des;
        }
        else { ceosc2 = 0; pceosc2 = 0;
              }
        break;
    case 9: if(L[indr] != 0){
            ceo = co; pceo = des;
        }
        else { ceo = 0; pceo = 0;
              }
        break;
    case 10: if(L[indr] != 0){
            ce3 = co; pce3 = des;
        }
        else { ce3 = 0; pce3 = 0;
              }
        break;
    case 11: if(L[indr] != 0){
            ce4 = co; pce4 = des;
        }
        else { ce4 = 0; pce4 = 0;
              }
        break;
    case 12: if(L[indr] != 0){
            ce5 = co; pce5 = des;
        }
        else { ce5 = 0; pce5 = 0;
              }
        break;
    }
    if (((cein != 0) || (ceosc != 0) || (cein2 != 0) || (ceosc2 != 0) ||
        (ceo != 0) || (ced != 0) || (ces != 0) || (ce1 != 0) ||
        (ce2 != 0) || (ce3 != 0) || (ce4 != 0) || (ce5 != 0))) graf = 1;
    else
        graf=0;
    if(graf == 0 && (des==317 || op ==317))mensaje(2021Escy,43,0,23, " E R R O R ! Ningun grafico ha sido
seleccionado *");
    while((des != 317 && op != 317) || graf == 0);
    if(qq==1){
        pl = linea;
    }

```

```

x = 124*Escr;y = 283*Escr;
setviewport(0,0,MaxX,MaxY,1);
setviewport(0,284*Escr,MaxX,MaxY,1);
clearviewport();
setviewport(0,0,MaxX,MaxY,1);
setlinestyle (0, 0, NORM_WIDTH);
settextjustify( LEFT_TEXT, TOP_TEXT );
settextstyle (DEFAULT_FONT, HORIZ_DJR, 1);
setfillstyle (SOLID_FILL, 0);
setcolor(getmaxcolor());
rectangle(80,284*Escr,MaxX-80*Escr,MaxY-45*Escr);
qprintf (x, y,2, "ESCALA VERTICAL? se recomienda < %d",ycanal);
ayuda_linea(68,25,1);
op = qread (500*Escr, 290*Escr, linea,0, 10, op,25,2);
pl++;x = atoi (pl);
}
KK1=x*Escr;KK2=x*Escr;KK3=x*Escr;KK4=x*Escr;KK5=x*Escr;KK6=x*Escr;
KK13=x*Escr;KK14=x*Escr;KK15=x*Escr;KK7=x*Escr;KK8=x*Escr;KK9=x*Escr;
if(qg==1)cleardevice();
}

/*INIT: Inicializa las constantes de control de la simulacion
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void init(void)
{
int i,j;
nc=N2*ncb;
if(nbucle==7 !! np11==4)nc=N2*ncb/2;
XVCO0=0;CCCO=1;OO=OO1;
if(OO<PI/2 && OO>=0)
CCCO=-1;
if(OO>PI/2 && OO<=PI)
CCCO=-1;
if(OO>PI && OO<=3*PI/2){
OO=2*PI-OO1;CCCO=1;
}
if(OO>PI*3/2 && OO<=2*PI){
OO=2*PI-OO1;CCCO=1;
}
T0=(PI/2-OO)*2/PI;ch=0; l=0; pd[0]=0;
ed=0;eo=0;ein2=0;eosc=0;eosc2=0;el=0;e2=0;e3=0;e4=0;e5=0;es=0;pce3=0;pce4=0;
pced=0;pceo=0;pcein=0;pcein2=0;pceosc=0;pceosc2=0;pcel=0;pce2=0;pce5=0;pces=0;pcsb=0;
for(i=0;i<10;+i){
for(j=0;j<12;+j)
pun[i][j]=0;
}
iniciol=0;inicio=0;M=0;inicioba=0;iniciobb=0;
inicloea=0;inicioeb=0;iniciobandaA=0;iniciobandaB=0;
}

/*INITC: Ipicializa las constantes de seleccion de ondas a simular
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void initc(void)
{
cein=0;ceosc=0;ced=0;ceo=0;cein=0;cein2=0;ceosc=0;
ceosc2=0;cel=0;ce2=0;ce3=0;ce4=0;ce5=0;ces=0;csb=0;
}

```

```

/*MOD4PSK: Modulation 4-PSK
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void mod4psk(void)
{
int y,s1=0,s=0,acc=0;
float dall;
W0=2*PI*f;At=(float)ncb*H2*PI/W0/N;y=H2*2;
if(y !=0)H2+=1;
pcc=2*H/H2;nc=ncb*H2/2;
for(m1=0;m1<H2/2;+m1){
i=0;
if(in[nbb]==0){
++nbb;
if(in[nbb]==0) dall=3*PI/4;
else dall=PI/4;
}
else {
++nbb;
if(in[nbb]==0)dall=5*PI/4;
else dall=7*PI/4;
}
if(nbb<=H2r)
do{
ein[acc]=cos(dall)*cos(W0*t)-sin(dall)*sin(W0*t);
++n;+acc;t=At;
}while(a<pcc);
else
do{
ein[acc]=0;
++n;+acc;
}while(a<pcc);
ppbit++;
if(ppbit!=0 && ppbit!=ppbit)--nbb;
if(ppbit==0 ;; (ppbit!=0 && ppbit==ppbit)){
++nbb;ppbit=0;
}
s=0;t=0;
}
O01=acos(ein[0])+O01;
}

```

```

/*MOD2PSK: Modulation 2-PSK
# Parametros :Ninguno
# Retorno : Ninguno
#####/
void mod2psk(void)
{
int m1=0,s=0,xacc=0;
t=0;W0=2*PI*f;At=(float)ncb*H2*2*PI/W0/N;pcc=H/H2;nc=ncb*H2;
for(m1=0;m1<H2;+m1){
if(nbb<H2r)
do{
ein[xacc]=cos(W0*t);
if(in[nbb]==0) ein[xacc]=-1*ein[xacc];
++n;+xacc;t=At;
}while(a<pcc);
else {
do{
ein[xacc]=0;+n;+xacc;
}while(a<pcc);
}
}
}

```

```

    }
    pbit++;
    if(pbit==0 || (pbit!=0 && pbit==pbit)){
        ++nbb;pbit=0;
    }
    m=0;t=0;
}

/*GREAD:Simula ventana de texto en modo grafico
* Parametros: x.- coordenada X de inicio del recuadro
*             y.- coordenada Y de inicio del recuadro
*             str.-mensaje
*             indicador.- Tipo de entrada que se acepta(numeros,letras,teclas de funciones,etc)
* Retorno :   Indicador [Datos acaptados? (S/N) o el codigo de la tecla de funcion]
#####
Código se repite en ENTRADAS.C (Módulo ENTRADA.EXE)

/*NUMERO_CICLOS:Determina el # de ciclos por pantalla de resultados
* Parametros: Ninguno
* Retorno :   # de ciclos por pantalla de resultados
#####
int numero_ciclos(void)
{
    int nciclos,y,bit_extra;
    float x;
    if(nbucle==7 || npl==4)ncb=ncb/2;
    if(nc>13){
        pbit=0;bit_extra=N2;
        if(ncb>14){
            pbit=ncb/10;x=ncb;
            while(x>10)x-=10;
            if(x!=0)pbit++;
            N2=1;
            if(nbucle==7 || npl==4)mensaje(222+Escy,62,0,73, "#ciclos por estado >14, se simula
c/estado en mas de una pantalla");
            else mensaje(222+Escy,62,0,73,"# de ciclos por bit >14, se simula c/bit en mas de una
pantalla");
        }
        if(ncb>8 && ncb<=14){
            N2=1;
            if(nbucle==7 || npl==4)mensaje(222+Escy,56,0,73, "ADVERTENCIA! # de ciclos por estado>10,
se simula estado por estado");
            else mensaje(222+Escy,56,0,73,"ADVERTENCIA! Numero de ciclos por bit >10, se simula
bit por bit");
        }
        if(ncb >= 5 && ncb<8 && N2 > 2){
            N2=2;
            if(nbucle==7 || npl==4)mensaje(222+Escy,57,0,70, "ADVERTENCIA! # de ciclos por estado >4,
se simula de 2 en 2 estados");
            else mensaje(222+Escy,57,0,70,"ADVERTENCIA! Numero de ciclos por bit >4, se simula de
2 en 2 bits");
        }
        if(ncb>=3 && ncb<5 && N2 > 3){
            if(nbucle==7 || npl==4)mensaje(222+Escy,57,0,68, "ADVERTENCIA! # de ciclos por estado >3,
se simula de 3 en 3 estados");
            else mensaje(222+Escy,57,0,68,"ADVERTENCIA! Numero de ciclos por bit >3, se simula de
3 en 3 bits");
            N2=3;
        }
        if(ncb >=1.2 && ncb<3 && N2 > 4){

```

```

        if(nbucle==7 !! npl==4)mensaje(222Escy,50,0,68, "ADVERTENCIA! Numero de ciclos >14, se
            simula de 4 en 4 estados ");
        else
            mensaje(222Escy,50,0,68,"ADVERTENCIA! Numero de ciclos >14, se simula de 4 en 4 bits ");
            N2=4;
        }
        if(ncb < 2 && N2 >10){
            if(nbucle==7 !! npl==4)mensaje(222Escy,45,0,68, "ADVERTENCIA! Numero de ciclos >14, se
                simula de 4 en 4 estados ");
            else
                mensaje(222Escy,45,0,68,"ADVERTENCIA! Numero de ciclos >14, se simula de 4 en 4 bits ");
                N2=10;
            }
            if(pbit !=0){
                nciclos=pbit*N2r;
                ncb=10;
            }
            if(pbit==0)
                nciclos=bit_extra/N2;
            y=bit_extra % N2;
            if(y != 0)nciclos=nciclos+1;
        }
        else nciclos=1;
        if(nbucle==7)ncb=ncbr;
        return nciclos;
    }
    /*BEEP:Control de sonido
    * Parametros: f.- ciclos de sonido
    *             d.- tiempo de sonido
    * Retorno : Ninguno
    *****/
    void beep (int f, int d)
    {
        sound (f);
        delay (d);
        nosound();
    }
    /*MOV:Movimiento del cursor en las ventanas de texto
    * Parametros: x.- coordenada X de inicio del caracter
    *             yy.-coordenada Y de inicio del caracter
    *             x2.-coordenada X de fin del caracter
    *             lor.- Indicador de desplazamiento hacia atras
    * Retorno : Ninguno
    *****/
    Código se repite en ENRADAS.C (Módulo ENTRADA.EXE)

    /*
    Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.
    LECTURA.C
    Archivo que controla el movimiento del cursor en la eleccion de la onda a graficar
    */
    #include <conio.h>
    #include <dos.h>
    #include <stdlib.h>
    #include <math.h>
    #include "proyfil.h"
    #include <graphics.h>
    #include <stdio.h>
    extern float MaxX,MaxY,Escx,Escy;
    extern int gg,nbucle,tbn[13],tppcuadro;

```



```
int p2,l[13],indr,des;
```

```
/*ONDAREAD_BUC2: Controla el movimiento del cursor en la eleccion de la onda a graficar
```

```
* Parametros : x .- coordenada X base
```

```
* y .- coordenada Y base
```

```
* Retorno : # de onda seleccionada
```

```
#####/
```

```
int ondaread_BUC2(float x, float y)
```

```
{
```

```
float dirx,diry;
```

```
float x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12;
```

```
float y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12;
```

```
int ch2,espx,dxg,dyg,n,numeros[13],indr;
```

```
switch(nbucle){
```

```
case 2: n=10;
```

```
numeros[1] = 1;numeros[2] = 2;
```

```
numeros[3] = 3;numeros[4] = 4;
```

```
numeros[5] = 5;numeros[6] = 7;
```

```
numeros[7] = 9;numeros[8] = 6;
```

```
numeros[9] = 8;
```

```
x1 = x+Escx*195; y1 =-y+Escy*120;
```

```
x2 = x+Escx*187;y2 =-y+Escy*106;
```

```
x3 = x+Escx*262;y3 =-y+Escy*85;
```

```
x4 = x+Escx*370;y4 =-y+Escy*85;
```

```
x5 = x+Escx*565;y5 =-y+Escy*85;
```

```
x6 = x+Escx*305;y6 =-y+Escy*120;
```

```
x7 = x+Escx*363;y7 =-y+Escy*195;
```

```
x8 = x+Escx*260;y8 =-y+Escy*195;
```

```
x9 = x+Escx*185;y9 =-y+Escy*187;
```

```
break;
```

```
case 5:n=10;
```

```
numeros[1] = 1;numeros[2] = 2;
```

```
numeros[3] = 3;numeros[4] = 4;
```

```
numeros[5] = 5;numeros[6] = 7;
```

```
numeros[7] = 9;numeros[8] = 6;
```

```
numeros[9] = 8;
```

```
x1 = x+Escx*170; y1 =-y+Escy*145;
```

```
x2 = x+Escx*140;y2 =-y+Escy*118;
```

```
x3 = x+Escx*213;y3 =-y+Escy*195;
```

```
x4 = x+Escx*310;y4 =-y+Escy*195;
```

```
x5 = x+Escx*415;y5 =-y+Escy*195;
```

```
x6 = x+Escx*310;y6 =-y+Escy*131;
```

```
x7 = x+Escx*380;y7 =-y+Escy*192;
```

```
x8 = x+Escx*210;y8 =-y+Escy*192;
```

```
x9 = x+Escx*142;y9 =-y+Escy*192;
```

```
break;
```

```
case 6:n=11;
```

```
numeros[1] = 1;numeros[2] = 2;
```

```
numeros[3] = 3;numeros[4] = 4;
```

```
numeros[5] = 5;numeros[6] = 7;
```

```
numeros[7] = 10;numeros[8] = 9;
```

```
numeros[9] = 6;numeros[10] = 8;
```

```
x1 = x+Escx*172; y1 =-y+Escy*145;
```

```
x2 = x+Escx*147;y2 =-y+Escy*115;
```

```
x3 = x+Escx*210;y3 =-y+Escy*195;
```

```
x4 = x+Escx*310;y4 =-y+Escy*195;
```

```
x5 = x+Escx*415;y5 =-y+Escy*195;
```

```
x6 = x+Escx*310;y6 =-y+Escy*130;
```

```
x7 = x+Escx*415;y7 =-y+Escy*195;
```

```
x8 = x+Escx*310;y8 =-y+Escy*195;
```

```
x9 = x+Escx*212;y9 =-y+Escy*195;
```

```
x10 = x+Escx*143;y10 =-y+Escy*192;
```

```

        break;
case 3: n=8;
    numeros[1] = 1;numeros[2] = 2;
    numeros[3] = 3;numeros[4] = 4;
    numeros[5] = 5;numeros[6] = 7;
    numeros[7] = 6;
    x1 = x+Escrt110;y1 =-y+Escyt105;
    x2 = x+Escrt195;y2 =-y+Escyt135;
    x3 = x+Escrt212;y3 =-y+Escyt105;
    x4 = x+Escrt310;y4 =-y+Escyt105;
    x5 = x+Escrt415;y5 =-y+Escyt105;
    x6 = x+Escrt380;y6 =-y+Escyt195;
    x7 = x+Escrt205;y7 =-y+Escyt182;
    break;
case 4: n=8;
    numeros[1] = 1;numeros[2] = 2;
    numeros[3] = 3;numeros[4] = 4;
    numeros[5] = 5;numeros[6] = 7;
    numeros[7] = 6;
    x1 = x+Escrt162;y1 =-y+Escyt105;
    x2 = x+Escrt195;y2 =-y+Escyt130;
    x3 = x+Escrt213;y3 =-y+Escyt105;
    x4 = x+Escrt310;y4 =-y+Escyt105;
    x5 = x+Escrt415;y5 =-y+Escyt105;
    x6 = x+Escrt160;y6 =-y+Escyt150;
    x7 = x+Escrt245;y7 =-y+Escyt195;
    break;
case 1:n=10;
    numeros[1] = 1;numeros[2] = 2;
    numeros[3] = 3;numeros[4] = 4;
    numeros[5] = 5;numeros[6] = 9;
    numeros[7] = 6;numeros[8] = 8;
    numeros[9] = 7;
    x1 = x+Escrt130;y1 =-y+Escyt185;
    x2 = x+Escrt227;y2 =-y+Escyt105;
    x3 = x+Escrt304;y3 =-y+Escyt185;
    x4 = x+Escrt428;y4 =-y+Escyt185;
    x5 = x+Escrt152;y5 =-y+Escyt185;
    x6 = x+Escrt330;y6 =-y+Escyt118;
    x7 = x+Escrt320;y7 =-y+Escyt186;
    x8 = x+Escrt227;y8 =-y+Escyt175;
    x9 = x+Escrt210;y9 =-y+Escyt186;
    break;
case 7:n=13;
    numeros[1] = 1;numeros[2] = 2;
    numeros[3] = 3;numeros[4] = 7;
    numeros[5] = 4;numeros[6] = 5;
    numeros[7] = 12;numeros[8] = 11;
    numeros[9] = 10;numeros[10] = 9;
    numeros[11] = 6;numeros[12] = 8;
    x1 = x+Escrt175;y1 =-y+Escyt145;
    x2 = x+Escrt195;y2 =-y+Escyt113;
    x3 = x+Escrt210;y3 =-y+Escyt190;
    x4 = x+Escrt270;y4 =-y+Escyt135;
    x5 = x+Escrt310;y5 =-y+Escyt190;
    x6 = x+Escrt415;y6 =-y+Escyt190;
    x7 = x+Escrt500;y7 =-y+Escyt113;
    x8 = x+Escrt500;y8 =-y+Escyt170;
    x9 = x+Escrt110;y9 =-y+Escyt195;
    x10 = x+Escrt315;y10 =-y+Escyt195;
    x11 = x+Escrt212;y11 =-y+Escyt195;
    x12 = x+Escrt145;y12 =-y+Escyt192;

```

```

        break;
    case 0: n=3;
        numeros[1] = 1;numeros[2] = 6;
        numeros[3] = 9;numeros[4] = 2;
        x1 = x+Escx1165;y1 =-y+Escy195;
        x2 = x+Escx1350;y2 =-y+Escy195;
        x3 = x+Escx1415;y3 =-y+Escy1173;
        x4 = x+Escx1220;y4 =-y+Escy1145;
    }
    if(des != 0 && L[indr] != 0 ){
        switch (indr)
        {
            case 1:dirx = x1; diry = y1;break;
            case 2:dirx = x2; diry = y2;break;
            case 3:dirx = x3; diry = y3;break;
            case 4:dirx = x4; diry = y4;break;
            case 5:dirx = x5; diry = y5;break;
            case 6:dirx = x6; diry = y6;break;
            case 7:dirx = x7; diry = y7;break;
            case 8:dirx = x8; diry = y8;break;
            case 9:dirx = x9; diry = y9;break;
            case 10:dirx = x10; diry = y10;break;
            case 11:dirx = x11; diry = y11;break;
            case 12:dirx = x12; diry = y12;break;
        }
        settxtstyle(SMALL_FONT,HORIZ_DIR,4);
        diry+=textheight("H")+10-3;diry+=textheight("H")/2;
        dirx-=textwidth("H");dxq=dirx;dyg=diry;
        if (gg==1)
            gprintf(&dxq,&dyg,1,"Czd",des);
        else
            gprintf(&dxq,&dyg,1,"OH");
        settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    }
    settxtstyle(SMALL_FONT,HORIZ_DIR,4);
    espr=4+textwidth("H")+0;
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    selfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    settextjustify( LEFT_TEXT, TOP_TEXT );
    ind1=1;
    if(p2==0)        p2=1;
    else            ind1=0;
    ch2=328;
    do
    {
        if (ch2 == 328 ||ch2 == 336 || ch2 == 331 ||ch2 == 333)
            switch (indr)
            {
                case 1:cuadro(x1,y1);break;
                case 2:cuadro(x2,y2);break;
                case 3:cuadro(x3,y3);break;
                case 4:cuadro(x4,y4);break;
                case 5:cuadro(x5,y5);break;
                case 6:cuadro(x6,y6);break;
                case 7:cuadro(x7,y7);break;
                case 8:cuadro(x8,y8);break;
                case 9:cuadro(x9,y9);break;
                case 10:cuadro(x10,y10);break;
                case 11:cuadro(x11,y11);break;
                case 12:cuadro(x12,y12);break;
            }
    }

```

```

ind1=p2;
ch2 = tecla();
if(ch2 == 321 && (gg==2 || gg==3)) GrPrint();
if (ch2 == 328 || ch2 == 324 || ch2 == 336 || ch2 == 331 || ch2 == 333 || (ch2 == 317 && (gg==1 || gg==3)) ||
    ch2==13 ||(ch2 == 27 && (gg==2 || gg==3)) || (( ch2 == 318 ) && gg==2) || ((ch2 == 323 ) && gg==2) )
{
    switch (ind1)
    {
        case 1: pulimage(x1-espx/2,y1-4,ppcuadro, COPY_PUT);break;
        case 2: pulimage(x2-espx/2,y2-4,ppcuadro, COPY_PUT);break;
        case 3: pulimage(x3-espx/2,y3-4,ppcuadro, COPY_PUT);break;
        case 4: pulimage(x4-espx/2,y4-4,ppcuadro, COPY_PUT);break;
        case 5: pulimage(x5-espx/2,y5-4,ppcuadro, COPY_PUT);break;
        case 6: pulimage(x6-espx/2,y6-4,ppcuadro, COPY_PUT);break;
        case 7: pulimage(x7-espx/2,y7-4,ppcuadro, COPY_PUT);break;
        case 8: pulimage(x8-espx/2,y8-4,ppcuadro, COPY_PUT);break;
        case 9: pulimage(x9-espx/2,y9-4,ppcuadro, COPY_PUT);break;
        case 10: pulimage(x10-espx/2,y10-4,ppcuadro, COPY_PUT);break;
        case 11: pulimage(x11-espx/2,y11-4,ppcuadro, COPY_PUT);break;
        case 12: pulimage(x12-espx/2,y12-4,ppcuadro, COPY_PUT);break;
    }
    if(ch2 == 328 ||ch2 == 331){
        ind1--;
        if(ind1<1)
            ind1=n-1;
    }
    if( ch2 == 336 || ch2 == 333){
        if (ind1 == (n-1))
            ind1 = 1;
        else ind1++;
    }
    p2=ind1;
}
} while (ch2 != '\r' && ch2 != 324 && (ch2 != '\x1b' || gg==1) && !(ch2 ==317 && (gg==1 || gg==3)) &&
    !(( ch2 == 318 ) && gg==2) && !(ch2 == 323 ) && gg==2) );
switch (ind1)
{
    case 1: cuadro(x1,y1);break;
    case 2: cuadro(x2,y2);break;
    case 3: cuadro(x3,y3);break;
    case 4: cuadro(x4,y4);break;
    case 5: cuadro(x5,y5);break;
    case 6: cuadro(x6,y6);break;
    case 7: cuadro(x7,y7);break;
    case 8: cuadro(x8,y8);break;
    case 9: cuadro(x9,y9);break;
    case 10: cuadro(x10,y10);break;
    case 11: cuadro(x11,y11);break;
    case 12: cuadro(x12,y12);break;
}
if(ch2==13 && (gg==1 || gg==3))
switch (ind1)
{
    case 1:if(L[1]==0) L[1]=1;
        else L[1]=0;
        bon(x1,y1,L[1],ind1);
        break;
    case 2;if(L[2]==0) L[2]=1;
        else L[2]=0;
        bon(x2,y2,L[2],ind1);
        break;
    case 3;if(L[3]==0) L[3]=1;

```

```

        else      L[3]=0;
        box(x3,y3,L[3],ind1);
        break;
    case 4:if(L[4]==0) L[4]=1;
        else      L[4]=0;
        box(x4,y4,L[4],ind1);
        break;
    case 5:if(L[5]==0) L[5]=1;
        else      L[5]=0;
        box(x5,y5,L[5],ind1);
        break;
    case 6:if(L[6]==0) L[6]=1;
        else      L[6]=0;
        box(x6,y6,L[6],ind1);
        break;
    case 7:if(L[7]==0) L[7]=1;
        else      L[7]=0;
        box(x7,y7,L[7],ind1);
        break;
    case 8:if(L[8]==0) L[8]=1;
        else      L[8]=0;
        box(x8,y8,L[8],ind1);
        break;
    case 9:if(L[9]==0) L[9]=1;
        else      L[9]=0;
        box(x9,y9,L[9],ind1);
        break;
    case 10:if(L[10]==0) L[10]=1;
        else      L[10]=0;
        box(x10,y10,L[10],ind1);
        break;
    case 11:if(L[11]==0) L[11]=1;
        else      L[11]=0;
        box(x11,y11,L[11],ind1);
        break;
    case 12:if(L[12]==0) L[12]=1;
        else      L[12]=0;
        box(x12,y12,L[12],ind1);
        break;
    }
    indr=ind1;
    if(ch2 == 318 || ch2 == 324 )
        return ch2;
    if(ch2 == 317 && (gg==1 || gg==3))return ch2;
    if((ch2 == 323 ) && gg==2)return ch2;
    if ( ch2 == '\x1b')
        return ch2;
    else
        return numeros[ind1];
}

/*BOX: Presenta un recuadro en la salida a graficar seleccionada
* Parametros : x ,- coordenada X base
*              L ,- estado [seleccionado?(S/N)]
*              y ,- coordenada Y base
*              ind1 ,- # de onda en cuestion
* Retorno : Ninguno
*/
void box(float x1,float y1,int L,int ind1)
{
    float espxb,espyb;
    settextstyle(SMALL_FONT,HORIZ_DIR,4);

```

```

y1=textheight("H")+10-3;
espxb=4*textwidth("H")+4;
espyb=textheight("H")+10;
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setviewport(0,0,MaxX,MaxY,1);
switch(L){
  case 0: setviewport(x1+espxb/2,y1,x1+espxb/2,y1+espyb,1);
          clearviewport();
          setviewport(0,0,MaxX,MaxY,1);
          putimage (x1-espxb/2,y1,ba[ind1],COPY_PUT);
          break;
  case 1: getimage (x1-espxb/2,y1,x1+espxb/2,y1+espyb,ba[ind1]);
          setfillstyle (SOLID_FILL, 0);
          bar (x1-espxb/2,y1,x1+espxb/2,y1+espyb);
          setcolor (getmaxcolor());
          rectangle (x1-espxb/2,y1,x1+espxb/2,y1+espyb);
          setfillstyle( SOLID_FILL, YELLOW);
          /*setfillstyle( INTERLEAVE_FILL, YELLOW );*/
          bar (x1-espxb/2+2,y1+2,x1+espxb/2-2,y1+espyb-2);
          setcolor (BLACK);
          setfillstyle( SOLID_FILL, BLACK);
          filloval( x1,y1+espyb/2,textwidth("H")+2,textheight("H")/2+2);
          setcolor (0);
}
setviewport(0,0,MaxX,MaxY,1);
}

```

/\*CUADRO: forma el cursor en la eleccion de la onda a graficar

† Parametros : x .- coordenada X base

† y .- coordenada Y base

† Retorno : Ninguno

#####

void cuadro(float x1,float y1)

```

{
  int espx,espy;
  size_t sscuadro;
  settextstyle(SMALL_FONT,HORIZ_DIR,4);
  espx=4*textwidth("H")+8;
  espy=textheight("H")+10;
  settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
  getimage (x1-espx/2,y1-4,x1+espx/2,y1+espy-4,ppcuadro);
  setfillstyle (SOLID_FILL, 0);
  bar (x1-espx/2,y1-4,x1+espx/2,y1+espy-4);
  setcolor (getmaxcolor());
  rectangle (x1-espx/2+1,y1-4,x1+espx/2-1,y1+espy-4);
  setfillstyle (SOLID_FILL, getmaxcolor());
  bar (x1-espx/2+4,y1-1,x1+espx/2-4,y1+espy-7);
  putimage (x1-espx/2,y1-4,ppcuadro,XOR_PUT);
  setcolor (0);
}

```

/\*

Escuela Politécnica Nacional-Facultad de Ingeniería Eléctrica.

FILTRO.C

Archivo que simula filtros pasabajos, filtros pasabanda y  
duplicadores de frecuencia

\*/

```

#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include "proyfil.h"

```

```

#include <graphics.h>
#include <stdio.h>
float AfpeA,AfpeB,AfreA,AfreB,AfpaA,AfpaB,AfpaB,AfpaB,fpeA,fpeB,freA,freB,fpaA,fpaB,frbA,frbB;
float Aho,AA0[6],ABB[6],AC[6],Bho,BAM[6],BP[6],BC[6],DA0[5],DA1[5],DA2[5],DB0[5],DB1[5],DB2[5];
float Bha,Bhb,fsea,fseb,fsba,fsbh,fsbna,fsbanb,AfpanA,AfpanB,AfpanA,AfpanB,pra,prh,Bs0,A=0;
float frbanA,frbanB,AM,BK,Afo,Bfo,fsbna,fsbanb,ncb,ff,nc;
float Ah0[5],Ah1[5],Ah2[5],Ah3[5],Ah4[5],AH0[5],AH1[5],AH2[5],AH3[5],AH4[5];
float Bh0[5],Bh1[5],Bh2[5],Bh3[5],Bh4[5],BH0[5],BH1[5],BH2[5],BH3[5],BH4[5];
float Aj2,Aj0,Aj1,Aj2,Aj3,Aj4,Bj2,Bj0,Bj1,Bj2,Bj3,Bj4,BAho,BBho,Bfo,Bpend,Bini,Bfp,Bfr,EPs,OO1;
float Ay0[7][3],By0[7][3],BUAy0[7][3],BUBy0[7][3],BAy0[7][5],BBy0[7][5];
int iniciobandaA,iniciobandaB,indiceg,inicioea,inicioeb,inicioea,iniciobb;
int Dn,Ann,Bnn,An,Bn,DAnn,DAn,DRnn,PRo,BAnn,BAn,BBnn,BBn,H2,in[20],cf1,cf2;
extern int nbucle;

/*DOBLADOR:Duplicador de Frecuencia
# Parametros : y .- valor de la onda que sera duplicada su frecuencia
# caso .- indicador del duplicador que debe actuar
# Retorno : valor de la onda de doble frecuencia
#####/
float doblador(float y,int caso)
{
float adob,bdob,dob;
if(y<0)y=-1*y;
if(caso==1){
adob=filtrobanda(y,1);
}
if(caso==2){
bdob=filtrobanda(y,2);
}
if(caso==1)
dob=adob;
else
dob=bdob;
return dob*PI/2*.9;
}

/*FILTRO:Filtro ELIPTICO pasabajos
# Parametros : err .- valor de la onda que sera filtrada
# dc .- indicador del filtro que debe actuar (contiene dos filtros)
# Retorno : valor de la onda filtrada
#####/
float filtro(float err,int dc)
{
int A1a,A1l,r;
int i,ii;
float y,fs1,Ay,By,Aho,Boo,y1,y2;
switch(dc){
case 1:
fs1=fsea/2;A1a=2*fs1;A1l=pow(2*fs1,2);
r=Ann==0?An/2:(An-1)/2;
if(Ann==1)r=0;
if(inicioea==0){
for(i=0;i<7;i++){
Ay0[i][1]=0;Ay0[i][0]=0;
}
inicioea=1;
}
Ay0[0][2]=err;
if(Ann==1){
Ay0[6][2]=1/(A1a+As0)*((A1a-As0)*Ay0[6][1]+Aho*(Ay0[0][2]+Ay0[0][1]));
Ay=Ay0[6][2];goto Afin0;
}
}
}

```

```

for(i=1;i<r+1;++i){
    Aho0= i==1?Aho:1;
    y1=2*(A11-AC[i])*Ay0[i][1]+(A00[i]*A10-At1-AC[i])*Ay0[i][0];
    y2=Aho0*(A11+A00[i])*Ay0[i-1][2]+2*(A00[i]-At1)*Ay0[i-1][1]+(At1+A00[i])*Ay0[i-1][0];
    Ay0[i][2]=1/(A11+A00[i])*A10+AC[i]*(y1+y2);
    Ay=Ay0[i][2];
}
if(Aho!=0){
    Ay0[6][2]=1/(A10+A50)*((A10-A50)*Ay0[6][1]+Ay0[i-1][2]+Ay0[i-1][1]);
    Ay=Ay0[6][2];
}
for(i=1;i<r+1;++i){
    Ay0[i][0]=Ay0[i][1];Ay0[i][1]=Ay0[i][2];
}
Afin0: Ay0[6][0]=Ay0[6][1];Ay0[6][1]=Ay0[6][2];
    Ay0[0][0]=Ay0[0][1];Ay0[0][1]=Ay0[0][2];
break;
case 2:
fsl=fseb/2;A10=2*fsl;A11=pow(2*fsl,2);
r=Bno==0?Bn/2:(Bn-1)/2;
if(Bn==1)r=0;
if(inicioeb==0){
    for(i=0;i<7;++i){
        By0[i][1]=0;By0[i][0]=0;
    }
    inicioeb=1;
}
By0[0][2]=err;
if(Bn==1){
    By0[6][2]=1/(A10+B50)*((A10-B50)*By0[6][1]+Bho*(By0[0][2]+By0[0][1]));
    By=By0[6][2];goto Bfin0;
}
for(i=1;i<r+1;++i){
    Bho0= i==1?Bho:1;
    By0[i][2]=1/(A11+BB[i]*A10+BC[i])*((2*(A11-BC[i])*By0[i][1]+(BB[i]*A10-At1-BC[i])*By0[i][0]+
    Bho0*(A11+B00[i])*By0[i-1][2]+2*(B00[i]-At1)*By0[i-1][1]+(At1+B00[i])*By0[i-1][0]));
    By=By0[i][2];
}
if(Bho!=0){
    By0[6][2]=1/(A10+B50)*((A10-B50)*By0[6][1]+By0[i-1][2]+By0[i-1][1]);
    By=By0[6][2];
}
for(i=1;i<r+1;++i){
    By0[i][0]=By0[i][1];By0[i][1]=By0[i][2];
}
Bfin0: By0[6][0]=By0[6][1];By0[6][1]=By0[6][2];
    By0[0][0]=By0[0][1];By0[0][1]=By0[0][2];
break;
}
y=dc==1?Ay:By;
return y;
}

```

/\*FILTR0B: Filtro de Butterworth pasabajos

\* Parametros : err .- valor de la onda que sera filtrada

\* dc .- indicador del filtro que debe actuar (contiene dos filtros)

\* Retorno : valor de la onda filtrada

#####

float fillrob(float err, int dc)

```

{
    int n,i,ii,r;
    float fsl,Ay,By,fillr,Aho0,Bho0;

```



```

switch(dc){
  case 1:  fs1=fsba; n=2*fs1;
          r=DA0n==0?DA0n/2:(DA0n-1)/2;
          if(DA0n==1)r=0;
          if(inicioba==0){
              for(i=0;i<7;i++){
                  BUAY0[i][1]=0;BUAY0[i][0]=0;
              }
              inicioba=i;
          }
          BUAY0[0][2]=err;
          if(DA0n!=1){
              BUAY0[6][2]=1/(1+pra)*((1-pra)*BUAY0[6][1]+Dh0*(BUAY0[0][2]+BUAY0[0][1]));
              Ay=BUAY0[6][2];goto AFin0;
          }
          for(i=1;i<r+1;i++){
              Ahoo= i==1?Dha:1;
              BUAY0[i][2]=1/DA0[i-1]*(Ahoo*(BUAY0[i-1][2]+2*BUAY0[i-1][1]+
              BUAY0[i-1][0])-DB1[i-1]*BUAY0[i][1]-DA2[i-1]*BUAY0[i][0]);
              Ay=BUAY0[i][2];
          }
          if(DA0n!=0){
              BUAY0[6][2]=1/(1+pra)*((1-pra)*BUAY0[6][1]+BUAY0[0][2]+BUAY0[0][1]);
              Ay=BUAY0[6][2];
          }
          for(i=1;i<r+1;i++){
              BUAY0[i][0]=BUAY0[i][1];BUAY0[i][1]=BUAY0[i][2];
          }
          AFin0: BUAY0[6][0]=BUAY0[6][1];BUAY0[6][1]=BUAY0[6][2];
                 BUAY0[0][0]=BUAY0[0][1];BUAY0[0][1]=BUAY0[0][2];
          break;
  case 2:  fs1=fsbb; n=2*fs1;
          r=DB0n==0?DB0n/2:(DB0n-1)/2;
          if(DB0n==1)r=0;
          if(iniciobb==0){
              for(i=0;i<7;i++){
                  BUBy0[i][1]=0;BUBy0[i][0]=0;
              }
              iniciobb=i;
          }
          BUBy0[0][2]=err;
          if(DB0n==1){
              BUBy0[6][2]=1/(1+prb)*((1-prb)*BUBy0[6][1]+Dhb*(BUBy0[0][2]+BUBy0[0][1]));
              By=BUBy0[6][2];goto BFin0;
          }
          for(i=1;i<r+1;i++){
              Bhoo= i==1?Dhb:1;
              BUBy0[i][2]=1/DB0[i-1]*(Bhoo*(BUBy0[i-1][2]+2*BUBy0[i-1][1]+
              BUBy0[i-1][0])-DB1[i-1]*BUBy0[i][1]-DB2[i-1]*BUBy0[i][0]);
              By=BUBy0[i][2];
          }
          if(DB0n!=0){
              BUBy0[6][2]=1/(1+prb)*((1-prb)*BUBy0[6][1]+BUBy0[0][2]+BUBy0[0][1]);
              By=BUBy0[6][2];
          }
          for(i=1;i<r+1;i++){
              BUBy0[i][0]=BUBy0[i][1];BUBy0[i][1]=BUBy0[i][2];
          }
          BFin0: BUBy0[6][0]=BUBy0[6][1];BUBy0[6][1]=BUBy0[6][2];
                 BUBy0[0][0]=BUBy0[0][1];BUBy0[0][1]=BUBy0[0][2];
          }
}

```

```

    if(dc==1) fillr=Ay;
    else fillr=By;
    return fillr;
}

/*FILTROBANDA:Filtro Pasabanda Eliptico
# Parametros :   err  .- valor de la onda que sera filtrada
#               dc   .- indicador del filtro que debe actuar (contiene dos filtros)
# Retorno :     valor de la onda filtrada
#####/
float filtrobanda(float err,int cd)
{
    int n,i,ii,r;
    float Ay,By,fillr,Ahoo,Bhoo,y;
    switch(cd){
        case 1:
            r=BAAn==0?BAAn/2:(BAAn-1)/2;
            if(BAAn==1)r=0;
            if(iniciobandaA==0){
                for(i=0;i<7;i++){
                    BAy0[i][0]=0;BAy0[i][1]=0;
                    BAy0[i][2]=0;BAy0[i][3]=0;
                }
                iniciobandaA=1;
            }
            BAy0[0][4]=err;
            if(BAAn==1){
                BAy0[6][4]=1/AJ0*(Aj2*(BAy0[0][4]-BAy0[0][2])-AJ2*BAy0[6][2]-AJ1*BAy0[6][3]);
                Ay=BAy0[6][4];goto Afin0;
            }
            for(i=1;i<r+1;+i){
                Ahoo= i==1?BAhoo:1;
                BAy0[i][4]=1/AH0[i]*(Ah4[i]*BAy0[i-1][0]+Ah3[i]*BAy0[i-1][1]+Ah2[i]*BAy0[i-1][2]+Ah1[i]*
                BAy0[i-1][3]+Ah0[i]*BAy0[i-1][4]+Ahoo-AH4[i]*BAy0[i][0]-AH3[i]*BAy0[i][1]-
                AH2[i]*BAy0[i][2]-AH1[i]*BAy0[i][3]);
                Ay=BAy0[i][4];
            }
            if(BAAn!=0){
                BAy0[6][4]=1/AJ0*(Aj2*(BAy0[0][4]-BAy0[0][2])-AJ2*BAy0[6][2]-AJ1*BAy0[6][3]);
                Ay=BAy0[6][4];
            }
            for(i=1;i<r+1;+i){
                BAy0[i][0]=BAy0[i][1];BAy0[i][1]=BAy0[i][2];
                BAy0[i][2]=BAy0[i][3];BAy0[i][3]=BAy0[i][4];
            }
            Afin0: BAy0[6][0]=BAy0[6][1];BAy0[6][1]=BAy0[6][2];
                BAy0[6][2]=BAy0[6][3];BAy0[6][3]=BAy0[6][4];
                BAy0[0][0]=BAy0[0][1];BAy0[0][1]=BAy0[0][2];
                BAy0[0][2]=BAy0[0][3];BAy0[0][3]=BAy0[0][4];
                break;
        case 2:
            r=BBAn==0?BBAn/2:(BBAn-1)/2;
            if(BBAn==1)r=0;
            if(iniciobandaB==0){
                for(i=0;i<7;i++){
                    BBy0[i][0]=0;BBy0[i][1]=0;
                    BBy0[i][2]=0;BBy0[i][3]=0;
                }
                iniciobandaB=1;
            }
            BBy0[0][4]=err;
            if(BBAn==1){

```

```

        BBy0[6][4]=1/Bj0*(Bj2*(BBy0[0][4]-BBy0[0][2])-Bj2*BBy0[6][2]-Bj1*BBy0[6][3]);
        By=BBy0[6][4];qolo Bfin0;
    }
    for(i=1;i<r+1;++i){
        Bho0= i-1?RPho;1;
        BBy0[i][4]=1/BH0[i]*((Bh1[i]*BBy0[i-1][0]+Bh3[i]*BBy0[i-1][1]+Bh2[i]*BBy0[i-1][2]+
        Bh1[i]*BBy0[i-1][3]+Bh0[i]*BBy0[i-1][4])*Pho0-BH4[i]*BBy0[i][0]-
        BH3[i]*BBy0[i][1]-BH2[i]*BBy0[i][2]-BH1[i]*BBy0[i][3]);
        By=BBy0[i][4];
    }
    if(BBon!=0){
        BBy0[6][4]=1/Bj0*(Bj2*(BBy0[0][4]-BBy0[0][2])-Bj2*BBy0[6][2]-Bj1*BBy0[6][3]);
        By=BBy0[6][4];
    }
    for(i=1;i<r+1;++i){
        BBy0[i][0]=BBy0[i][1];BBy0[i][1]=BBy0[i][2];
        BBy0[i][2]=BBy0[i][3];BBy0[i][3]=BBy0[i][4];
    }
    Bfin0: BBy0[6][0]=BBy0[6][1];BBy0[6][1]=BBy0[6][2];
        BBy0[6][2]=BBy0[6][3];BBy0[6][3]=BBy0[6][4];
        BBy0[0][0]=BBy0[0][1];BBy0[0][1]=BBy0[0][2];
        BBy0[0][2]=BBy0[0][3];BBy0[0][3]=BBy0[0][4];
    break;
}
y=cd==1?Ay:By;
return y;
}

```

/\*LEER: Lee de ARCHIVOS de enlace las constantes que necesita el archivo  
"SIMULAR.EXE", para poder simular los elementos de los distintos BUCLES.

† Parametros : Ninguno

† Retorno : Ninguno( Todas las constantes son globales)

#####

void leer(void)

```

{
FILE *fp;
fp=fopen("f","rb");
rewind(fp);
fread(&Ann,sizeof(int),1,fp);
fread(&An,sizeof(int),1,fp);
fread(&Bnn,sizeof(int),1,fp);
fread(&Bn,sizeof(int),1,fp);
fread(&DAnn,sizeof(int),1,fp);
fread(&DAn,sizeof(int),1,fp);
fread(&DBnn,sizeof(int),1,fp);
fread(&DBn,sizeof(int),1,fp);
fread(&cf1,sizeof(int),1,fp);
fread(&cf2,sizeof(int),1,fp);
fread(&Aho,sizeof(float),1,fp);
fread(&Dho,sizeof(float),1,fp);
fread(&Dha,sizeof(float),1,fp);
fread(&Bhb,sizeof(float),1,fp);
fread(&As0,sizeof(float),1,fp);
fread(&Bs0,sizeof(float),1,fp);
fread(&pra,sizeof(float),1,fp);
fread(&prb,sizeof(float),1,fp);
fread(&fsea,sizeof(float),1,fp);
fread(&fseb,sizeof(float),1,fp);
fread(&fsba,sizeof(float),1,fp);
fread(&fsbb,sizeof(float),1,fp);
fread(&A0,sizeof(float),10,fp);
fread(&AB0,sizeof(float),10,fp);

```





```

selviewport(dx1+11*Escx,diy1+22*Escy,dx2-11*Escx,diy2-22*Escy,1);
clearviewport();
settextjustify(LEFT_TEXT, TOP_TEXT);
setcolor(15);
switch(caso){
case 20:
    x=10*Escx;y=-7*Escy;
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2,"          NUMERO DE CANALES QUE SE ESTRUCTURA EN LA PANTALLA");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Para observar las respuestas de los distintos bloques del");
    gprintf(&x,&y,2," sistema se puede formar tantos canales en toda la pantalla como");
    gprintf(&x,&y,2," deseé.");
    gprintf(&x,&y,2," Si el número ingresado es mayor a 9 la resolución es");
    gprintf(&x,&y,2," DEFICIENTE razón por la que se ha impuesto un máximo número de");
    gprintf(&x,&y,2," canales de 9.");
    break;
case 21:
    x=10*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"          PRESENTAR DIAGRAMA DE BLOQUES DEL BUCLE          ");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Se da la oportunidad de ver el diagrama de bloques de bucle");
    gprintf(&x,&y,2," en estudio, en caso de que se necesite recordar la secuencia de");
    gprintf(&x,&y,2," de Voltaje creada en el mismo.");
    break;
case 25:
    x=10*Escx;y=-2*Escy;
    gprintf(&x,&y,2,"          ESCALA EN EL EJE VERTICAL DEL GRAFICO          ");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Se pregunta por el valor de ampliación que se hace al valor");
    gprintf(&x,&y,2," de la ONDA original, para conseguir una visualización que permita");
    gprintf(&x,&y,2," distinguir de buena forma las características de los resultados");
    gprintf(&x,&y,2," del BUCLE en estudio.");
    gprintf(&x,&y,2," Se RECOMIENDA un valor que es el máximo que puede ingresar");
    gprintf(&x,&y,2," para EVITAR SUPERPOSICIONES entre canales. ");
    gprintf(&x,&y,2," ");
    gprintf(&x,&y,2," Su valor es ENTERO y menor o igual al que se recomienda.");
    break;
case 23:
    x=10*Escx;y=-4*Escy;
    gprintf(&x,&y,2,"          UBICACION DEL GRAFICO");
    gprintf(&x,&y,2," La ONDA seleccionada, puede ubicarse en cualquier canal");
    gprintf(&x,&y,2," estructurado anteriormente. Ud. decide en donde lo ubica de");
    gprintf(&x,&y,2," acuerdo a su conveniencia.");
    gprintf(&x,&y,2," Si su respuesta es un canal que no existe, se presentara");
    gprintf(&x,&y,2," mensaje de ERROR. ");
    break;
case 500:fora(-100*Escx,88*Escy);break;
case 501:forabuc1(-100*Escx,75*Escy);break;
case 502:forabuc2(-87*Escx,82*Escy);break;
case 503:forabuc3(-50*Escx,88*Escy);break;
case 504:forabuc4(-45*Escx,78*Escy);break;
case 505:forabuc5(-25*Escx,90*Escy);break;
case 506:forabuc6(-25*Escx,90*Escy);break;
case 507:forabuc7(-60*Escx,85*Escy);break;
}
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
Main#window(dx1,diy1,dx2,diy2,1,4,"          A y u d a          ");
getch();
clearviewport();
selviewport(0,0,MaxX,MaxY,1);
putimage(dx1-1,diy1-1,buf,COPY_PUT);

```

```

    _graphfreezes (buf,ta);
    setfillstyle (SOLID_FILL, getmaxcolor());
    setcolor (getmaxcolor());
    settextjustify( LEFT_TEXT, TOP_TEXT );
    changetextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
}

#AYUDA_LINEA:Despliega en pantalla ayuda de acuerdo al caso ( SIN ACCESO AL USUARIO)
#Parametros: num  .- cantidad de caracteres a desplegarse
#            pos  .- posicion del mensaje
#            caso .- indicador de que caso de ayuda se ha solicitado
#Retorno : Ninguno
#####
void ayuda_linea(int num,int caso,int pos){
int x,y,mil,espxl,prn;
float a;
    if(pos==1)a=10*Escy;
    else      a=0;
    mil=(MaxX-24*Escx)/2;
    espxl=textwidth("M");
    x=15*Escx;y=MaxY-30*Escy;
    setviewport(0,0,MaxX,MaxY,1);
    setviewport(5*Escx,MaxY-39*Escy,MaxX-30*Escx,MaxY,1);
    clearviewport();
    setfillstyle(SOLID_FILL,0);
    MainWindow_linea(mil-(num*2)*espxl/2,MaxY-28*Escy-a,mil+(num*2)*espxl/2,MaxY-a,1);
    settextjustify(LEFT_TEXT,TOP_TEXT);
    setcolor(15);
        x=10*Escx;y=7*Escy;
        gprintf(&x,&y,2," Su valor es ENTERO y menor o igual al que se recomienda.");
        break;
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setviewport(0,0,MaxX,MaxY,1);
    if(pos==1){
        rectangle(mil+(num*2)*espxl/2-12*textwidth("M"),MaxY-textheight("M")-4,mil+(num*2)*espxl/2,MaxY);
        outtextxy(mil+(num*2)*espxl/2-11*textwidth("M"),MaxY-textheight("M")/2-4,"[F1] AYUDA");
    }
}

```

## SIMBOLOGIA

<b>AM</b>	: Amplitude Modulation
<b>FM</b>	: Frequency Modulation
<b>PM</b>	: Phase Modulation
<b>ASK</b>	: Amplitude Shift Keying
<b>FSK</b>	: Frequency Shift Keying
<b>PSK</b>	: Phase Shift Keying
<b>OOK</b>	: ON-OFF Keying
<b>VCO</b>	: Oscilador Controlado por Voltaje
<b>FPB</b>	: Filtro Pasabajos
<b>PD</b>	: Comparador de fase
<b>ADC</b>	: Ecuilizador
<b>X2</b>	: Duplicador de frecuencia
<b>X</b>	: Multiplicador
$\Omega$	: Frecuencia Analogica
<b>W</b>	: Frecuencia Digital

- 1 Onda sinusoidal de entrada al bucle
- 2 Onda sinusoidal creada por el V.C.O.
- 3 Onda resultante de multiplicar las ondas 1 y 2
- 4 Onda resultante de filtrar la onda 3 (salida de FPB1)
- 5 Onda resultante de ecualizar la onda 4 (salida de ADC)
- 6 Salida del comparador de fase
- 7 Onda de doble frecuencia de la senal recibida en el bucle con MULTIPLICACION, senal de control del V.C.O. en los bucles de COSTAS y Onda remodulada en los bucles con REMODULACION
- 8 Onda de doble frecuencia del V.C.O. en el bucle con MULTIPLICACION o salida del defasador de 90 grados en los bucles de COSTAS
- 9 Salida del filtro auxiliar del bucle (FPB2) o del brazo en cuadratura para los bucles de COSTAS
- 0 Onda ecualizada en el brazo en cuadratura en los bucles de COSTAS
- a Componente del brazo en cuadratura del bucle de COSTAS para QPSK
- b Componente del brazo en fase del bucle de COSTAS para QPSK



## BIBLIOGRAFIA

- [1] Hedin George y Holmes Jack, "THEORY OF FALSE LOCK IN COSTAS LOOP", IEEE Transactions on Communications, VOL. COM-26, No. 1, Enero 1978
- [2] Simon Marvin, "THE FALSE LOCK PERFORMANCE OF COSTAS LOOP WITH HARD-LIMITED IN PHASE CHANNEL", IEEE Transactions on Communications, VOL. COM-26, No.1, Enero 1978.
- [3] GERHEIM, Albert, "A Synthesis Procedure for 90° Fan Filters", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, VOL. COM-30, No. 12, Diciembre 1983.
- [4] Oetting John, "MODULATION TECHNIQUES FOR DIGITAL RADIO", IEEE Transactions on Communications, VOL. COM-27, No. 12, Diciembre 1979.
- [5] Mischa Schwartz, "TRANSMISION DE INFORMACION Y MODULACION Y RADIO", McGraw-Hill, México, 1966.
- [6] Gardner Floyd, "PHASELOCK TECHNIQUES", Wiley, New York, 1966.
- [7] Feher Kamilo, "DIGITAL COMMUNICATIONS, SATELLITE/EARTH STATION ENGINEERING", Prentice-Hall, Englewood Cliffs, 1983.

- [8] Oppenheim Alan, " DIGITAL SIGNAL PROCESSING ",  
Prentice-Hall, Englewood Cliffs.
- [9] Bellenger Maurice, " TRAITEMENT NUMERIQUE DU SIGNAL  
THEORIE ET PRACTIQUE ", Masson et C.N.E.T.- ENST,  
Paris, 1980.
- [10] Boite R. et Leitch H., " LES FILTRES NUMERIQUES ",  
Masson et CNET-ENST, Paris, 1980.
- [11] Lindsey William C. y Simon Marvin K., "TELECOMUNICATIONS  
SYSTEM ENGINEERING ", Prentice-Hall Inc., 1973
- [12] Oppenheim Alan V. Willsky Alan S. con Yong Ian T.,  
" SIGNAL AND SYSTEMS ", Prentice-Hall, Inc, Englewood  
Cliffs, New Jersey, 1983.
- [13] Vidaller L. Riera J. B. Viñas J. , " TRANSMISION DE  
DATOS ", Universidad Politécnica de Madrid ,  
Departamento de Publicaciones E.T.S.I. de  
Telecomunicaciones c/. Ciudad Universitaria, s/n  
Madrid-3 , 1979.
- [14] Antoniu Albert , " DIGITAL FILTERS ", Mc Graw\_Hill,  
New York, 1980.
- [15] Schildt Helbert, " C: MANUAL DE REFERENCIA .", McGraw-  
Hill/Interamericana de España S.A., Madrid, 1990.