

T
62

R.5343



UNA APORTACION AL ESTUDIO DE FILTROS Y
OSCILADORES DIGITALES DE BLOQUE.

Memoria presentada por
ARANZAZU AYERBE GARCIA
para optar al Grado de Doctor en Ciencias Físicas.

A handwritten signature in black ink, which appears to be "Aranzazu Ayerbe Garcia", enclosed within a hand-drawn oval.

EL DIRECTOR

A handwritten signature in black ink, which appears to be "J.I. Acha Catalina", written in a cursive style.

J.I.ACHA CATALINA
Catedrático de Electrónica

Departamento de Electricidad y Electrónica

Facultad de Física

Universidad de Sevilla

Marzo 1987



UNIVERSIDAD DE SEVILLA
SECRETARIA GENERAL

UNIVERSIDAD DE SEVILLA

Depositado en El. Dpto. de Electricidad y Electrónica

Queda registrada esta Tesis Doctoral de la *F. de Física*
al folio 1 número 8 del libro de esta Universidad desde el día *25-Marzo-87*
correspondiente. hasta el día *13-Abril-87*.

Sevilla, 25 MAR. 1987

El Jefe del Negociado de Tesis,

H. Galanda Diaz Roldán

Sevilla *29 de Mayo* de *1987*
EL DIRECTOR DEL DPTO.

A mis padres

Quiero expresar mi agradecimiento a:

- J.I. Acha por la dirección de este trabajo, cuya ayuda y dedicación han hecho posible su realización.
- Al Departamento de Electricidad y Electrónica por el apoyo prestado.
- Al Departamento de Investigación y Desarrollo de Telefónica por haberme facilitado los medios necesarios para la elaboración de este trabajo.
- A Gerardo Abella Gavela por su colaboración en la realización de los gráficos.
- A Juan y a Asun por su ayuda en la edición de la Memoria.

INDICE.

Capítulo 1. FILTROS DE BLOQUE

- 1.0. Introducción
- 1.1. Métodos de descripción de estructuras de bloques:
 - 1.1.A. Representación Matricial.
 - 1.1.B. Representación en variables de estado.
 - 1.1.C. Comparación entre ambas representaciones.
- 1.2. Conclusiones.
- 1.3. Referencias.

Capítulo 2. SENSIBILIDAD DE ESTRUCTURAS DE BLOQUE DE ESTADO

- 2.0. Introducción
- 2.1. Sensibilidad de la magnitud de la respuesta en frecuencia de un filtro.
- 2.2. Matriz de transferencia de una estructura de bloques.
- 2.3. Sensibilidad de la respuesta en frecuencia de un filtro realizado con estructuras de bloques.
- 2.4. Programa "NUV" de medida de sensibilidad.
- 2.5. Resultados experimentales de simulación.
- 2.6. Referencias.

Capítulo 3. OSCILADOR DE BLOQUES

- 3.0. Introducción.
- 3.1. Consideraciones generales.
- 3.2. Oscilador digital de bloques.
- 3.3. Estructura de oscilador en bloque de estado con sensibilidad mínima.
- 3.4. Escalado de la señal.
- 3.5. Efectos de la cuantización de productos: ruido.
- 3.6. Otros osciladores de bloques.
- 3.7. Resultados experimentales.
- 3.8. Comparación con el oscilador escalar.
- 3.9. Referencias.

Capítulo 4. REALIZACION SOBRE TMS32010 DE UN OSCILADOR DE BLOQUES DE BAJA FRECUENCIA

- 4.0. Introducción.
- 4.1. Manejo y aplicaciones del TMS32010 DSP.
- 4.2. Equipo de desarrollo del procesador.
- 4.3. Procedimiento general para la realización de un sistema sobre dicho procesador.
- 4.4. Descripción del oscilador de bloques realizado.
- 4.5. Resultados experimentales: simulación y emulación.
- 4.6. Referencias.

CONCLUSIONES

CAPITULO 1.

FILTROS DE BLOQUE

1.0. INTRODUCCION.

En el campo del procesamiento digital de la señal y, como consecuencia, en el del filtrado digital, la aparición de los circuitos integrados a gran escala (VLSI) ha reducido el énfasis que existía por minimizar el número de operaciones y ha desviado la atención a la búsqueda de estructuras que usen muchos dispositivos en paralelo. Además, el cambio experimentado por la tecnología, así como los avances en Microelectrónica y el abaratamiento de los microprocesadores han contribuido a avanzar en dicha dirección.

En este sentido, el filtrado de bloques, caracterizado esencialmente por realizar el procesado de muestras de entrada por secciones o bloques, en lugar de una muestra cada vez como en el caso del filtrado escalar, tiene la propiedad de ser isomorfo con el procesamiento en paralelo, y por lo tanto se convierte en objeto directo de nuestro interés.

Las ventajas que ofrece el filtrado por bloques son numerosas y han sido ampliamente descritas en la Bibliografía [1,2,9]. Enumeremos a continuación algunas de ellas:

- Posibilidad de uso de técnicas rápidas de convolución (FFT) para la realización de operaciones intermedias.

- Implementación eficiente de dichas estructuras a través de procesadores paralelo.

- Reducción del ruido de redondeo.

- Posibilidad de representar a los filtros variables en el tiempo de forma periódica por filtros digitales invariables en el tiempo.

- La técnica de formulación por bloques parece ser aplicable en la búsqueda de cotas absolutas en los ciclos límite que se presentan en los filtros escalares.

- Parece lógico, además, suponer que dichos filtros tendrán una figura de sensibilidad de la estabilidad a la cuantización de los coeficientes menor que sus correspondientes escalares, debido a que, como veremos más adelante, el movimiento de los polos sigue una espiral hacia el origen.

Centrado, pues, nuestro interés en el filtrado digital por bloques, se hará en este Capítulo una breve exposición de los distintos métodos de descripción de los filtros de bloque.

Estos métodos son fundamentalmente dos, el primero que se basa en la "matriz de convolución" [1-7] y el segundo basado en la formulación por variables de estado [9-13]. A estos dos, hay que añadir otros basados en la técnica de realización por "two pairs" [8-14].

1.1. METODOS DE DESCRIPCION DE ESTRUCTURAS DE BLOQUE

1.1.A.- METODO BASADO EN LA REPRESENTACION MATRICIAL DE LA CONVOLUCION.

Consideremos un filtro digital IIR, de orden M , lineal e invariante con el tiempo, descrito por una función de transferencia racional donde $u(n)$ es la entrada, $y(n)$ es la salida y $h(n)$ es la respuesta impulsiva. Resumiendo, las relaciones que lo caracterizan son :

$$y(n) = \sum_{j=0}^{\infty} h(n-j) \cdot u(j) = h(n) * u(n) \quad (1.1)$$

$$H(z) = \sum_{i=0}^{\infty} h(i)z^{-i} = \left(\sum_{i=0}^N a_i z^{-i} \right) / \left(\sum_{i=0}^M b_i z^{-i} \right) = A(z) / B(z) \quad (1.2)$$

$$Y(z) = H(z) \cdot U(z) \quad (1.3)$$

donde,

* representa la convolución

$b_0=1$ sin pérdida de generalidad

$U(z)$, $H(z)$, $Y(z)$ son las transformadas z de $u(n)$, $h(n)$ e $y(n)$ respectivamente.

Si suponemos ahora la entrada, la salida y la respuesta impulsiva divididas en bloques de longitud L , tendremos los vectores de dimensión L siguientes:

$$\begin{aligned} \hat{\tilde{u}}(k) &= [u(kL) \quad u(kL+1) \quad \dots \quad u(kL+L-1)]^t = \\ &= [\hat{u}_1(k) \quad \hat{u}_2(k) \quad \dots \quad \hat{u}_L(k)]^t \\ \hat{\tilde{y}}(k) &= [y(kL) \quad y(kL+1) \quad \dots \quad y(kL+L-1)]^t = \\ &= [\hat{y}_1(k) \quad \hat{y}_2(k) \quad \dots \quad \hat{y}_L(k)]^t \\ \hat{\tilde{h}}(k) &= [h(kL) \quad h(kL+1) \quad \dots \quad h(kL+L-1)]^t = \\ &= [\hat{h}_1(k) \quad \hat{h}_2(k) \quad \dots \quad \hat{h}_L(k)]^t \end{aligned} \quad (1.4)$$

Por otro lado, de las expresiones (1.2) y (1.3) puede deducirse que

$$B(z) \cdot Y(z) = A(z) \cdot U(z) \quad (1.5)$$

lo cual en el dominio del tiempo significa :

$$b(n) * y(n) = a(n) * u(n) \quad (1.6)$$

siendo $b(n)$ una secuencia finita cuya transformada z es $B(z)$ y $a(n)$ otra secuencia de transformada $A(z)$.

Así pues,

$$\sum_{j=0}^{\infty} y(n-j) \cdot b(j) = \sum_{j=0}^{\infty} u(n-j) \cdot a(j) \quad (1.7)$$

y expresando dicha ecuación en forma matricial :

$$\begin{bmatrix} b_0 & 0 & 0 & \dots & 0 \\ b_1 & b_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_N & b_{N-1} & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_M & b_{M-1} & \dots & \dots & b_0 \end{bmatrix} \cdot \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} a_0 & 0 & 0 & \dots & 0 \\ a_1 & a_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_N & a_{N-1} & \dots & a_0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & a_{M-N} & \vdots \end{bmatrix} \cdot \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad (1.8)$$

B A

siendo,

$$b_i = b(i)$$

$$a_i = a(i)$$

Por otro lado, si escribimos en forma matricial la ecuación (1.1), tendremos:

$$\begin{bmatrix} h_0 & 0 & 0 & \dots & \dots & \dots \\ h_1 & h_0 & 0 & \dots & \dots & \dots \\ h_2 & h_1 & h_0 & \dots & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots & \dots \\ \cdot & \cdot & \cdot & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \cdot \\ \cdot \end{bmatrix} \quad (1.9)$$

\underline{H}

donde,

$$h_i = h(i)$$

y si agrupamos los elementos de la matriz \underline{H} en submatrices L-L de la forma:

$$\underline{H}_0 = \begin{bmatrix} h_0 & 0 & \dots & \dots & 0 \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ h_{L-1} & \cdot & \dots & \dots & h_0 \end{bmatrix} \quad \underline{H}_1 = \begin{bmatrix} h_L & \dots & \dots & \dots & h_1 \\ h_{L+1} & \dots & \dots & \dots & h_2 \\ \cdot & \dots & \dots & \dots & \cdot \\ \cdot & \dots & \dots & \dots & \cdot \\ \cdot & \dots & \dots & \dots & \cdot \end{bmatrix} \quad (1.10)$$

la ecuación (1.9) se transformará en:

$$\begin{bmatrix} \underline{H}_0 & 0 & \dots\dots & 0 \\ \underline{H}_1 & \underline{H}_0 & \dots\dots & 0 \\ \underline{H}_2 & \underline{H}_1 & \dots\dots & 0 \\ \cdot & \cdot & \dots\dots & \cdot \\ \cdot & \cdot & \dots\dots & \cdot \\ \cdot & \cdot & \dots\dots & \cdot \end{bmatrix} \cdot \begin{bmatrix} \hat{\tilde{u}}(0) \\ \hat{\tilde{u}}(1) \\ \hat{\tilde{u}}(2) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \hat{\tilde{y}}(0) \\ \hat{\tilde{y}}(1) \\ \hat{\tilde{y}}(2) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (1.11)$$

con lo que para el instante $k+1$ ($k=0,1,\dots$) tendremos:

$$\begin{aligned} \hat{\tilde{y}}(k+1) &= \sum_{j=0}^{k+1} \underline{H}_{k+1-j} \cdot \hat{\tilde{u}}(j) = \\ &= \sum_{j=0}^k \underline{H}_{k+1-j} \cdot \hat{\tilde{u}}(j) + \underline{H}_0 \cdot \hat{\tilde{u}}(k+1) \end{aligned} \quad (1.12)$$

En nuestro estudio posterior distinguiremos dos casos:

a). $L \geq M$

En este caso, las matrices \underline{A} y \underline{B} de (1.8) se pueden partir en unas submatrices cuadradas $L \cdot L$ de la forma:

$$\underline{B}_0 = \begin{bmatrix} b_0 & 0 & \dots & 0 \\ b_1 & b_0 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ b_M & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & b_0 \end{bmatrix} \quad \underline{A}_0 = \begin{bmatrix} a_0 & 0 & \dots\dots & 0 \\ \cdot & \cdot & \dots\dots & 0 \\ a_N & \cdot & \dots\dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a_0 \end{bmatrix} \quad (1.13a)$$

$$\underline{B}_1 = \begin{bmatrix} 0 & \dots & 0 & b_M & \dots & b_1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & b_M & \dots \\ 0 & \dots & \dots & \dots & \dots & 0 \end{bmatrix} \quad \underline{A}_1 = \begin{bmatrix} 0 & \dots & \dots & \dots & a_N & \dots & a_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & a_N \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \end{bmatrix} \quad (1.13b)$$

con lo que la ecuación (1.8) puede escribirse:

$$\begin{bmatrix} \underline{B}_0 & 0 & \dots & \dots & 0 \\ \underline{B}_1 & \underline{B}_0 & \dots & \dots & \dots \\ 0 & \underline{B}_1 & \underline{B}_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \hat{\underline{y}}(0) \\ \hat{\underline{y}}(1) \\ \dots \\ \dots \\ \dots \end{bmatrix} = \begin{bmatrix} \underline{A}_0 & 0 & \dots & \dots \\ \underline{A}_1 & \underline{A}_0 & \dots & \dots \\ 0 & \underline{A}_1 & \underline{A}_0 & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \hat{\underline{u}}(0) \\ \hat{\underline{u}}(1) \\ \dots \\ \dots \\ \dots \end{bmatrix} \quad (1.14)$$

y para el instante $(k+1)(k=0,1..)$, tendremos:

$$\underline{B}_0 \cdot \hat{\underline{y}}(k+1) + \underline{B}_1 \cdot \hat{\underline{y}}(k) = \underline{A}_0 \cdot \hat{\underline{u}}(k+1) + \underline{A}_1 \cdot \hat{\underline{u}}(k) \quad (1.15)$$

Y teniendo en cuenta (1.13a) resulta evidente que \underline{B}_0^{-1} siempre existe, por lo que despejando en (1.15),

$$\hat{\underline{y}}(k+1) = -\underline{B}_0^{-1} \underline{B}_1 \cdot \hat{\underline{y}}(k) + \underline{B}_0^{-1} \underline{A}_0 \cdot \hat{\underline{u}}(k+1) + \underline{B}_0^{-1} \underline{A}_1 \cdot \hat{\underline{u}}(k) \quad (1.16)$$

ecuación que proporciona un bloque de salida conociendo el bloque anterior en la salida, y el presente y anterior en la entrada.

Identificando esta expresión con la obtenida en (1.12), resulta que:

$$\underline{H}_0 = \underline{B}_0^{-1} \cdot \underline{A}_0 = \underline{A}_0 \cdot \underline{B}_0^{-1} \quad (1.17)$$

y realizando operaciones se obtiene:

$$\underline{H}_1 = -\underline{B}_0^{-1} \underline{B}_1 \underline{H}_0 + \underline{B}_0^{-1} \underline{A}_1 = -\underline{H}_0 \cdot \underline{B}_1 \cdot \underline{B}_0^{-1} + \underline{A}_1 \cdot \underline{B}_0^{-1} \quad (1.18)$$

Tanto en (1.17), como en (1.18) se ha tenido en cuenta que las matrices conmutan colectivamente [8].

La representación de la estructura se encuentra en la Figura I.1, donde el bloque D es un bloque de retraso.

b). $L < M$

En este caso, las relaciones resultan un poco más complicadas.

Definimos los enteros p y q de la siguiente forma:

$$p = \lceil M/L \rceil \quad , \quad q = \lceil N/L \rceil$$

donde $\lceil x \rceil$ representa al menor entero mayor o igual a x.

Obtenemos (p+1) y (q+1) matrices distintas, de forma que la ecuación (1.8) se transforma en :

$$\begin{bmatrix} \underline{B}_0 & 0 & \cdot & \cdot & \cdot \\ \underline{B}_1 & \underline{B}_0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{B}_p & \underline{B}_{p-1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \cdot \begin{bmatrix} \hat{\underline{y}}(0) \\ \hat{\underline{y}}(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \underline{A}_0 & 0 & \cdot & \cdot & \cdot \\ \underline{A}_1 & \underline{A}_0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{A}_q & \underline{A}_{q-1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \hat{\underline{u}}(0) \\ \hat{\underline{u}}(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (1.19)$$

y para el instante k resultará:

$$\begin{aligned} \underline{B}_0 \cdot \hat{\underline{y}}(k) + \underline{B}_1 \cdot \hat{\underline{y}}(k-1) + \dots + \underline{B}_p \cdot \hat{\underline{y}}(k-p) &= \\ &= \underline{A}_0 \cdot \hat{\underline{u}}(k) + \underline{A}_1 \cdot \hat{\underline{u}}(k-1) + \dots + \underline{A}_q \cdot \hat{\underline{u}}(k-q) \end{aligned} \quad (1.20)$$

que no es más que una ecuación en diferencias de orden p.

Resulta evidente que cuando $L \geq M$, $p=1$ y se obtiene la ecuación (1.16).

La estructura correspondiente a la ecuación (1.20) se encuentra en la Figura I.2.

Comentarios sobre estabilidad.

En el caso particular y más frecuente, $L > M$ la estabilidad quedará determinada por los autovalores de $\underline{B}_0^{-1}\underline{B}_1$.

\underline{B}_0 es de rango L , y \underline{B}_1 , aunque es de orden $L+1$, tiene de rango M (orden del filtro).

Así pues, el rango de $\underline{B}_0^{-1}\underline{B}_1$ será M , por lo que al menos tendremos $(L-M)$ autovalores iguales a cero.

Un examen detenido de la ecuación de bloques (1.16), permite afirmar que la misma trabaja a $1/L$ veces la velocidad a la que trabaja la implementación directa de $H(z)$. Si se interpreta dicha ecuación como una ecuación estandarizada de la ecuación normal de salida, el efecto sería el mismo que muestrear la salida cada L puntos, lo cual implica que los autovalores de la estructura de bloques son los de la estructura escalar elevados a la L -ésima potencia.

Esto da idea exacta de la localización de los autovalores, $(L-M)$ están en el origen, y M son los originales elevados a L , por lo que si la función de transferencia original era estable, la implementación por bloques será "más estable" en el sentido de que los autovalores están todavía más dentro del círculo unidad.

Este hecho se encuentra demostrado en [10] de forma totalmente general, independientemente de la estructura elegida para la realización del filtro de bloques.

En cuanto al ahorro computacional que se consigue con la realización por bloques de los filtros, puede encontrarse una amplia discusión en [4] y que, nosotros, por motivos de brevedad, excluimos de esta introducción.

1.1.B.- METODO DE LAS VARIABLES DE ESTADO.

Una segunda técnica, muy utilizada, para la descripción de estructuras de bloque es aquella basada en la formulación por variables de estado, desarrollada por Barnes [9-11].

Consideremos un filtro digital invariante en el tiempo con una sola secuencia de entrada $u(n)$ y una de salida $y(n)$. Su descripción en variables de estado, suponiendo el filtro de orden M , será:

$$\begin{aligned} \tilde{x}(n+1) &= \underline{A} \cdot \tilde{x}(n) + \underline{B} \cdot u(n) \\ y(n) &= \underline{C} \cdot \tilde{x}(n) + d \cdot u(n) \end{aligned} \tag{1.21}$$

donde $x(n)$ es el vector de estados y tendrá de dimensión M . \underline{A} es la matriz de estados de orden M , \underline{B} y \underline{C} son matrices de orden $(M \cdot 1)$ y $(1 \cdot M)$ respectivamente y d es un escalar.

Diremos entonces que $[\underline{A}, \underline{B}, \underline{C}, d]$ constituyen la realización simple ó escalar en el espacio de estados del filtro.

Supongamos ahora, la entrada y la salida agrupadas en bloques de longitud L , definiendo $\hat{\tilde{u}}(k)$, $\hat{\tilde{y}}(k)$ como en (1.4). Análogamente, el vector $\tilde{x}(n)$ dará lugar a una matriz de la forma :

$$\begin{aligned}
 \underline{x}(k) &= \begin{bmatrix} x^1(kL) & x^1(kL+1) & \dots & x^1(kL+L-1) \\ x^2(kL) & x^2(kL+1) & \dots & x^2(kL+L-1) \\ \cdot & \cdot & \dots & \cdot \\ x^M(kL) & x^M(kL+1) & \dots & x^M(kL+L-1) \end{bmatrix} = \\
 &= \begin{bmatrix} x_1^1(k) & x_2^1(k) & \dots & x_L^1(k) \\ x_1^2(k) & x_2^2(k) & \dots & x_L^2(k) \\ \cdot & \cdot & \dots & \cdot \\ x_1^M(k) & x_2^M(k) & \dots & x_L^M(k) \end{bmatrix} \quad (1.22)
 \end{aligned}$$

donde designamos por $\hat{x}(k) = \underline{x}_1(k)$, es decir la primera columna de la matriz (1.22). Como veremos a continuación, la información contenida en el resto de las columnas puede concentrarse de alguna forma sobre la primera.

Veamos ahora como queda la descripción en variables de estado.

Para las muestras tomadas en el instante kL , la siguiente muestra será en $kL+1$, luego:

$$\underline{\hat{x}}(kL+1) = \underline{A} \cdot \underline{\hat{x}}(kL) + \underline{B} u(kL)$$

$$\begin{aligned}
\tilde{x}(kL+2) &= \underline{A} \cdot \tilde{x}(kL+1) + \underline{B} \cdot u(kL+1) = \\
&= \underline{A}^2 \cdot \tilde{x}(kL) + \underline{A} \cdot \underline{B} \cdot u(kL) + \underline{B} \cdot u(kL+1) \\
&\cdot \\
&\cdot \\
&\cdot \\
\tilde{x}(kL+L) &= \underline{A}^L \cdot \tilde{x}(kL) + \underline{A}^{L-1} \underline{B} \cdot u(kL) + \underline{A}^{L-2} \underline{B} \cdot u(kL+1) + \dots + \\
&+ \underline{A} \cdot \underline{B} \cdot u(kL+L-2) + \underline{B} \cdot u(kL+L-1)
\end{aligned}$$

(1.23)

y teniendo en cuenta la nomenclatura ya empleada anteriormente , podemos escribir:

$$\tilde{x}_1(k+1) = \underline{A}^L \cdot \tilde{x}_1(k) + [\underline{A}^{L-1} \underline{B} \ ; \ \underline{A}^{L-2} \underline{B} \ ; \ \dots \ \underline{A} \cdot \underline{B} \ ; \ \underline{B}] \cdot \begin{bmatrix} \hat{u}_1(k) \\ \hat{u}_2(k) \\ \cdot \\ \cdot \\ \cdot \\ \hat{u}_L(k) \end{bmatrix}$$

(1.24)

o lo que es lo mismo:

$$\hat{\tilde{x}}(k+1) = \hat{\underline{A}} \cdot \hat{\tilde{x}}(k) + \hat{\underline{B}} \cdot \hat{\tilde{u}}(k)$$

(1.25)

donde:

$$\hat{\underline{A}} = \underline{A}^L$$

$$\hat{\underline{B}} = [\underline{A}^{L-1} \underline{B} \ ; \ \underline{A}^{L-2} \underline{B} \ \dots \ \underline{A} \cdot \underline{B} \ ; \ \underline{B}]$$

En cuanto a la ecuación de salida y siguiendo el mismo proceso, podemos escribir,

$$y(kL) = \underline{C} \cdot \underline{x}(kL) + d \cdot u(kL)$$

$$y(kL+1) = \underline{C} \cdot \underline{x}(kL+1) + d \cdot u(kL+1) = \underline{C} \cdot \underline{A} \cdot \underline{x}(kL) + \underline{C} \cdot \underline{B} \cdot u(kL) + d \cdot u(kL+1)$$

$$y(kL+2) = \underline{C} \cdot \underline{x}(kL+2) + d \cdot u(kL+2) =$$

$$= \underline{C} \cdot \underline{A}^2 \cdot \underline{x}(kL) + \underline{C} \cdot \underline{A} \cdot \underline{B} \cdot u(kL) + \underline{C} \cdot \underline{B} \cdot u(kL+1) + d \cdot u(kL+2)$$

·
·
·

$$y(kL+L-1) = \underline{C} \cdot \underline{A}^{L-1} \underline{x}(kL) + \underline{C} \cdot \underline{A}^{L-2} \underline{B} \cdot u(kL) + \underline{C} \cdot \underline{A}^{L-3} \underline{B} \cdot u(kL+1) + \dots +$$

$$+ \underline{C} \cdot \underline{B} \cdot u(kL+L-2) + d \cdot u(kL+L-1)$$

(1.26)

Y expresándolo en forma matricial :

$$\hat{\underline{y}}(k) = \hat{\underline{C}} \cdot \hat{\underline{x}}(k) + \hat{\underline{D}} \cdot \hat{u}(k) \quad (1.27)$$

donde,

$$\hat{\underline{C}} = \begin{bmatrix} \underline{C} \\ \underline{C} \cdot \underline{A} \\ \underline{C} \cdot \underline{A}^2 \\ \cdot \\ \cdot \\ \underline{C} \cdot \underline{A}^{L-1} \end{bmatrix}$$

orden L·M

$$\hat{\underline{D}} = \begin{bmatrix} d & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \underline{C} \cdot \underline{B} & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \underline{C} \cdot \underline{A} \cdot \underline{B} & \underline{C} \cdot \underline{B} & d & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{C} \cdot \underline{A}^{L-2} \underline{B} & \cdot & \cdot & \cdot & \cdot & \underline{C} \cdot \underline{B} & d \end{bmatrix}$$

Resumiendo, tenemos la realización en "bloques de estado" del filtro escalar (1.21), $[\hat{\underline{A}}, \hat{\underline{B}}, \hat{\underline{C}}, \hat{\underline{D}}]$ descrita por las ecuaciones:

$$\hat{\underline{x}}(k+1) = \hat{\underline{A}} \cdot \hat{\underline{x}}(k) + \hat{\underline{B}} \cdot \hat{\underline{u}}(k) \tag{1.28}$$

$$\hat{\underline{y}}(k) = \hat{\underline{C}} \cdot \hat{\underline{x}}(k) + \hat{\underline{D}} \cdot \hat{\underline{u}}(k)$$

donde, $\hat{\underline{x}}(k)$ es un vector columna de dimensión M

$\hat{\underline{y}}(k)$ y $\hat{\underline{u}}(k)$ son vectores columna de dimensión L

$$\hat{\underline{A}} = \underline{A}^L$$

$$\hat{\underline{B}} = [\underline{A}^{L-1} \underline{B} \ ; \ \underline{A}^{L-2} \underline{B} \ ; \ \dots \ \dots \ \underline{A} \cdot \underline{B} \ ; \ \underline{B}] \quad M \cdot L$$

$$\hat{\underline{C}} = \begin{bmatrix} \underline{C} \\ \underline{C} \cdot \underline{A} \\ \cdot \\ \underline{C} \cdot \underline{A}^{L-1} \end{bmatrix} \quad L \cdot M \tag{1.29a}$$

y $\hat{\underline{D}}$ es una matriz de dimensión $L \cdot L$, triangular inferior y de Toeplitz, definida por:

$$\hat{D}_{ij} = \begin{cases} 0 & i < j \\ d & i = j \\ \underline{C} \cdot \underline{A}^{i-j-1} \underline{B} & i > j \end{cases} \quad (1.29b)$$

La función de transferencia del filtro podrá realizarse por un filtro multientrada-multisalida, con un registro de entrada serie-salida paralelo y otro de entrada paralelo-salida serie, como se indica en la Figura I.3.

Como ejemplo, el diagrama de flujo correspondiente a la realización por bloques, con $L=3$, del filtro escalar de segundo orden, cuyo diagrama de flujo se presenta en la Figura I.4, puede observarse en la Figura I.5.

Comparando las ecuaciones (1.21) y (1.28), se pueden hacer una serie de consideraciones:

1.- Como $\hat{\underline{A}} = \underline{A}^L$, es evidente que los autovalores de $\hat{\underline{A}}$ son los de \underline{A} elevados a la L -ésima potencia.

2.- La dimensión del espacio de estados de la estructura de bloques es la misma que la del escalar.

3.- Si la realización de la función de transferencia a través de la estructura escalar de estado es mínima, la correspondiente realización de la matriz de transferencia a través de la estructura de bloques de estado también lo es.

4.- Los polos de la realización de bloques son la potencia L -ésima de los polos de la realización escalar.

5.- Cualquier transformación de equivalencia en la realización escalar de estado $[\underline{S} \cdot \underline{A} \cdot \underline{S}^{-1}, \underline{S} \cdot \underline{B}, \underline{C} \cdot \underline{S}^{-1}, d]$, dará lugar a la transformación de equivalencia correspondiente en la realización en bloques de estado, esto es, $[\underline{S} \cdot \hat{\underline{A}} \cdot \underline{S}^{-1}, \underline{S} \cdot \hat{\underline{B}}, \hat{\underline{C}} \cdot \underline{S}^{-1}, \hat{\underline{D}}]$.

6.- Además, se encuentra exhaustivamente demostrado en [9-11] que, en lo que se refiere al escalado, si la realización escalar se encuentra escalada con norma L_2 , la correspondiente realización por bloques de estado también lo estará con las mismas cotas. Y en lo que se refiere al ruido de redondeo a la salida de la realización de la estructura de bloques se encuentra que es no-estacionario, que su varianza es periódica con la longitud de los bloques y que la máxima varianza del ruido de redondeo no es nunca superior, es decir, no sobrepasa la varianza del ruido de redondeo de la realización escalar correspondiente. En general puede decirse que el efecto de la realización por bloques es reducir el ruido de redondeo internamente generado.

1.1.C. COMPARACION ENTRE AMBAS REPRESENTACIONES.

Examinando las dos formas de representación de estructuras de bloque que hemos presentado, puede observarse que existen dos diferencias fundamentales entre ellas.

La primera consiste en que en la descripción convolucional todas las matrices características de la estructura son matrices del tipo convolución, de orden $L \cdot L$, mientras que en la descripción por variables de estado no puede decirse lo mismo y tan solo la matriz \hat{D} es de Toeplitz y de dimensión $L \cdot L$.

Y la segunda diferencia se refiere a la dimensión del vector de estados en cada caso. En el segundo, es independiente de la longitud del bloque [9], mientras que en el primero aumenta directamente con L [8].

Como ya hemos visto, dependiendo del tipo de descripción se ha atendido a unas propiedades u otras. Así, para la primera, se han considerado aquellas propiedades referentes a la eficiencia computacional [3,4], y para la segunda se han encontrado propiedades referentes al escalado, ruido y eliminación de ciclos límite [9,11].

Una estructura combinada, propuesta por Meek [6], ha puesto de manifiesto la posibilidad de establecer una equivalencia entre ambos tipos de descripciones [15] con lo cual muchas de las propiedades y técnicas de una descripción pueden aplicarse a la otra.

1.2. CONCLUSIONES

En resumen, hemos realizado en este Capítulo una descripción de los dos métodos fundamentales de descripción de filtros de bloque, el basado en la representación matricial de la convolución y el de variables de estado.

Existen, lógicamente, una gran variedad de estructuras alternativas, como los "two pairs" [8,14], etc, que no hemos descrito por considerar que se alejaban en cierto modo del objetivo del presente trabajo.

Hemos pretendido introducir las formas generales de descripción de las estructuras de bloque, así como presentar cualitativamente las ventajas que representan las mismas sobre las estructuras escalares correspondientes.

1.3. REFERENCIAS.

- [1]. B.Gold and K.L.Jordan,Jr., "A note on Digital Filter Synthesis". Proc.IEEE.(Lett), Vol.56, pp.1717-1718, 1968.
- [2]. H.B.Voelcker and E.E.Hartquist, "Digital Filtering via Block recursion". Trans. Audio Electroacoust. (Special Issue on Digital Filtering), Vol.AU-18, pp.169-176. 1970.
- [3]. C.S.Burrus, "Block Implementation of Digital Filters". IEEE Trans. Circuit Theory, Vol.CT-18, pp.697-701. 1971.
- [4]. C.S.Burrus, "Block realization of Digital Filters". IEEE Trans. Audio Electroacoust. Vol.AU-20, pp.230-235. 1972.
- [5]. Read and J. Meek, "Digital Filters with poles via the FFT". IEEE Trans. Audio Electroacoust. (Corresp.), Vol.AU-19, pp.322-323. 1971
- [6]. J.W.Meek and A.S. Veletsos, "Fast Convolution for Recursive Digital Filters". IEEE Trans. Audio Electroacoust. (Corresp.), Vol.AU-20, pp.93-94. 1972.

- [7]. R.Gnanasekaran and S.K.Mitra, "A note on block implementation of IIR digital filters". Proc. IEEE, vol.65, pp.1063-1064. 1977.
- [8]. S.K.Mitra and R.Gnanasekaran, "Block implementation of recursive digital filters-new structures and properties". IEEE Trans. Circuits Syst., Vol.CAS-25, pp.200-207. 1978.
- [9]. C.W.Barnes and S. Shinnanka, "Finite word effects in block-state realizations of fixed point digital filters". IEEE Trans. Circuits Syst., Vol.CAS-27, pp.345-349. 1980
- [10]. C.W.Barnes and S.Shinnanka, "Block shift invariance and block implementation of discrete-time filters". IEEE Trans. Circuits Syst., Vol.CAS-27, pp.667-672. 1980.
- [11]. C.W.Barnes and S.Shinnanka, "Stability Domains for Second-order Recursive Digital Filters in Normal Form with 'Matrix Power' Feedback". IEEE Trans. Circuits Syst. Vol.CAS-27, pp.841-843. 1980.
- [12]. P.Ananthakrishna and S.K.Mitra, "A general Theory of Block State Digital Filters". Proc. IEEE. Int. Symp. on Circuits and Syst., pp.1025-1028. 1982.
- [13]. P.Ananthakrishna and S.K.Mitra, "Minimum Round-off noise realization of Block-state Recursive Digital Filters". Circuits Syst. Signal Process.,

Vol.1, No.2, pp.217-231. 1982

- [14]. P.Ananthakrishna, B.D.O.Anderson and S.K.Mitra,
"Realization of Digital Transfer Functions
using cascaded lattice and ladder Block Structures".
Journal of the Franklin Inst. Vol.136, No.1.,
pp.93-124. 1983.
- [15]. C.M.Loeffler, and C.S.Burrus, "Equivalence
of block filter representations". Int. Symp.
on Circuits and Syst. 1981.

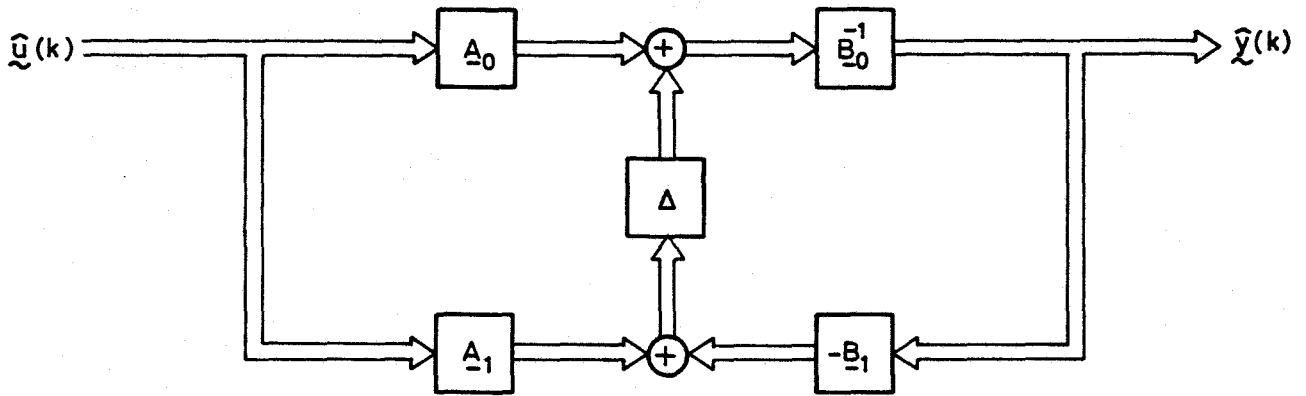


FIGURA 1.1- ESTRUCTURA DE BLOQUES DIRECTA DE UN FILTRO CON $L \geq M$

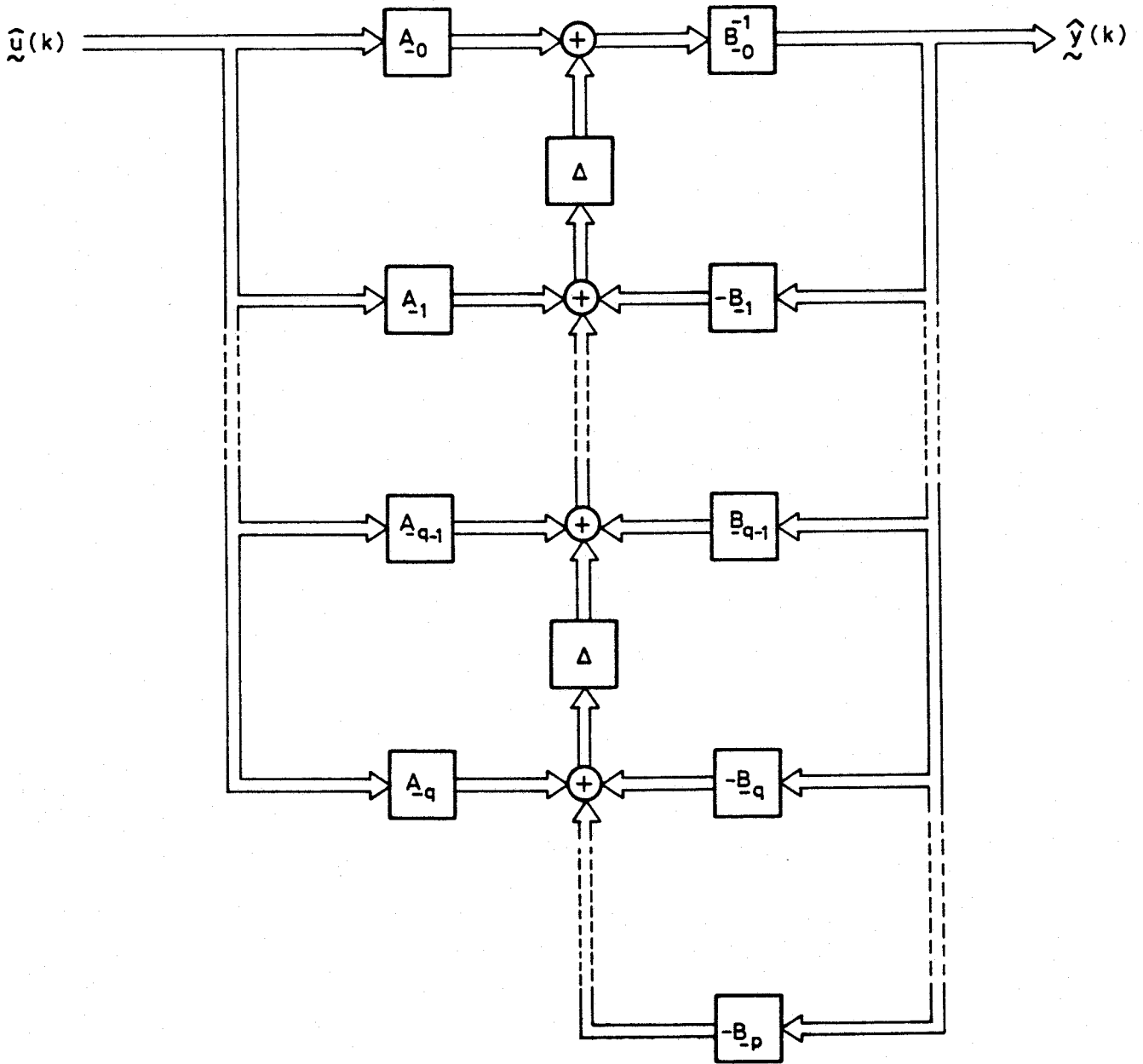


FIGURA 1.2.-REALIZACION POR ESTRUCTURA DE BLOQUE DIRECTA DE UN FILTRO DIGITAL CON $L < M$

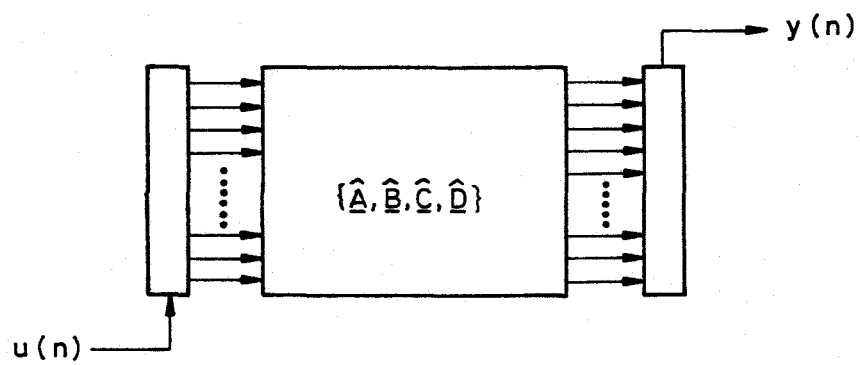


FIGURA I. 3.- REALIZACION EN BLOQUE DE ESTADO USANDO REGISTROS
 ENTRADA SERIE - SALIDA PARALELO Y ENTRADA
 PARALELO - SALIDA SERIE

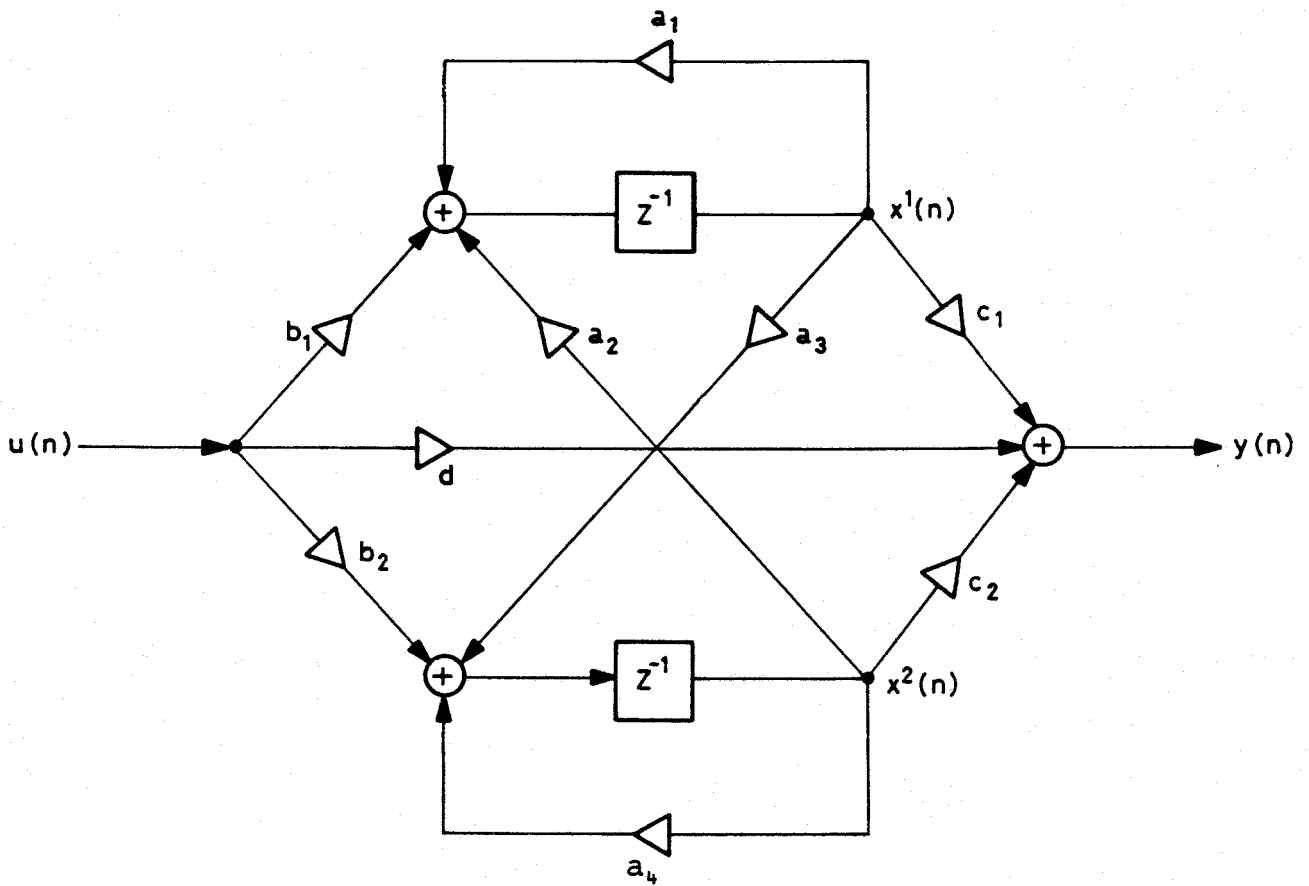


FIGURA I. 4:- FILTRO ESCALAR DE SEGUNDO ORDEN ($M = 2$)

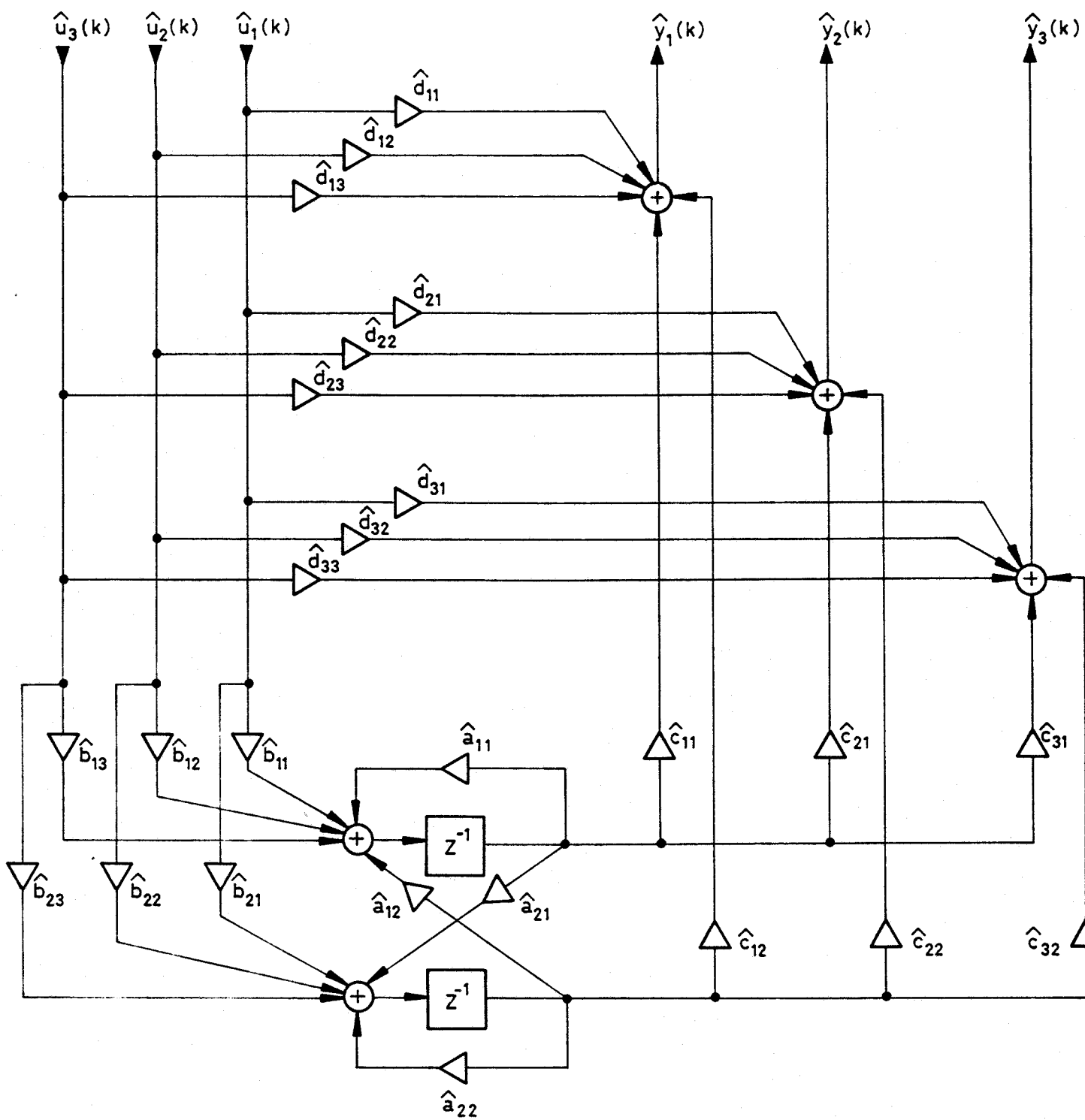


FIGURA 1.5.- FILTRO DE BLOQUES DE SEGUNDO ORDEN, CON $L = 3$

CAPITULO 2.

SENSIBILIDAD DE ESTRUCTURAS

DE BLOQUE DE ESTADO.

2.0. INTRODUCCION.

Uno de los problemas que se presentan a la hora de la realización de un filtro digital tiene su origen en la imposibilidad de utilización de registros de longitud infinita para la representación de los parámetros característicos del filtro. Los efectos de dicha limitación son principalmente [1] :

- Modificación de la función de transferencia deseada debido a la cuantificación de los coeficientes.
- Ruido de redondeo de las multiplicaciones.
- Ciclos Límites (aspectos no lineales)

Con la realización por bloques de los filtros digitales este problema no se elude y por tanto, nos encontramos ante el mismo simplemente con la esperanza de que la realización por bloques contribuya a una

disminución de los efectos anteriores o que, al menos, no los incremente.

Después de haber presentado en el Capítulo anterior las diferentes formas de descripción de los filtros de bloque, abordaremos en el presente el estudio de la sensibilidad de la función de transferencia de un filtro, realizado empleando estructuras de bloques, ante la cuantización de los coeficientes.

Evidentemente, y según se indica en [1], el comportamiento del filtro se verá afectado por el tipo de aritmética empleada en la realización de los algoritmos del filtro, del tipo de cuantización empleado, y de la estructura elegida para la realización del mismo en cada caso específico.

En nuestro caso, estudios previos [2,3] referentes a sensibilidad de filtros respecto a cuantización de coeficientes han puesto de manifiesto la conveniencia de realización de filtros recursivos mediante la conexión de cascadas de estructuras de segundo orden. Estos filtros presentan una sensibilidad muy baja para la banda pasante y para la banda de rechazo.

Siguiendo estos resultados centraremos nuestro interés en el estudio de la sensibilidad de los filtros de bloque realizados por conexión en cascada de secciones de segundo orden. De ahí que particularicemos nuestro estudio a estructuras de segundo orden.

En lo que se refiere al ruido de redondeo después de las multiplicaciones y el "overflow" después de las sumas, estos han sido ampliamente estudiados por Barnes y Shinnanka [4] y Mitra [5,6] y no los abordaremos en este Capítulo.

En el apartado 1 trataremos de la definición general de sensibilidad, continuando en el apartado 2 con la manera de determinar la función de transferencia de un filtro de bloques a través de su matriz de transferencia .

A continuación, en el apartado 3, encontraremos expresiones concretas para la sensibilidad de la respuesta en frecuencia de filtros de bloques de segundo orden.

En el apartado 4 se hará una descripción de los programas de simulación de realización de filtros y de cálculo de la sensibilidad, realizados para comprobar lo expuesto en los apartados anteriores.

Por último, en el apartado 5, se presentarán los resultados experimentales obtenidos con los programas anteriores sobre estructuras de bloque concretas y se compararán con las estructuras escalares correspondientes.

2.1. SENSIBILIDAD DE LA MAGNITUD DE LA RESPUESTA

EN FRECUENCIA DE UN FILTRO

Como ya hemos dicho, el número de bits usado para representar los coeficientes del filtro es finito. Esto introduce una modificación en el comportamiento del mismo respecto a las especificaciones que le han sido impuestas en su proceso de diseño. Esta modificación se hace del todo patente en la magnitud de la respuesta en frecuencia.

La respuesta en frecuencia de un filtro cuya función de transferencia es $H(z)$, se obtiene haciendo en la misma $z=e^{j\omega}$.

En general, si en la realización del filtro se emplean q coeficientes m_i sometidos a cuantización, la modificación producida sobre la respuesta en frecuencia del mismo vendrá dada por:

$$\Delta |H(e^{j\omega})| \approx \sum_{i=1}^q (\partial |H(e^{j\omega})| / \partial m_i) \cdot \Delta m_i \quad (2.1)$$

donde Δm_i es la modificación experimentada por la cuantización del coeficiente m_i .

Definimos la SENSIBILIDAD de primer orden de la respuesta en frecuencia como

$$S_{m_i} |H| = \partial |H(e^{j\omega})| / \partial m_i \quad (2.2)$$

y la SENSIBILIDAD SEMI-NORMALIZADA como,

$$\hat{S}_{m_i} |H| = m_i \cdot (\partial |H(e^{j\omega})| / \partial m_i) \quad (2.3)$$

Por lo general, existirán diferencias en los órdenes de magnitud de los distintos coeficientes m_i . Suponiendo entonces que las tolerancias de los mismos, $(\Delta m_i / m_i)$, son de igual magnitud y están normalizadas a la unidad, será posible encontrar una cota superior para la variación en $|H(e^{j\omega})|$ dada por (2.1),

$$\begin{aligned} |\Delta |H(e^{j\omega})|| &\leq \sum_{i=1}^q |m_i \cdot (\partial |H(e^{j\omega})| / \partial m_i)| = \\ &= \sum_{i=1}^q |\hat{S}_{m_i} |H| | \end{aligned} \quad (2.4)$$

Estamos, ahora, en situación de definir una función de la frecuencia, a la que denominaremos $M_{|H|}$, como :

$$M_{|H|}(f_i) = \sum_{i=1}^q |\hat{S}_{m_i}| |H| \quad (2.5)$$

Esta función, evaluada convenientemente, nos proporcionará una medida de la sensibilidad de la magnitud de la respuesta en frecuencia de una estructura dada a la cuantización de los coeficientes.

Para evaluar esta función, existen dos posibilidades:

a). Tomar el máximo de la misma en un margen adecuado de frecuencias,

$$M_{|H|}^{\text{máx}} = \max_{i=1, \dots, N} \{M_{|H|}(f_i)\} \quad (2.6)$$

b). Tomar un valor cuadrático medio del tipo:

$$M_{|H|}^{\text{máx}} = \left(\sum_{i=1}^N |M_{|H|}(f_i)|^2 / N \right)^{\frac{1}{2}} \quad (2.7)$$

siendo N un número de muestras igualmente espaciadas dentro del espectro de $M_{|H|}(f)$.

En cualquiera de los dos casos es necesario evaluar (2.5).

La medida anterior de sensibilidad, también llamada "determinista", tiene la limitación de suponer que $\Delta|H|$ es aproximable por los términos de primer orden de un desarrollo en serie de Taylor. Evidentemente, una cuantización muy fuerte de los coeficientes de un filtro digital puede implicar la necesidad de usar más términos del desarrollo de Taylor, es decir, funciones de sensibilidad de orden superior [7].

Para evitar esta limitación, es conveniente emplear dos medidas de tipo experimental y que no son más que el resultado de la comparación exhaustiva de la respuesta en frecuencia real, obtenida una vez que se ha realizado la cuantización, con la respuesta deseada, es decir con la respuesta en frecuencia del filtro cuando los coeficientes se representan en máxima precisión.

Estas dos medidas son el Error de Pico, E_p , y la raíz cuadrada del Error cuadrático medio, E_{rms} .

El Error de Pico, E_p , se define:

$$E_p = \text{máx} \{ |H_Q(f_i) - H_I(f_i)| \} \quad (2.8)$$

Y E_{rms} viene dado por :

$$E_{rms} = \left\{ (1/N) \sum_{i=1}^N |H_Q(f_i) - H_I(f_i)|^2 \right\}^{\frac{1}{2}} \quad (2.9)$$

donde, f_i son puntos de frecuencia equiespaciados dentro de la banda de interés, $H_Q(f)$ es la magnitud de la respuesta en frecuencia con los coeficientes cuantizados y $H_I(f)$ es la magnitud de la respuesta en frecuencia con los coeficientes representados en máxima precisión.

2.2.- MATRIZ DE TRANSFERENCIA DE UNA ESTRUCTURA DE BLOQUES

Según hemos visto en el Capítulo anterior, una estructura de bloques puede quedar descrita por su realización en bloque de estados dada por las ecuaciones (1.28).

La matriz de transferencia $\hat{H}(z)$, característica de esa estructura de bloques, se obtendrá aplicando transformada-z a dichas ecuaciones y suponiendo condiciones iniciales nulas. Así:

$$\begin{aligned} z \cdot \hat{\tilde{X}}(z) &= \hat{\underline{A}} \cdot \hat{\tilde{X}}(z) + \hat{\underline{B}} \cdot \hat{\tilde{U}}(z) \\ \hat{\tilde{Y}}(z) &= \hat{\underline{C}} \cdot \hat{\tilde{X}}(z) + \hat{\underline{D}} \cdot \hat{\tilde{U}}(z) \end{aligned} \tag{2.10}$$

siendo:

$$\begin{aligned} \hat{\underline{Y}}(z) &= [\hat{Y}_1(z) \quad \hat{Y}_2(z) \quad \dots \quad \hat{Y}_L(z)]^t \\ \hat{Y}_i(z) &= \sum_{k=0}^{\infty} y(kL+i-1)z^{-k} \end{aligned} \quad (2.11)$$

$$\begin{aligned} \hat{\underline{U}}(z) &= [\hat{U}_1(z) \quad \hat{U}_2(z) \quad \dots \quad \hat{U}_L(z)]^t \\ \hat{U}_i(z) &= \sum_{k=0}^{\infty} u(kL+i-1)z^{-k} \end{aligned}$$

$$\begin{aligned} \hat{\underline{X}}(z) &= [\hat{X}_1(z) \quad \hat{X}_2(z) \quad \dots \quad \hat{X}_M(z)]^t \\ \hat{X}_i(z) &= \sum_{k=0}^{\infty} x_i(kL)z^{-k} \end{aligned}$$

Operando puede obtenerse fácilmente la relación entre $\hat{\underline{Y}}(z)$ y $\hat{\underline{U}}(z)$, transformadas-z de los vectores de salida y entrada, respectivamente y, por tanto, la matriz de transferencia,

$$\hat{\underline{Y}}(z) = [\hat{\underline{C}} \cdot [z\underline{I} - \hat{\underline{A}}]^{-1} \cdot \hat{\underline{B}} + \hat{\underline{D}}] \cdot \hat{\underline{U}}(z) \quad (2.12)$$

con lo que,

$$\hat{\underline{H}}(z) = \hat{\underline{C}} \cdot [z\underline{I} - \hat{\underline{A}}]^{-1} \cdot \hat{\underline{B}} + \hat{\underline{D}} \quad (2.13)$$

Análogamente [8], si la descripción del filtro de bloques es a través de la representación matricial de la convolución, la ecuación característica del mismo viene dada por (1.20). Dicha ecuación puede dar lugar a distintas estructuras de bloque de estado:

a). Para el caso $L \geq M$, dicha ecuación se reduce a (1.16), que es una ecuación en diferencias de primer orden de la cual se han obtenido una gran cantidad de representaciones en variables de estado. Por ejemplo, teniendo en cuenta las relaciones (1.17) y (1.18), puede obtenerse la siguiente representación,

$$\begin{aligned}\hat{\tilde{x}}(k+1) &= -\underline{B}_0^{-1}\underline{B}_1 \cdot \hat{\tilde{x}}(k) + \underline{H}_1 \cdot \hat{u}(k) \\ \hat{\tilde{y}}(k) &= \hat{\tilde{x}}(k) + \underline{H}_0 \cdot \hat{u}(k)\end{aligned}\tag{2.14}$$

En este caso vemos, pues, como se comentó en el Capítulo anterior, que el número de variables de estado empleado es L y, por tanto, es superior al orden del filtro.

Siguiendo el mismo procedimiento y aplicando transformada- z a las ecuaciones (2.14), la matriz de transferencia característica de la estructura será:

$$\hat{\underline{H}}(z) = [z\underline{I} + \underline{B}_0^{-1}\underline{B}_1]^{-1} \cdot \underline{H}_1 + \underline{H}_0\tag{2.15}$$

A través de distintas técnicas Mitra y Gnanasekaran [9] proponen distintas estructuras de bloque de estado, todas ellas de orden L.

b). Para el caso en el que $L < M$, y $M = N$ sin pérdida de generalidad, de la ecuación (1.20) puede obtenerse la siguiente representación en el espacio de estados [5],

$$\hat{\underline{x}}'(k+1) = \hat{\underline{A}}' \cdot \hat{\underline{x}}'(k) + \hat{\underline{B}}' \cdot \hat{\underline{u}}(k) \quad (2.16)$$

$$\hat{\underline{y}}(k) = \hat{\underline{C}}' \cdot \hat{\underline{x}}'(k) + \hat{\underline{D}}' \cdot \hat{\underline{u}}(k)$$

siendo,

$$\hat{\underline{A}}' = \begin{bmatrix} 0 & \underline{I} & 0 & \dots & 0 \\ 0 & 0 & \underline{I} & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -\underline{B}_p \underline{B}_0^{-1} & -\underline{B}_{p-1} \underline{B}_0^{-1} & \cdot & \dots & -\underline{B}_1 \underline{B}_0^{-1} \end{bmatrix} \quad pL \cdot pL$$

$$\hat{\underline{B}}' = [\underline{0} \quad \underline{0} \quad \underline{0} \quad \dots \quad \underline{I}]^t \quad pL \cdot L$$

$$\hat{\underline{C}}' = [\hat{\underline{H}}_p \quad \hat{\underline{H}}_{p-1} \quad \dots \quad \hat{\underline{H}}_1] \quad (2.17)$$

$$\hat{\underline{D}}' = \underline{H}_0 = \underline{A}_0 \underline{B}_0^{-1} \quad L \cdot pL$$

$$\hat{H}_i = (\underline{A}_i - \underline{H}_0 \underline{B}_i) \cdot \underline{B}_0^{-1} \quad i=1, \dots, p$$

y $\hat{x}(k)$ es un vector de estados de longitud pL .

En el caso de que $pL > M$, $(pL-M)$ variables de estado serán redundantes. Y la matriz función de transferencia será:

$$\hat{H}(z) = [\hat{C}' [z\underline{I} - \hat{A}]^{-1} \cdot \hat{B}' + \hat{D}'] \quad (2.18)$$

En cualquier caso, se tiene una matriz cuadrada de orden $L \cdot L$ que estará directamente relacionada con la función de transferencia del filtro escalar correspondiente, $H(z)$. Encontremos dicha relación.

Para la estructura de bloques, se cumplirá en general:

$$\hat{Y}(z) = \hat{H}(z) \cdot \hat{U}(z) \quad (2.19)$$

Para la estructura escalar correspondiente se cumplirá,

$$Y(z) = H(z) \cdot U(z) \quad (2.20)$$

Evidentemente,

$$Y(z) = \sum_{i=1}^L z^{-i+1} \cdot \hat{Y}_i(z^L) = \sum_{i=1}^L z^{-i+1} \sum_{k=0}^{\infty} y(kL+i-1) z^{-kL} =$$

$$= \sum_{j=0}^{L-1} \sum_{k=0}^{\infty} y(kL+j)z^{-(kL+j)} = \sum_{n=0}^{\infty} y(n)z^{-n} \quad (2.21)$$

por lo que,

$$Y(z) = \sum_{i=1}^L z^{-i+1} \hat{Y}_i(z^L) \quad (2.22)$$

$$U(z) = \sum_{i=1}^L z^{-i+1} \hat{U}_i(z^L)$$

siendo $Y(z)$ y $U(z)$ las transformadas- z de las señales escalares de salida y entrada respectivamente, como ha podido deducirse de (2.20).

Ahora bien, los $\hat{Y}_i(z)$ y $\hat{U}_i(z)$ no son conocidos "a priori" y no resulta tan fácil calcular $H(z)$.

Se puede emplear, para el cálculo de $H(z)$, un procedimiento alternativo, que será el que utilicemos.

Supongamos que se introduce en el filtro la entrada impulsiva en el instante $n=j$. En ese caso,

$$u(n)=0 \quad , \quad n \neq j \quad n=0,1,\dots$$

$$u(n)=1 \quad , \quad n=j$$

luego

$$\hat{U}_{j+1}(z) = 1 \quad (2.23a)$$

$$\hat{U}_i(z) = 0 \quad , \quad i \neq j+1$$

y de (2.19) se tendrá :

$$Y_i(z) = \sum_{k=1}^L \hat{H}_{ik}(z) \cdot \hat{U}_k(z) = \hat{H}_{ij+1}(z) \quad \begin{matrix} i=1, \dots, L \\ j=0, \dots, L \end{matrix} \quad (2.23b)$$

siendo $\hat{H}_{lm}(z)$ un elemento de $\hat{H}(z)$.

La salida escalar vendrá dada por:

$$Y(z) = \sum_{i=1}^L z^{-i+1} \hat{H}_{ij+1}(z^L) = z^{-j} \cdot H(z) \quad (2.24)$$

De donde despejando $H(z)$ y haciendo un cambio de subíndices se obtiene,

$$H(z) = \sum_{i=1}^L z^{k-i} \cdot \hat{H}_{ik}(z^L) \quad (2.25)$$

$k=1, \dots, L$

Fórmula que nos da la función escalar a partir de los elementos de una cualquiera de las columnas de $\hat{H}(z)$. De igual forma puede obtenerse una fórmula semejante en función de los elementos de una fila cualquiera de $\hat{H}(z)$. Ambas fórmulas son obtenidas en [4] de forma general.

2.3.- SENSIBILIDAD DE LA RESPUESTA EN FRECUENCIA
DE UN FILTRO REALIZADO CON ESTRUCTURAS
DE BLOQUES.

Abordamos en este apartado el cálculo de la sensibilidad de la magnitud de la respuesta en frecuencia de un filtro realizado con estructuras de bloques. Este cálculo ha sido ampliamente tratado en [8] y ahora lo particularizaremos al caso de estructuras de segundo orden, encontrando las fórmulas explícitas de medida de la misma.

Supongamos la estructura escalar de segundo orden de la Figura II.1. Su descripción de estados tendrá la forma :

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \cdot u(n) \quad (2.26)$$

$$y(n) = [c_1 \quad c_2] \cdot \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + d \cdot u(n)$$

y su función de transferencia será de la forma,

$$H(z) = K \cdot (a_0 + a_1 z^{-1} + a_2 z^{-2}) / (1 + p_1 z^{-1} + p_2 z^{-2}) \quad (2.27)$$

donde,

$$\underline{A} = \begin{bmatrix} -p_1 & -p_2 \\ 1 & 0 \end{bmatrix} \quad (2.28)$$

$$\underline{B} = \begin{bmatrix} K \\ 0 \end{bmatrix}$$

$$\underline{C} = [a_1 - a_0 p_1 \quad a_2 - a_0 p_2]$$

$$d = a_0 K$$

A dicha realización escalar, le corresponderá una realización de bloques de orden L , con $q=4+4L+L^2$ coeficientes, representada en la Figura II.2, y cuyas ecuaciones de estado serán :

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} \\ \hat{a}_{21} & \hat{a}_{22} \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} \hat{b}_{11} & \hat{b}_{12} & \dots & \hat{b}_{1L} \\ \hat{b}_{21} & \hat{b}_{22} & \dots & \hat{b}_{2L} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}_1(k) \\ \hat{u}_2(k) \\ \vdots \\ \hat{u}_L(k) \end{bmatrix}$$

$$\begin{bmatrix} \hat{y}_1(k) \\ \vdots \\ \hat{y}_L(k) \end{bmatrix} = \begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} \\ \vdots & \vdots \\ \hat{c}_{L1} & \hat{c}_{L2} \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} \hat{d}_{11} & \hat{d}_{12} & \dots & \hat{d}_{1L} \\ \vdots & \vdots & \dots & \vdots \\ \hat{d}_{L1} & \hat{d}_{L2} & \dots & \hat{d}_{LL} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}_1(k) \\ \vdots \\ \hat{u}_L(k) \end{bmatrix}$$

(2.29)

y cuya matriz de transferencia vendrá dada por (2.13) y que por comodidad repetimos,

$$\hat{H}(z) = \hat{C} \cdot [zI - \hat{A}]^{-1} \cdot \hat{B} + \hat{D}$$

De la realización escalar se obtiene fácilmente la de bloques sin más que aplicar las relaciones (1.29) y luego, sustituyendo en (2.13), se obtendrá para cada elemento ij de $\hat{H}(z)$,

$$\begin{aligned} \hat{H}_{ij}(z) = & (1/|\Delta|) \{ \hat{d}_{ij} z^2 + z [\hat{c}_{i1} \hat{b}_{1j} + \hat{b}_{2j} \hat{c}_{i2} - \hat{d}_{ij} (\hat{a}_{11} + \hat{a}_{22})] + \\ & + (\hat{a}_{11} \hat{a}_{22} - \hat{a}_{21} \hat{a}_{12}) \hat{d}_{ij} + \hat{a}_{21} \hat{b}_{1j} \hat{c}_{i2} + \hat{a}_{12} \hat{b}_{2j} \hat{c}_{i1} - \\ & - \hat{a}_{22} \hat{b}_{1j} \hat{c}_{i1} - \hat{a}_{11} \hat{b}_{2j} \hat{c}_{i2} \} \end{aligned} \quad (2.30)$$

siendo

$$|\Delta| = z^2 - (\hat{a}_{11} + \hat{a}_{22})z + \hat{a}_{11} \hat{a}_{22} - \hat{a}_{21} \hat{a}_{12} \quad (2.31)$$

Con esto se tiene en general para $\hat{H}_{ij}(z)$,

$$\begin{aligned} \hat{H}_{ij}(z) = & \left(\sum_{k=0}^2 a_k^{ij} \cdot z^{-k} \right) / \left(\sum_{k=0}^2 b_k^{ij} \cdot z^{-k} \right) \\ & i, j = 1, \dots, L \end{aligned} \quad (2.32)$$

donde a_k^{ij} y b_k^{ij} son función de los q coeficientes de la estructura de bloques.

Como lo que se pretende es calcular la sensibilidad de la magnitud de la respuesta en frecuencia escalar, $|H(e^{j\omega})|$, respecto a la variación de los coeficientes de la estructura de bloques, habrá que tener en cuenta (2.25) para obtener la función sensibilidad definida en (2.5).

Encontremos, en primer lugar, una expresión para la sensibilidad (2.2).

Es fácil demostrar que si H es una función compleja, se cumple :

$$S_{m_i} |H| = \partial |H| / \partial m_i = (1/|H|) \cdot \text{Re}\{H^* \cdot \partial H / \partial m_i\} \quad (2.33)$$

donde H^* es la conjugada de H .

Por otro lado, y con ayuda de (2.25), se tiene una expresión para la respuesta en frecuencia del filtro dada por:

$$|H(e^{j\omega})|^2 = \sum_{m=1}^L |\hat{H}_{mk}(e^{j\omega L})|^2 + \sum_{m=1}^L \sum_{\substack{p=1 \\ m \neq p}}^L \text{Re} \{ \hat{H}_{mk}(e^{j\omega L}) \cdot \hat{H}_{pk}^*(e^{j\omega L}) \cdot e^{(p-m)j\omega} \} \quad (2.34)$$

Sustituyendo $H(z)$ por su valor en (2.25), con $z=e^{j\omega}$ y haciendo operaciones [8], se tendrá para (2.33),

$$S_{m_i} |H| = (1/|H|) \left[\sum_{m=1}^L |\hat{H}_{mk}| \cdot S_{m_i} |\hat{H}_{mk}| \right] + (1/|H|) \cdot \text{Re} \left\{ \sum_{m=1}^L \sum_{\substack{p=1 \\ m \neq p}}^L e^{(m-p)j\omega} \cdot \hat{H}_{mk}^* \cdot \partial \hat{H}_{pk} / \partial m_i \right\} \quad (2.35)$$

donde es necesario calcular,

$$S_{m_i} |\hat{H}_{ij}| = \partial |\hat{H}_{ij}| / \partial m_i \quad (2.36)$$

Teniendo en cuenta la expresión (2.32) y haciendo $z=e^{j\omega}$, se podrá deducir [8]:

$$\frac{\partial \hat{H}_{ij}}{\partial m_1} = \hat{H}_{ij} \left[\sum_{k=0}^2 e^{-j\omega kL} \left(\frac{l_{kl}(ij)}{\sum_{t=0}^2 a_t^{ij} \cdot e^{-j\omega Lt}} - \frac{g_{kl}(ij)}{\sum_{t=0}^2 b_t^{ij} \cdot e^{-j\omega Lt}} \right) \right] \quad (2.37)$$

$k=1, \dots, L$

donde se han definido las matrices $\underline{l}(ij)$ y $\underline{g}(ij)$, de orden $3 \cdot q$, de la siguiente forma:

$$\frac{\partial \hat{H}_{ij}}{\partial m_1} = \sum_{k=0}^2 (\frac{\partial \hat{H}_{ij}}{\partial a_k^{ij}}) \cdot l_{kl}(ij) + \sum_{k=0}^2 (\frac{\partial \hat{H}_{ij}}{\partial b_k^{ij}}) \cdot g_{kl}(ij) \quad (2.38)$$

es decir,

$$l_{kl}(ij) = \frac{\partial a_k^{ij}}{\partial m_1} \quad (2.39)$$

$$g_{kl}(ij) = \frac{\partial b_k^{ij}}{\partial m_1} \quad \begin{matrix} l=1, \dots, q \\ k=0, 1, 2 \end{matrix}$$

En concreto, como de (2.32) se observa que independientemente del elemento ij de $\hat{H}(z)$, el denominador de $\hat{H}_{ij}(z)$ es siempre el mismo, podemos afirmar que las matrices $\underline{g}(ij)$ serán todas iguales, por lo que hacemos, $\underline{g}(ij) = \underline{g}$.

En las Tablas 2.I y 2.II se encuentran detallados los elementos de ambas matrices, $\underline{1}^{(ij)}$ y g . y en (2.37), y teniendo en cuenta la definición (2.5), tendremos una expresión para la función SENSIBILIDAD, función de la frecuencia, cuya evaluación nos dará la sensibilidad de la estructura a la cuantización de los coeficientes.

2.4. PROGRAMA "NUV" DE MEDIDA DE LA SENSIBILIDAD.

Procederemos a continuación a aplicar los estudios anteriores a casos concretos, evaluando la sensibilidad de distintos filtros de bloque a la cuantización de los coeficientes.

Para ello lo primero que se ha hecho es adaptar a FORTRAN 77, para correr sobre un VAX11/785, el programa de diseño de filtros TAN empleado en [10], y que permite diseñar filtros paso bajo, según una de las cuatro aproximaciones clásicas: Butterworth, Chebyshev, Chebyshev Inverso y Elíptica. Este programa proporciona los coeficientes de las estructuras directas de segundo orden, que conectadas en cascada proporcionan el filtro en cuestión. El mencionado programa se ha incorporado como subrutina al programa principal de evaluación de la sensibilidad.

Como último paso, se ha realizado, también en FORTRAN 77, un programa al que se ha denominado "NUV". Este programa permite evaluar la sensibilidad de la respuesta en frecuencia de estructuras de bloques de segundo orden y a su vez evalúa también el error cuadrático medio de la respuesta en frecuencia de dichas estructuras cuando los coeficientes se representan con un número determinado de bits.

La representación de los coeficientes se realiza en coma flotante y la simulación de la misma se efectúa a través de la función RED(X,IB), que expresa la magnitud X con IB bits, y que corresponde a la cuantización por redondeo de los coeficientes.

La simulación se realiza transformando la parte fraccionaria del coeficiente a (B+1) bits, según el siguiente esquema:

Dado el coeficiente, X, lo representamos de la forma siguiente,

$$X = X' \cdot 2^D, \text{ con } D = \text{CEIL}(\log_2 |X|) \quad (2.40)$$

CEIL(.) es la función que da el menor entero mayor o igual a (.).

En general, $1/2 < |X'| \leq 1$.

La función que redondea X' a (B+1) bits, incluido el signo, será,

$$Q(X') = 2^{-B} \cdot \text{IFLOOR} (2^B \cdot X' + .5) \quad (2.41)$$

IFLOOR(.) es el mayor entero menor o igual a (.).

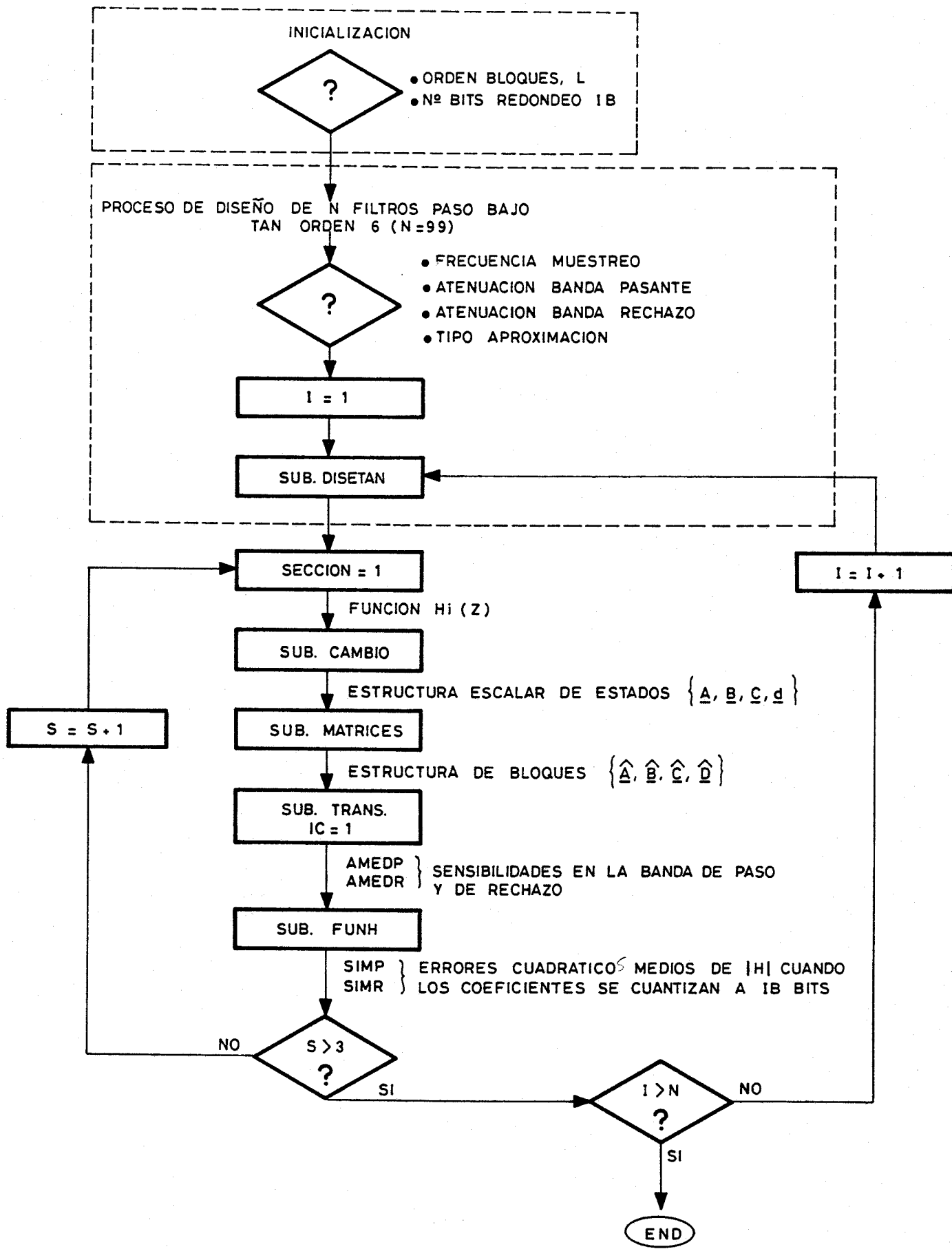
De esta forma, si

$$\begin{aligned}x' &= Q(x'), \text{ el nuevo coeficiente } x_Q \text{ ser\'a} \\x_Q &= x' \cdot 2^D. \qquad (2.42)\end{aligned}$$

El listado completo del programa NUV.FOR se incluye al final del Cap\'itulo.

Hay que hacer notar que en este programa las medidas de sensibilidad consideradas son las dadas por (2.7) y (2.9) para el error cuadr\'atico medio.

En ambos casos las funciones de la frecuencia se eval\'uan para cada secci\'on por separado en la banda de paso y en la banda de rechazo del filtro, como puede deducirse del diagrama de flujo que se presenta a continuaci\'on. Adem\'as en (2.25) y todas las ecuaciones que lo requieren se toman los elementos de la primera columna de la matriz $\hat{H}(z)$ para evaluar la sensibilidad.



PROGRAMA "NUV" : Diagrama de flujo.

2.5. RESULTADOS EXPERIMENTALES DE SIMULACION.

En concreto, hemos centrado nuestro estudio en las secciones de segundo orden que se obtienen en el diseño de filtros digitales paso de baja, empleando el programa NUV, y considerando las siguientes especificaciones:

1. Orden del filtro, 6
2. Rizado en la banda pasante, 0.2dB.
3. Atenuación en la banda de rechazo, 40dB.
4. Frecuencia de muestreo normalizada a la unidad.

Como parámetro variable de los diseños se ha considerado la frecuencia de corte ó frecuencia superior de la banda de paso, que se ha ido variando hasta conseguir 99 filtros distintos.

Así, para cada tipo de aproximación se han estudiado 99 filtros distintos, cada uno de los cuales consta de tres secciones de segundo orden. Para cada una de ellas se ha obtenido la estructura de bloques correspondiente y se ha realizado una estimación de la sensibilidad de la magnitud de la respuesta en frecuencia a la cuantización de los coeficientes, tanto en la banda de paso como en la de rechazo. Para dicha estimación se ha tomado la definición dada por (2.7), y el número de puntos equiespaciados en cada banda ha sido de 50.

Por otro lado, se ha repetido el proceso para distintos valores del orden de la estructura de bloques, incluyendo el caso escalar, $L=1$.

De esta forma, en las Figuras II.3, II.4, II.5, y II.6, se representa, para la banda pasante y para cada aproximación, dicha sensibilidad (eje de ordenadas) frente a la frecuencia de corte del filtro (eje de abcisas), para los valores $L=1,2$ y 4. Obsérvese como en cada Figura se representan las tres secciones del filtro.

De igual modo, y para la banda de rechazo, se tienen las Figuras II.7, II.8, II.9 y II.10.

De la observación de las mismas podemos hacer una serie de consideraciones:

Se pone de manifiesto en todas las Figuras un resultado ampliamente conocido, esto es, tanto en la banda pasante como en la de rechazo, se observa el buen comportamiento de la estructura directa [10] frente a la cuantización, en los casos de filtros de banda pasante de anchura media. Si bien para filtros de banda estrecha los valores de las sensibilidades aumentan considerablemente.

La sensibilidad de las estructuras en ambas bandas disminuye conforme aumenta el ancho de banda. Ello hace pensar que la degradación que se presente al cuantizar los coeficientes también será mayor para filtros de banda pasante estrecha.

En cuanto a la influencia del orden de la estructura de bloques sobre la sensibilidad, que es lo que verdaderamente ocupa nuestra atención, podemos decir que para la banda de paso (Fig. II.3 a II.6), esta influencia se deja sentir sobre todo en los filtros de banda pasante estrecha, pero que en cualquier caso, dicha influencia es positiva y el aumento del orden de la estructura de bloques contribuye a disminuir la sensibilidad.

Para filtros de banda pasante ancha, la sensibilidad es prácticamente independiente del orden de la estructura de bloques.

De todo esto se puede resaltar el hecho de que el empleo de la estructura de bloques permite mejorar la sensibilidad, esto es, reducirla, precisamente en aquellos casos en los que el comportamiento

de la estructura escalar (filtros de bandas de paso estrechas) dejaba bastante que desear.

En cuanto a la banda de rechazo puede decirse que el aumento del orden de los bloques mejora la situación para bandas pasantes estrechas y medias. Para bandas pasantes anchas no puede afirmarse lo mismo, la sensibilidad no disminuye al aumentar el orden de bloques, incluso en ocasiones aumenta.

Ahora bien, hay que considerar que recientemente [2], en un estudio completo del tema, determinista, estadístico y experimental, sobre la influencia de la cuantización de los coeficientes en la implementación de filtros digitales realizados en cascada, uno de los resultados obtenidos es que la influencia de esta cuantización sobre la banda de rechazo es despreciable para todos los tipos de aproximación. Intuitivamente esta propiedad puede interpretarse como una atenuación progresiva de la señal de entrada en cada sección de segundo orden y que, por tanto, será independiente de la fase introducida por cada sección.

Para completar el estudio anterior y con el mismo programa NUV, se ha calculado para cada uno de los filtros anteriores el error cuadrático medio de la magnitud de la respuesta en frecuencia de las secciones de segundo orden según (2.9), cuando los coeficientes se cuantizan a 6 y 9 bits.

Se incluyen las Figuras II.11 y II.12 que corresponden a la representación para la banda pasante del error cuadrático medio (eje de ordenadas) frente a la frecuencia de corte del filtro (eje de abscisas), para las aproximaciones Chebyshev Inversa y Elíptica, respectivamente, cuando los coeficientes se representan con 9 bits.

Se pone de manifiesto todo lo comentado anteriormente respecto a la mejora sobre la sensibilidad que se produce con el incremento del orden de bloques para filtros de banda pasante estrecha.

En las Figuras II.13 y II.14 se representa dicho error para la banda de rechazo, para las aproximaciones Chebyshev Inversa y Elíptica, respectivamente, cuando los coeficientes se representan con 6 bits.

Se han repetido los cálculos anteriores para distintos valores del número de bits empleados en la cuantización, poniéndose de manifiesto que los resultados anteriores se mantienen.

Un punto de vista muy ilustrativo es, desde luego, examinar la representación gráfica de la magnitud de la respuesta en frecuencia de un filtro cuando los coeficientes del mismo se someten a cuantización, y compararla con la misma cuando los coeficientes se representan en máxima precisión.

Por todo ello, se incluye a continuación en las Figuras II.15 y II.16 la representación de la banda pasante de la magnitud de la respuesta en frecuencia y la magnitud de la respuesta en frecuencia, respectivamente, de un filtro paso bajo de las siguientes características :

$$f_c = 0.0625 f_m$$

orden, 6

Rizado banda pasante, 0.2dB.

Atenuación banda rechazo, 40dB.

Aproximación Elíptica.

para el caso de que se empleen para realizarlo estructuras de bloques de orden 1, 2, 4. Lógicamente la respuesta en máxima precisión es la misma para los tres casos.

Así, para cada valor de L se representa dicha magnitud con los coeficientes en máxima precisión (H), y con los coeficientes cuantizados a 6 y 9 bits.

Para obtener dichas Figuras se ha empleado el programa de simulación SIM.FOR, cuyo listado se incluye al final del Capítulo.

Queda patente la mejora que se introduce al realizar el filtro con estructuras de bloques en lugar de emplear la estructura escalar, en lo que se refiere

a sensibilidad de la respuesta en frecuencia.

Esta mejora va siendo menos sustancial conforme aumenta el ancho de banda del filtro, lo cual puede observarse en las Figuras II.17 y II.18 donde se representan, la banda pasante y la magnitud de la respuesta en frecuencia, respectivamente, de un filtro paso bajo con las mismas características que el anterior salvo la frecuencia de corte, que en este caso es $f_c = 0.25f_m$.

2.6. REFERENCIAS.

- [1]. L.R.Rabiner and B.Gold, "Theory and applications of digital signal processing". Prentice-Hall. 1975.
- [2]. J.I.Acha, R.Robles-Díaz y A.Ayerbe, "Estructuras paralelo y cascada para realizar filtros digitales: Estudio comparativo de la sensibilidad". V Congreso de Informática y Automática. Madrid, 4-7 Mayo. pp.785-789. 1982.
- [3]. J.I.Acha, R.Robles-Díaz y A.Ayerbe, "Coefficient quantization effects in cascade and parallel form realization of low-pass digital filters". Int. Journal Electronics. Vol.54. No.4, pp.555-562. 1983.
- [4]. C.W.Barnes and S. Shinnanka, "Block shift invariance and block implementation of discrete-time filters". IEEE Trans. Circuits and Syst. Vol.CAS-27, pp.667-672. 1980.
- [5]. P. Ananthakrishna and K.Mitra, "Minimum Round-off noise realization of Block-state recursive digital filters". Circuits Syst. Signal Process. Vol.1, No.2, pp.217-231. 1982.

- [6]. P. Ananthakrishna, B.D.O. Anderson and K.Mitra, "Realization of Digital Transfer Functions using cascaded lattice and ladder block structures". Journal of the Franklin Institute. Vol.316, No.1, pp.93-124. 1983.
- [7]. R.E. Crochiere and A.V. Oppenheim, "Analysis of linear digital networks". Proc. IEEE, Vol.63, pp.581-595. 1975
- [8]. A.Ayerbe, "Sensibilidad de la magnitud de la respuesta en frecuencia de filtros digitales realizados usando estructuras de bloque". Premio IEEE Sección Española. 1983.
- [9]. S.K.Mitra and R.Gnanasekaran, "Block implementation of recursive digital filters-new structures and properties". IEEE Trans. Circuits and Syst. Vol.CAS-25, pp.200-207. 1978.
- [10]. R.Robles-Díaz, "Implementación de filtros digitales IIR en estructuras de tipo escalera y reticular". Tesis Doctoral. Universidad de Sevilla. 1981.

m	$l_{1m}(ij)$	$l_{2m}(ij)$	$l_{3m}(ij)$
1	0	$-\hat{a}_{ij}$	$\hat{a}_{22}\hat{d}_{ij}-\hat{b}_{2j}\hat{c}_{i1}$
2	0	0	$\hat{b}_{2j}\hat{c}_{i1}-\hat{a}_{21}\hat{d}_{ij}$
3	0	0	$\hat{b}_{1j}\hat{c}_{i2}-\hat{a}_{12}\hat{d}_{ij}$
4	0	$-\hat{a}_{ij}$	$\hat{a}_{11}\hat{d}_{ij}-\hat{b}_{1j}\hat{c}_{i1}$
5	0	0	0
⋮	0	0	0
4+j	0	\hat{c}_{i1}	$\hat{a}_{21}\hat{c}_{i2}-\hat{a}_{22}\hat{c}_{i1}$
⋮	⋮	⋮	⋮
4+L	0	0	0
4+L+1	0	0	0
⋮	0	0	0
4+L+j	0	\hat{c}_{i2}	$\hat{a}_{12}\hat{c}_{i1}-\hat{a}_{11}\hat{c}_{i2}$
⋮	⋮	⋮	⋮
4+2L	0	0	0
4+2L+1	0	0	0
⋮	⋮	⋮	⋮
4+2L+i	0	\hat{b}_{1j}	$\hat{a}_{12}\hat{b}_{2j}-\hat{a}_{22}\hat{b}_{1j}$
⋮	⋮	⋮	⋮
4+3L	0	0	0
4+3L+1	0	0	0
⋮	⋮	⋮	⋮
4+3L+i	0	\hat{b}_{2j}	$\hat{a}_{21}\hat{b}_{1j}-\hat{a}_{11}\hat{b}_{2j}$
⋮	⋮	⋮	⋮
4+4L	0	0	0
4+4L+1	0	0	0
⋮	1	$-\hat{a}_{11}-\hat{a}_{22}$	$\hat{a}_{11}\hat{a}_{22}-\hat{a}_{21}\hat{a}_{12}$
4+3L+Li+j	1		
4+4L+L ²	0	0	0

Tabla 2.I. Matriz $\underline{l}(ij)$

m	g_{1m}	g_{2m}	g_{3m}
1	0	-1	a_{22}
2	0	0	$-a_{21}$
3	0	0	a_{12}
4	0	-1	a_{11}
5	0	0	0
6	0	0	0
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
$4+4L+L^2$	0	0	0

Tabla 2.II. Matriz g

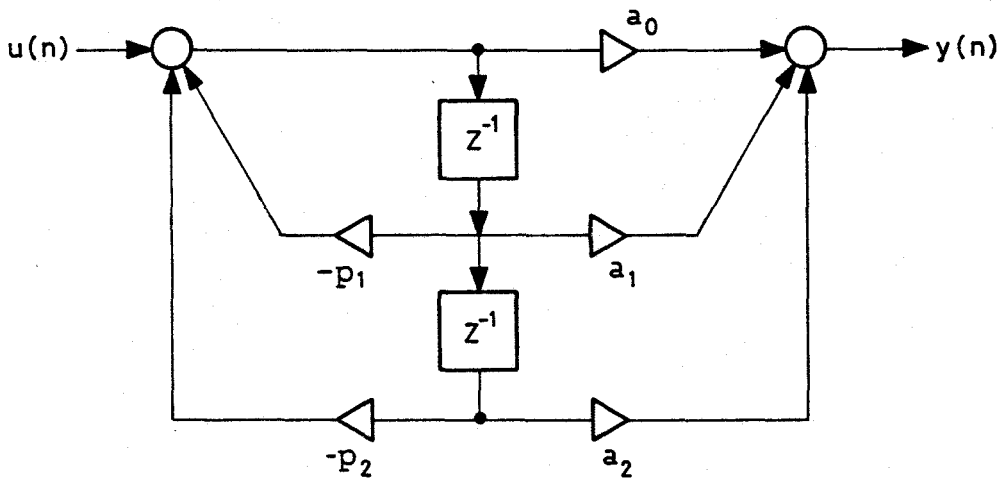


FIGURA II. 1.- REALIZACION ESCALAR DIRECTA DE UN FILTRO IIR DE SEGUNDO ORDEN

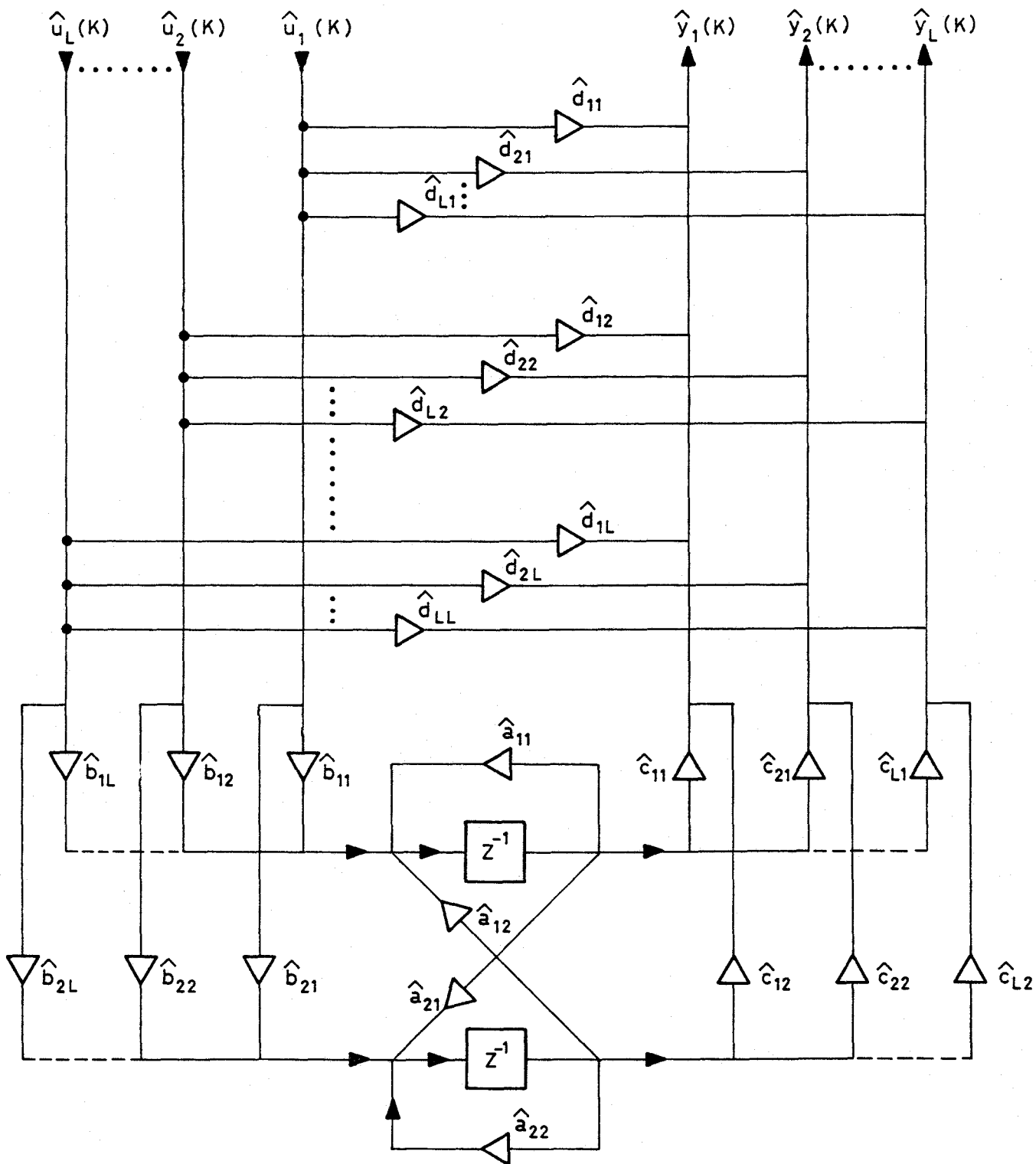


FIGURA II. 2 - REALIZACION EN ESTRUCTURA DE BLOQUES DE VARIABLE DE ESTADO DE UN FILTRO CON $M = 2$

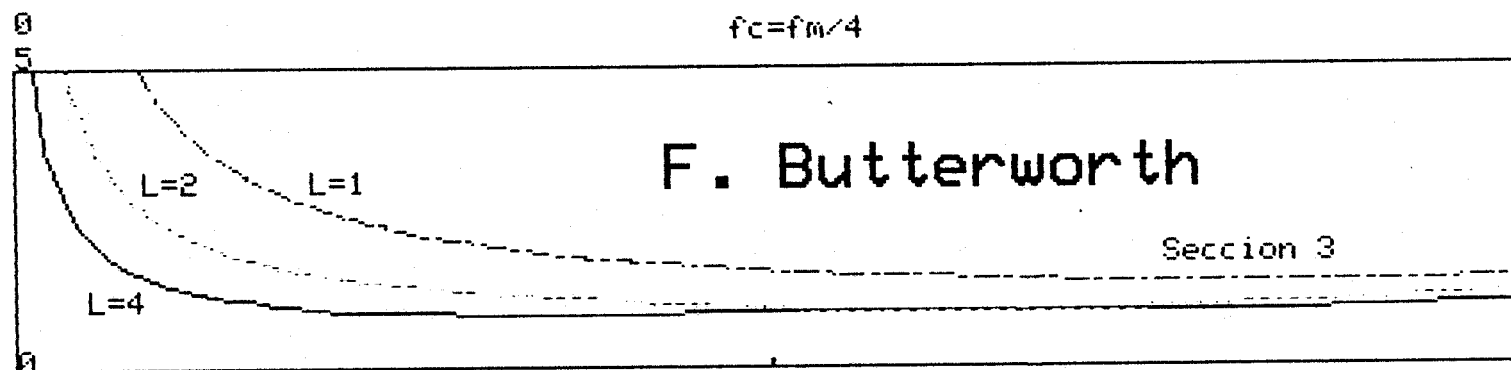
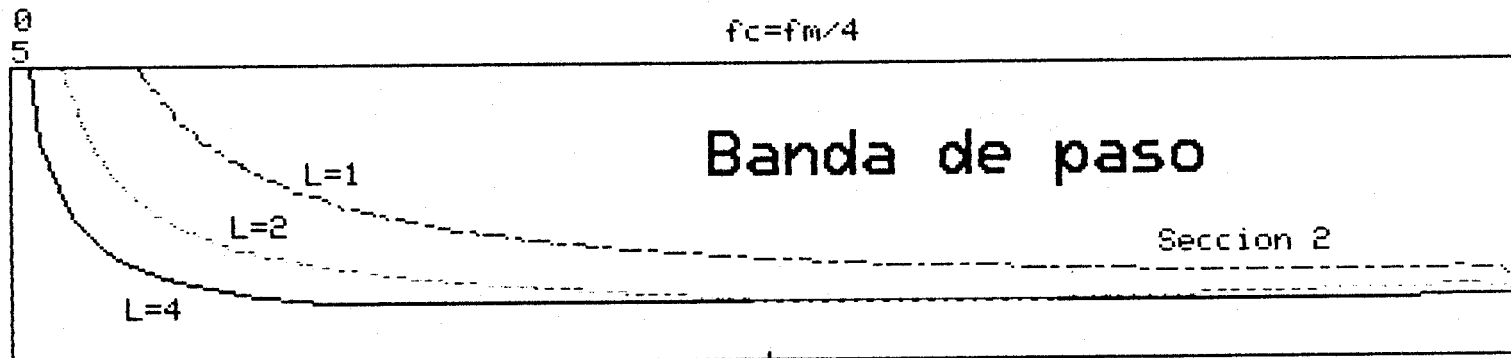
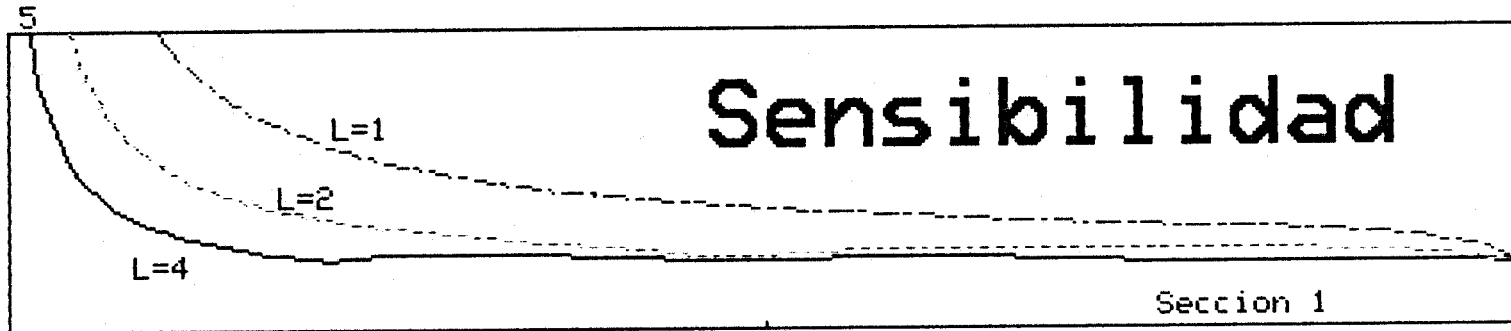


Figura II.3

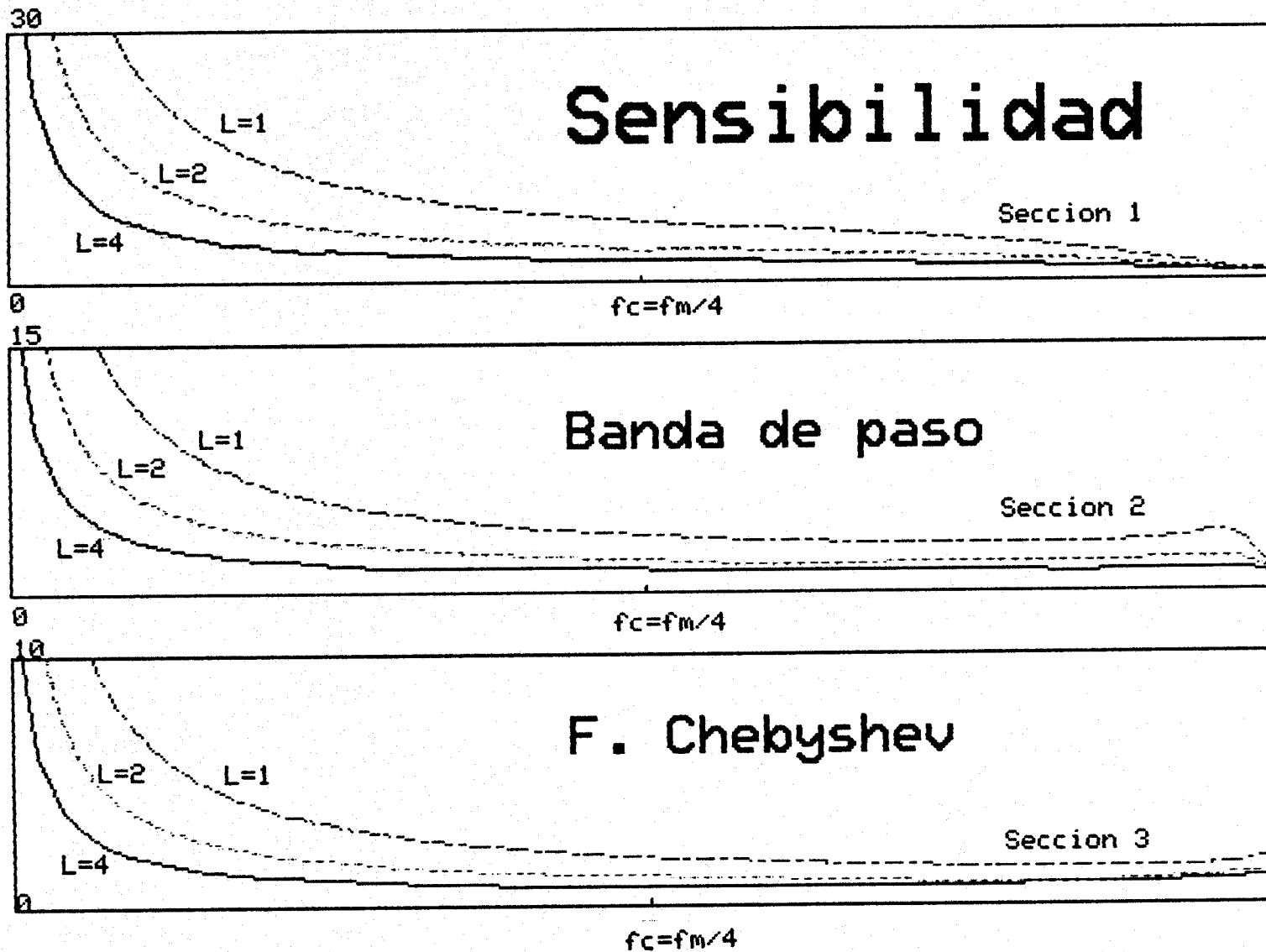
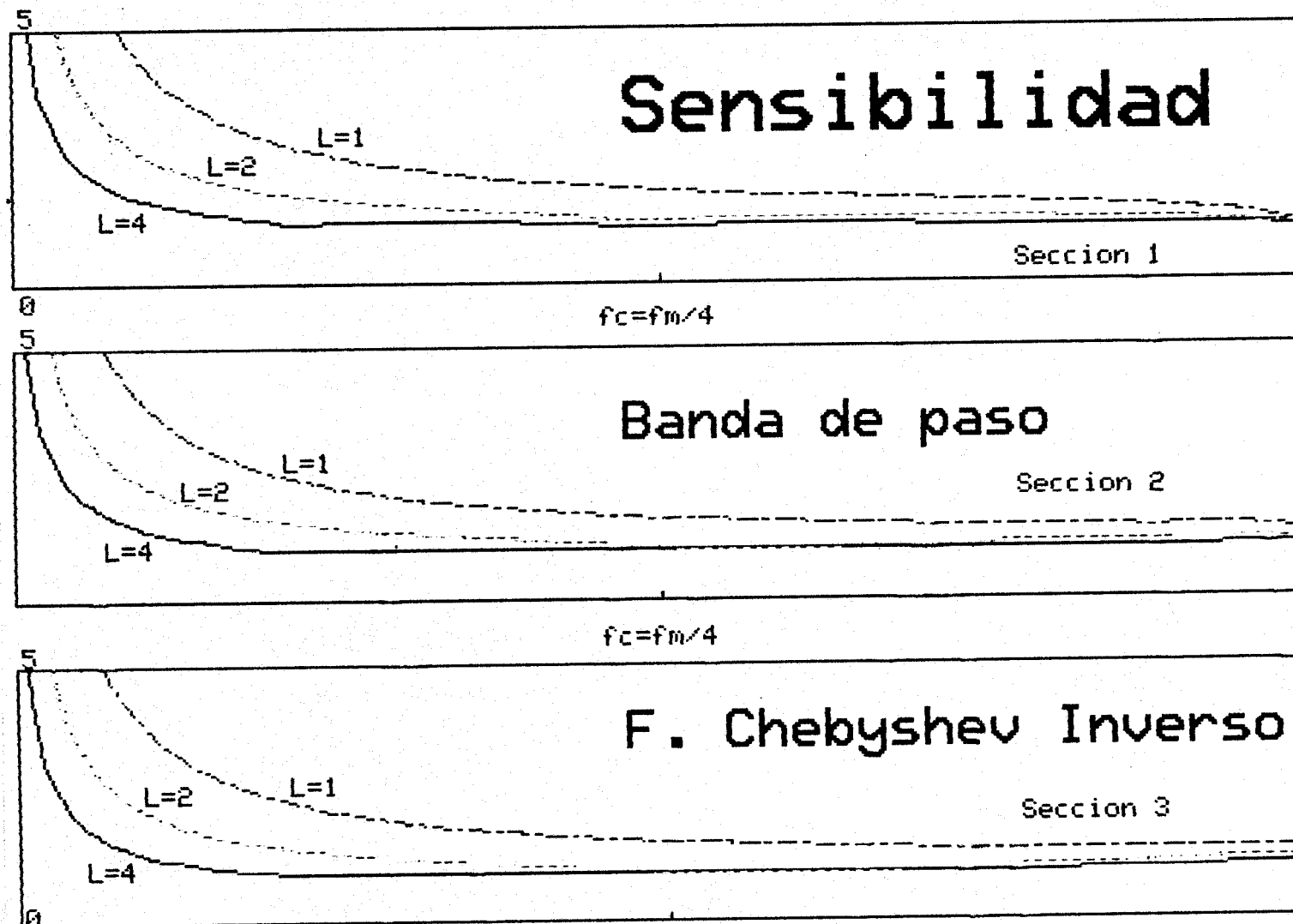


Figura II.4



$f_c = f_m/4$
Figura II.5

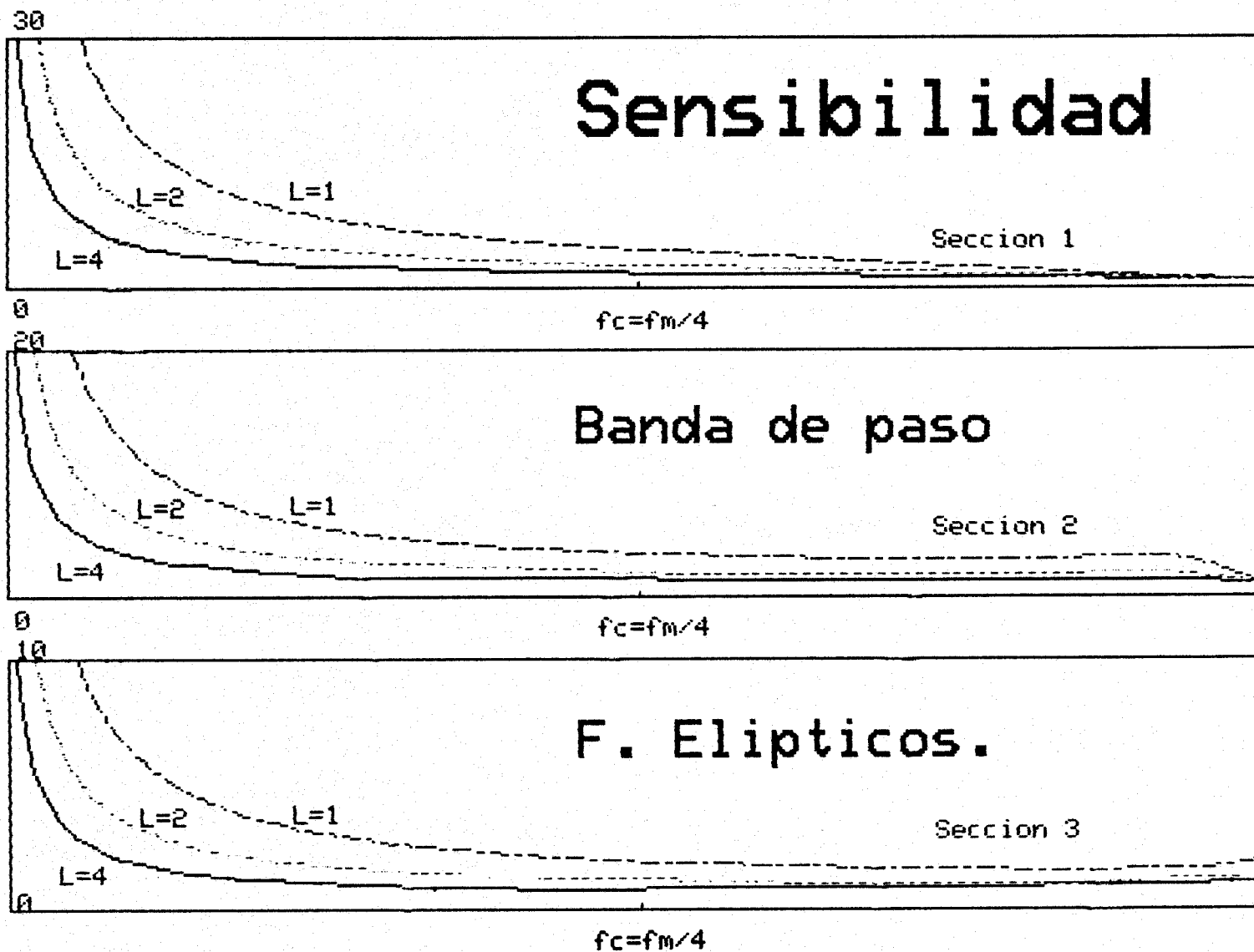


Figura II.6

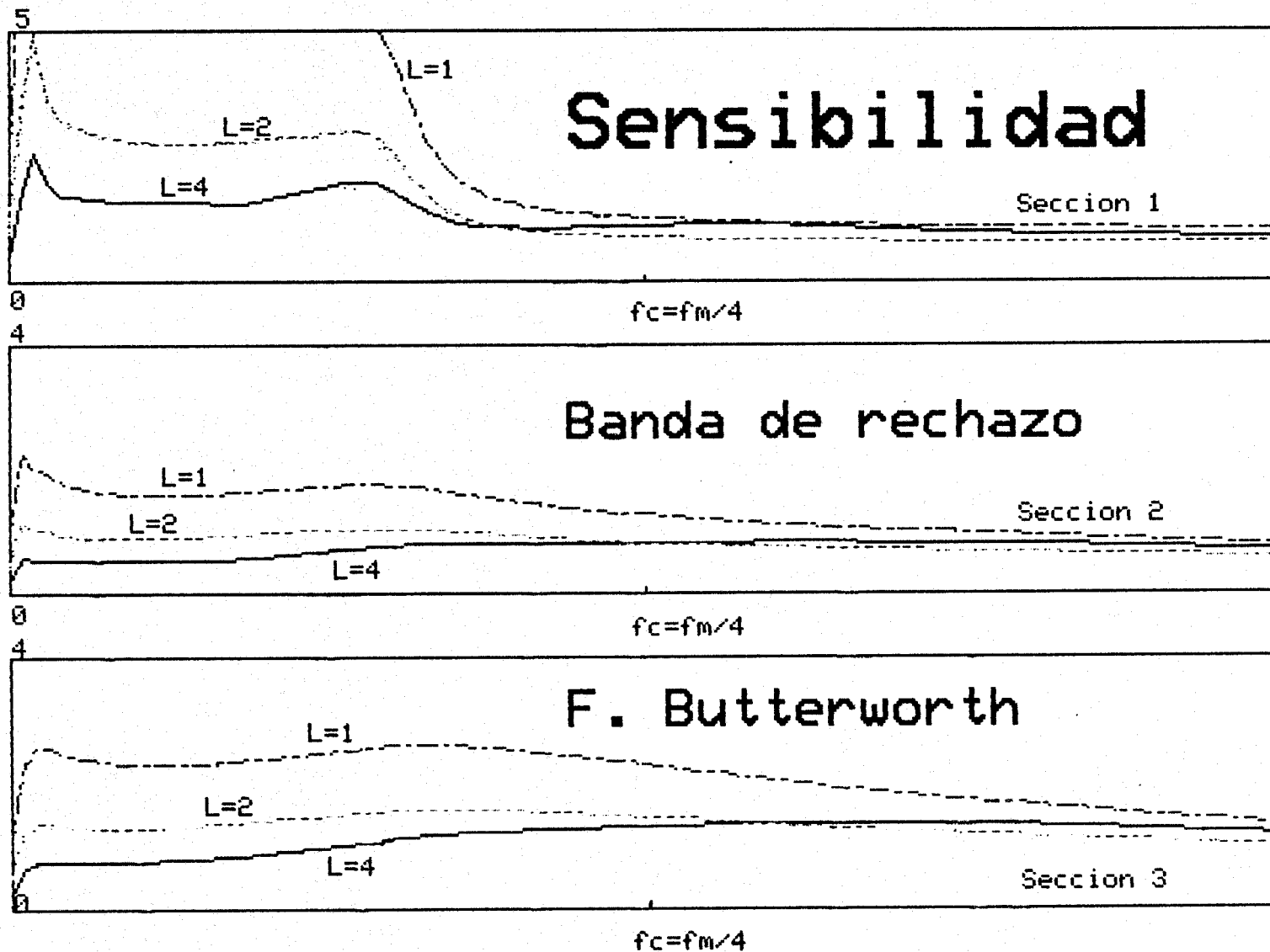


Figura II.7

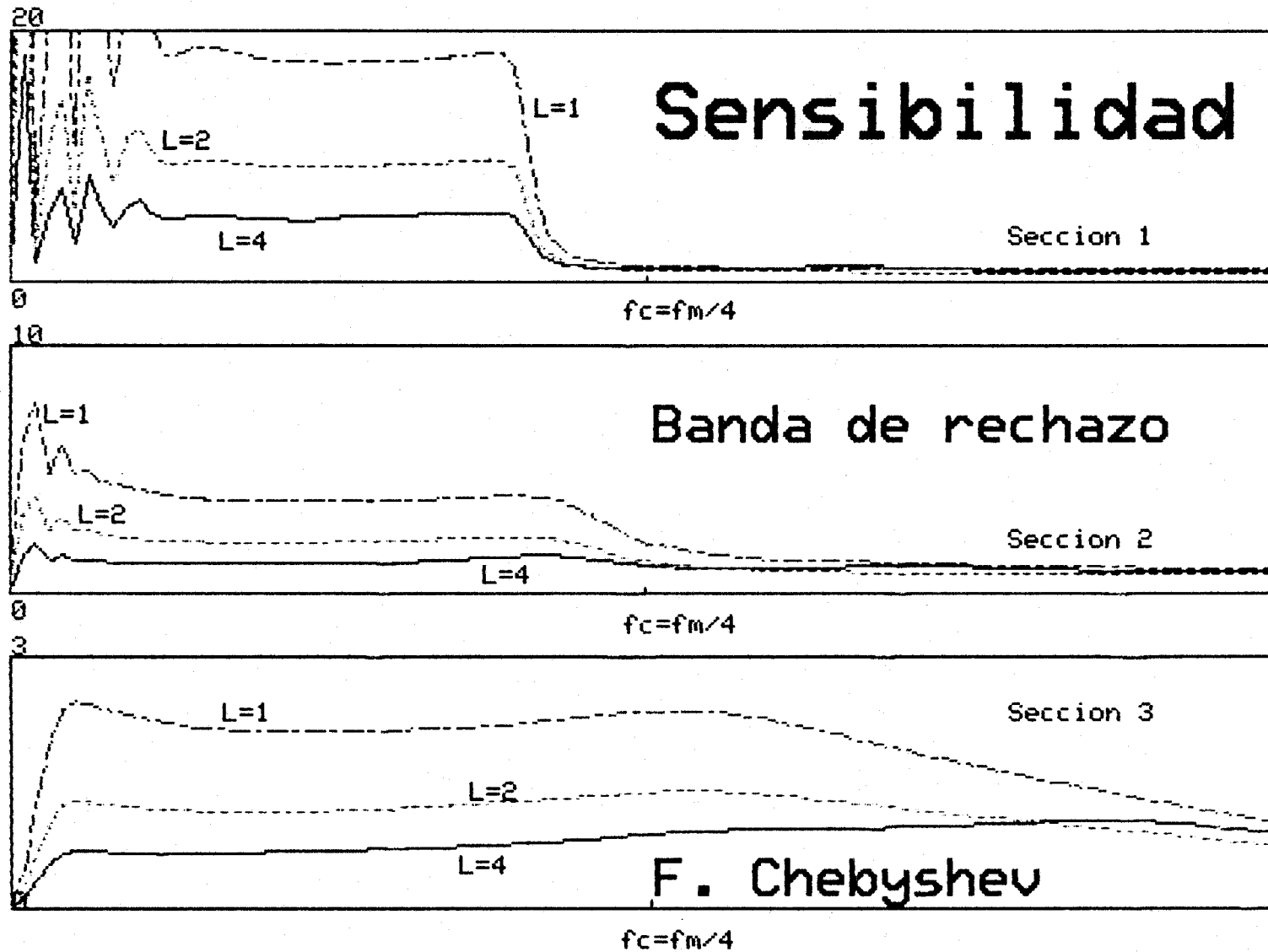


Figura II.8

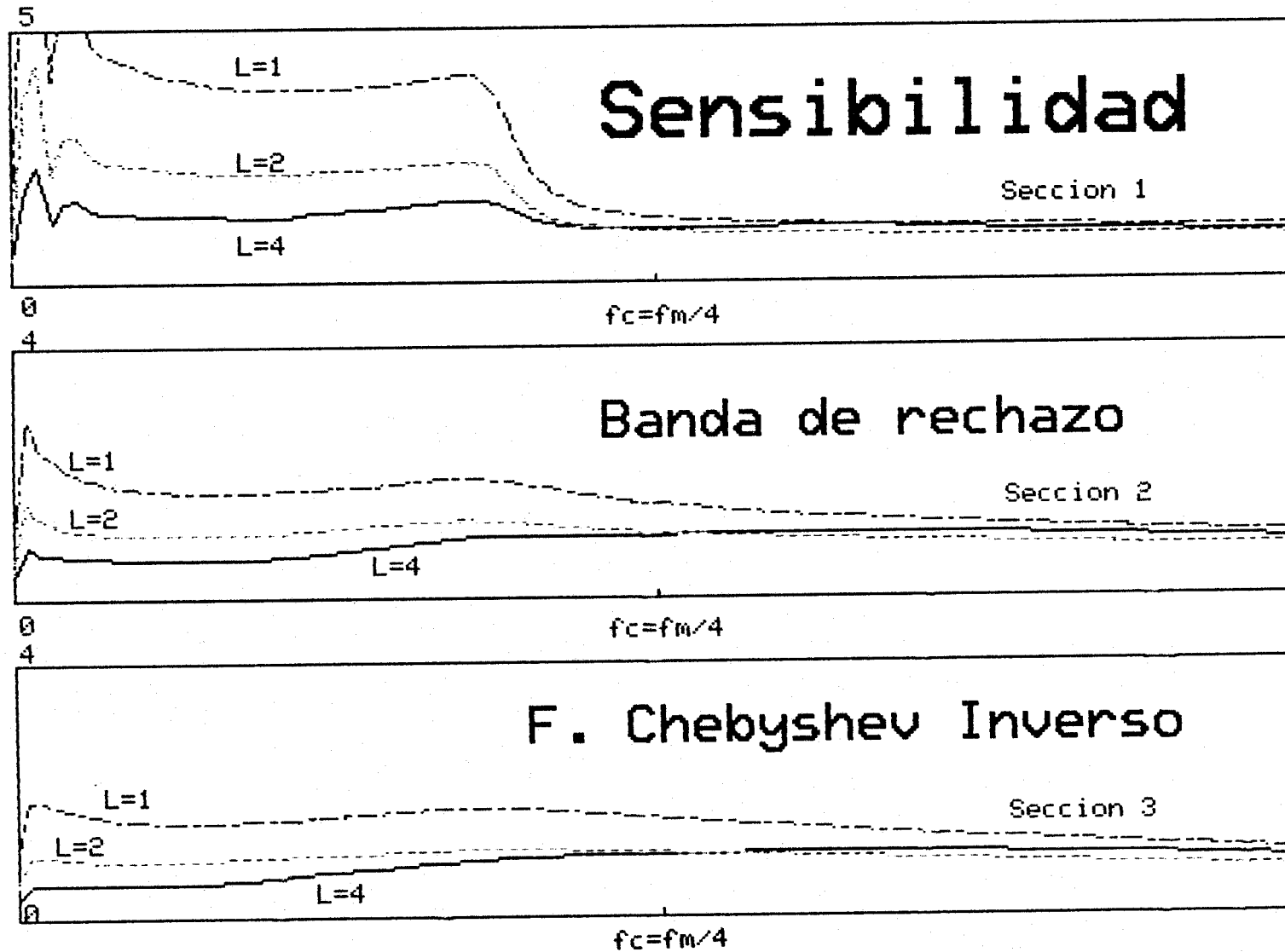


Figura II.9

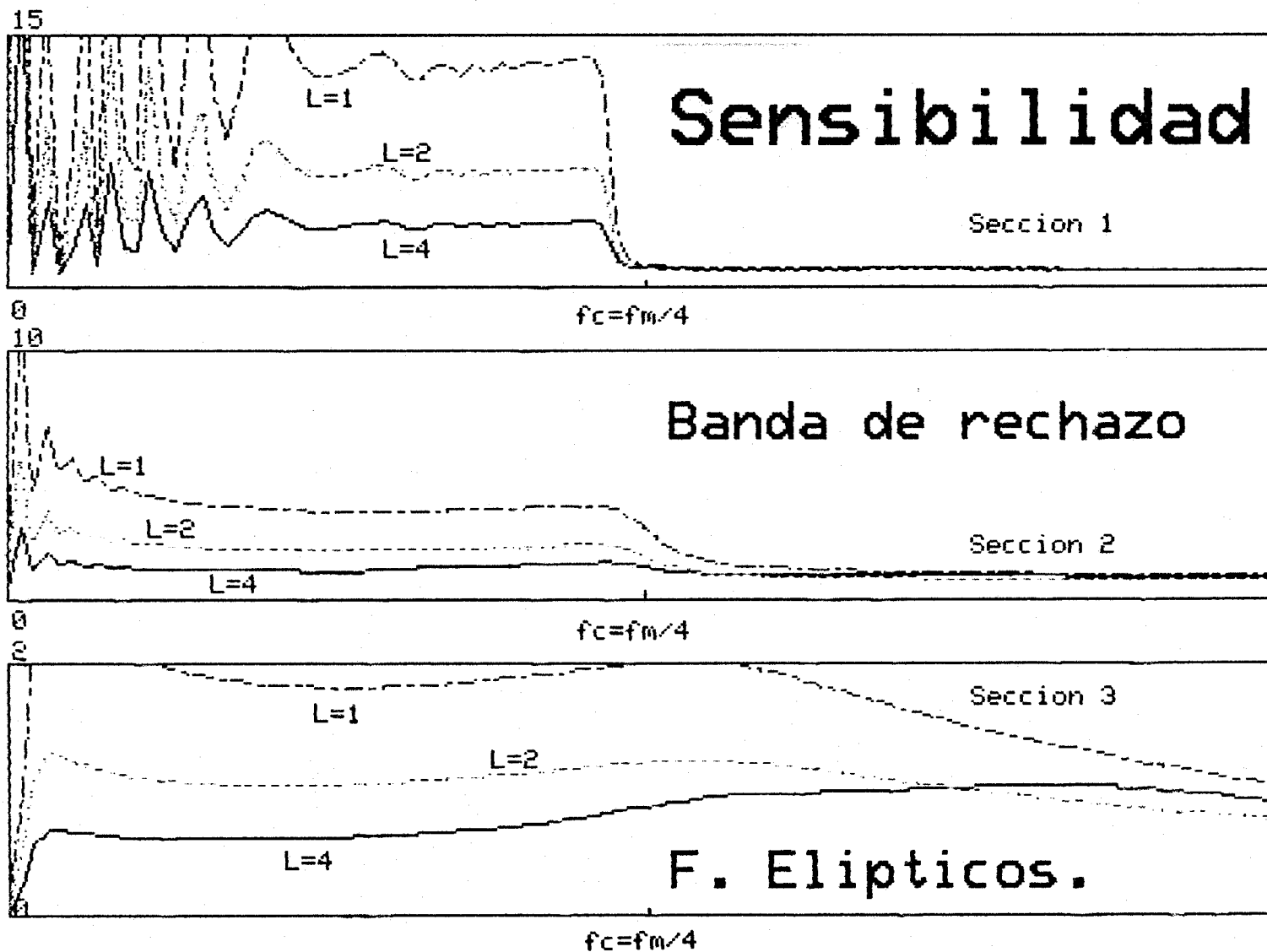
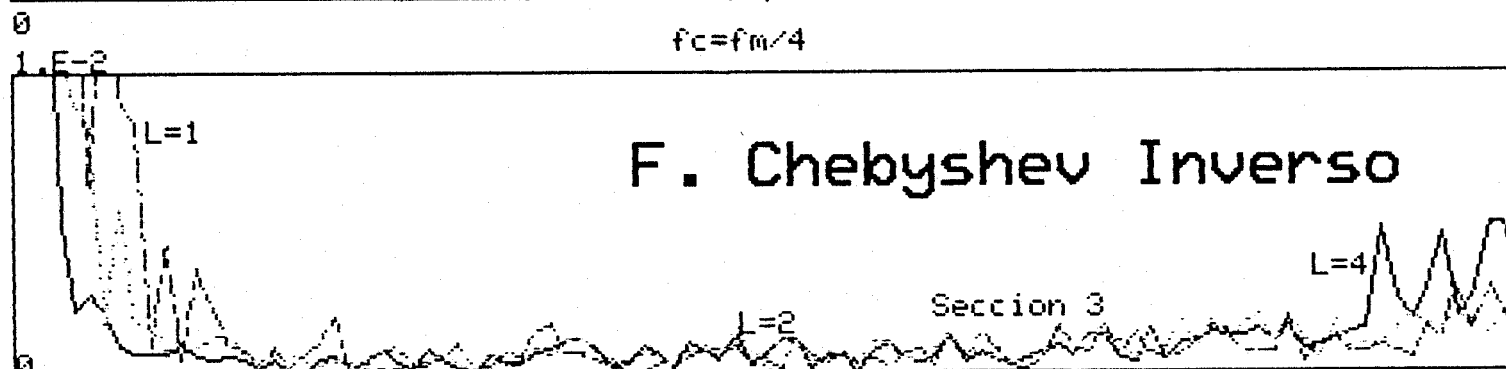
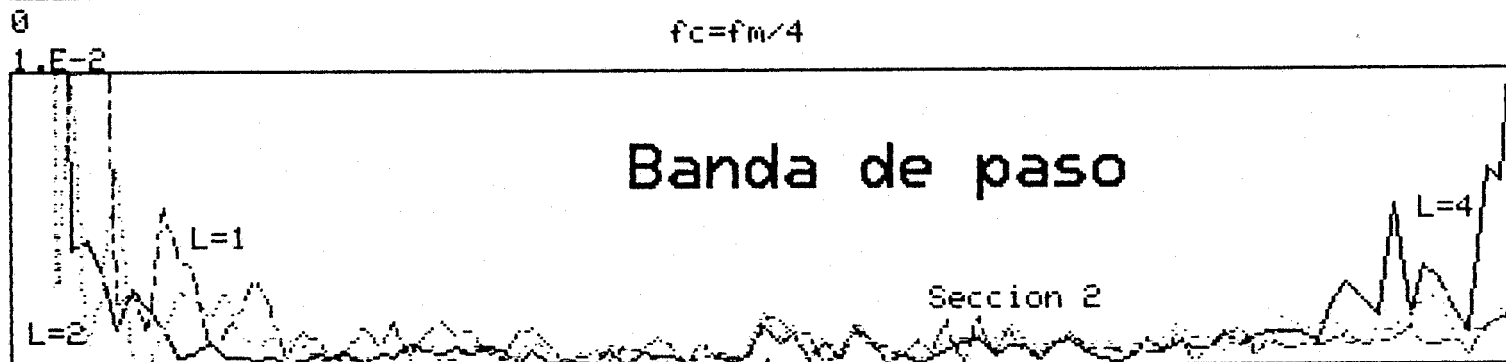
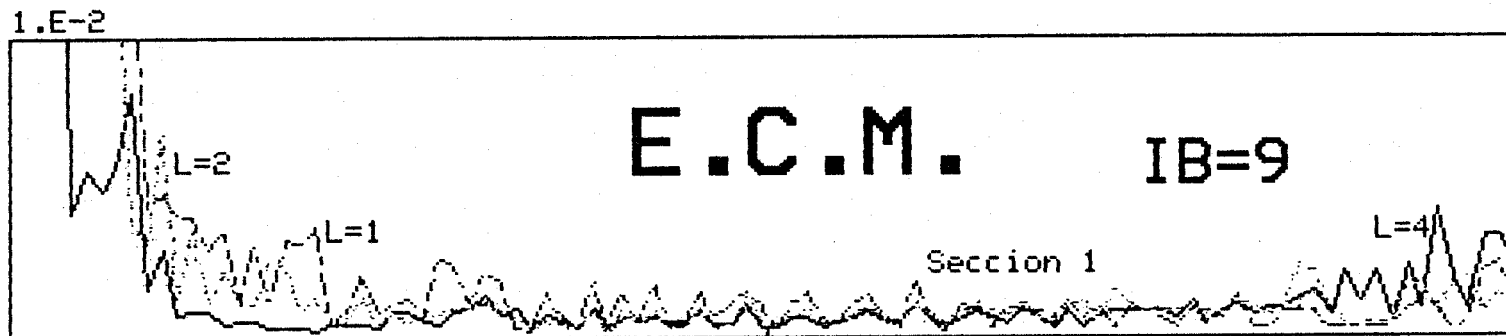


Figura II.10



$f_c = f_m / 4$

Figura II.11

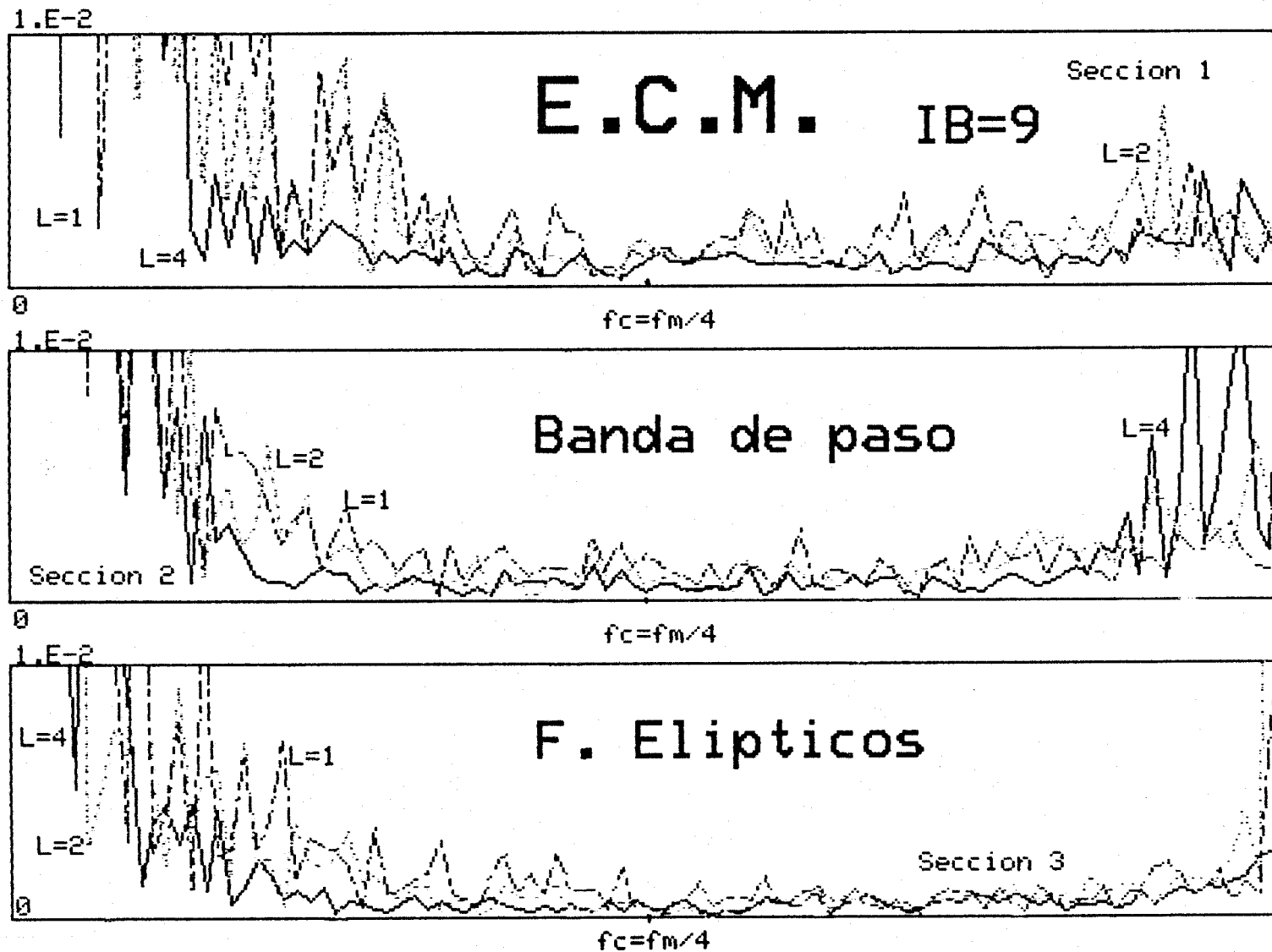


Figura II.12

2.50

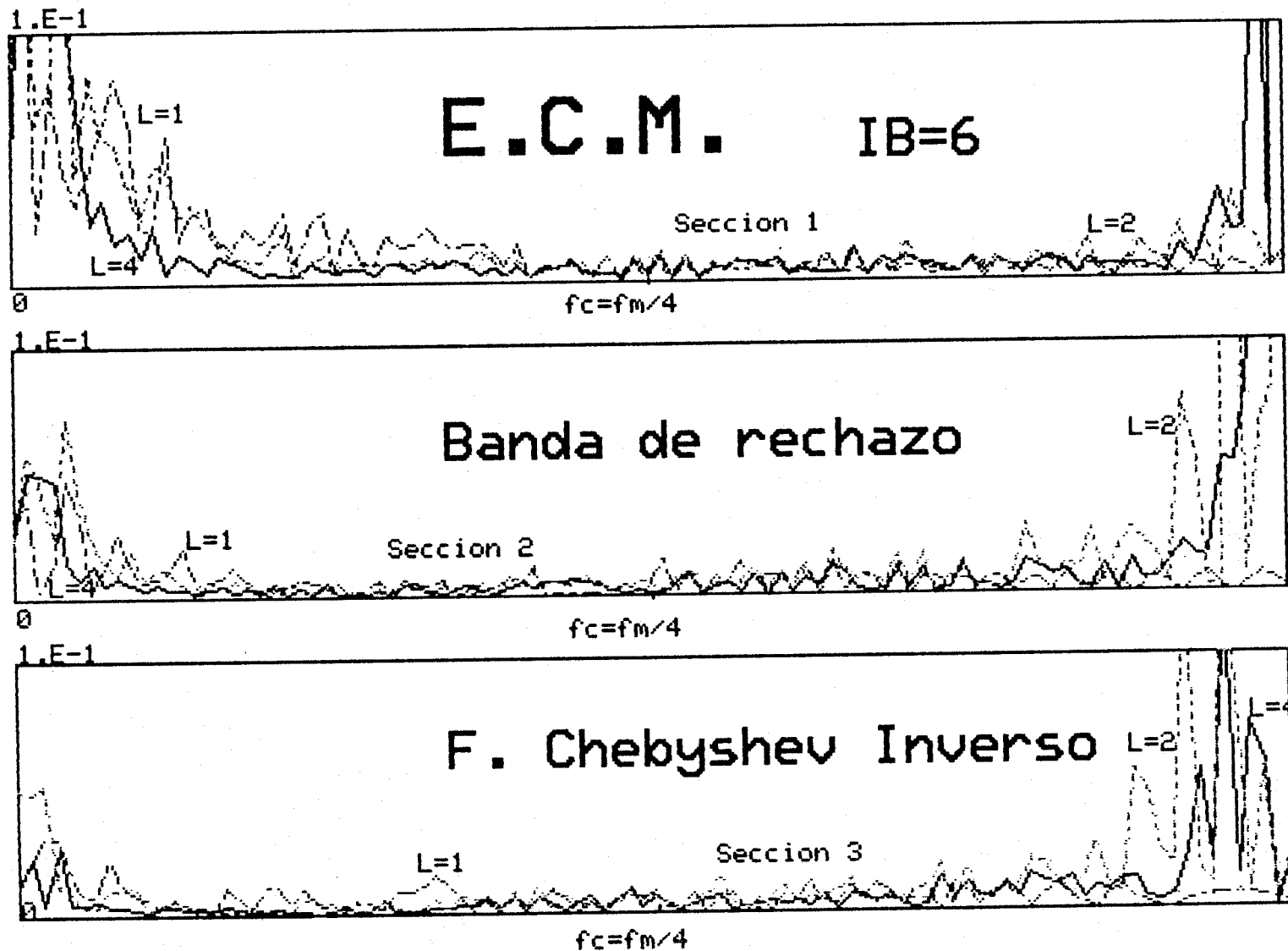


Figura II.13

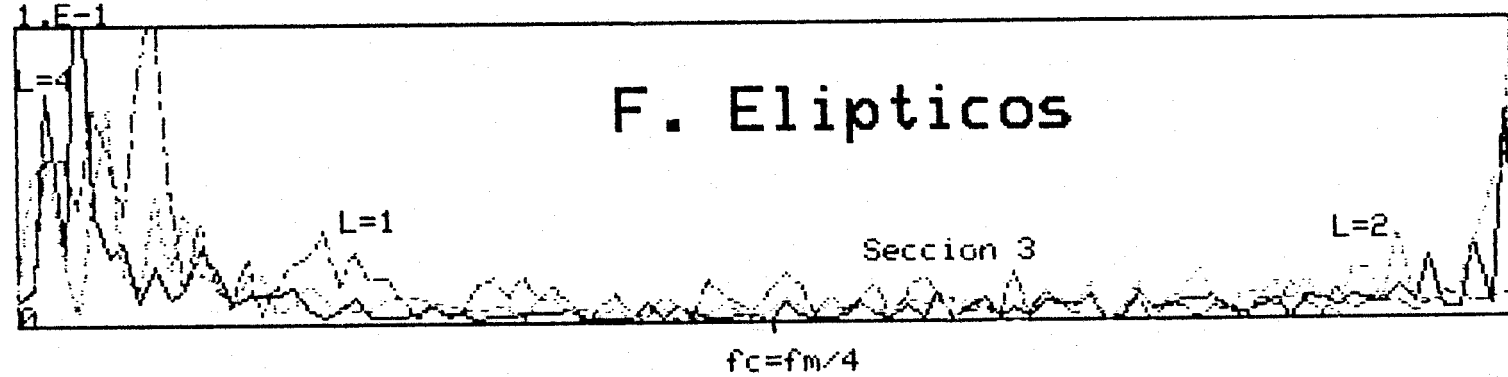
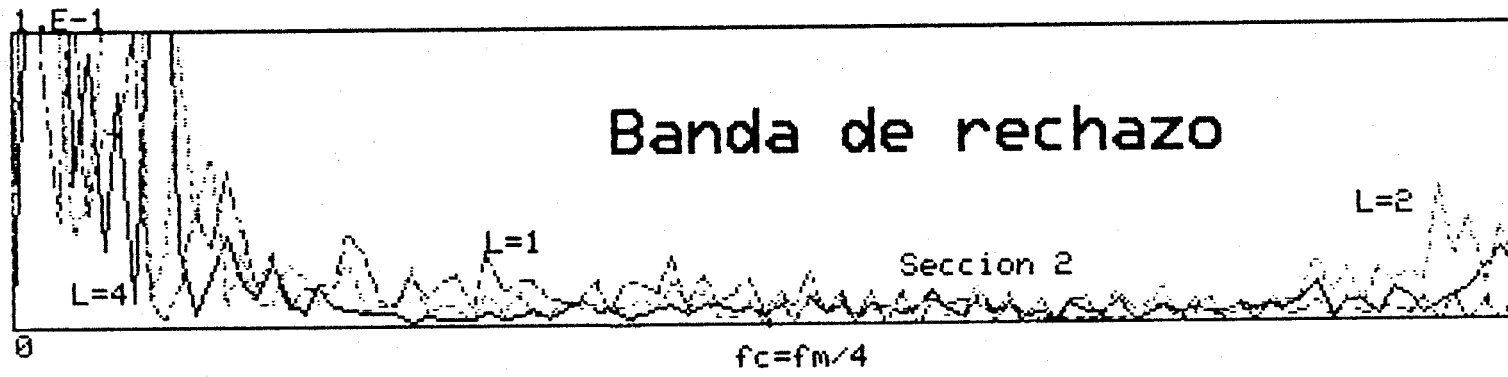
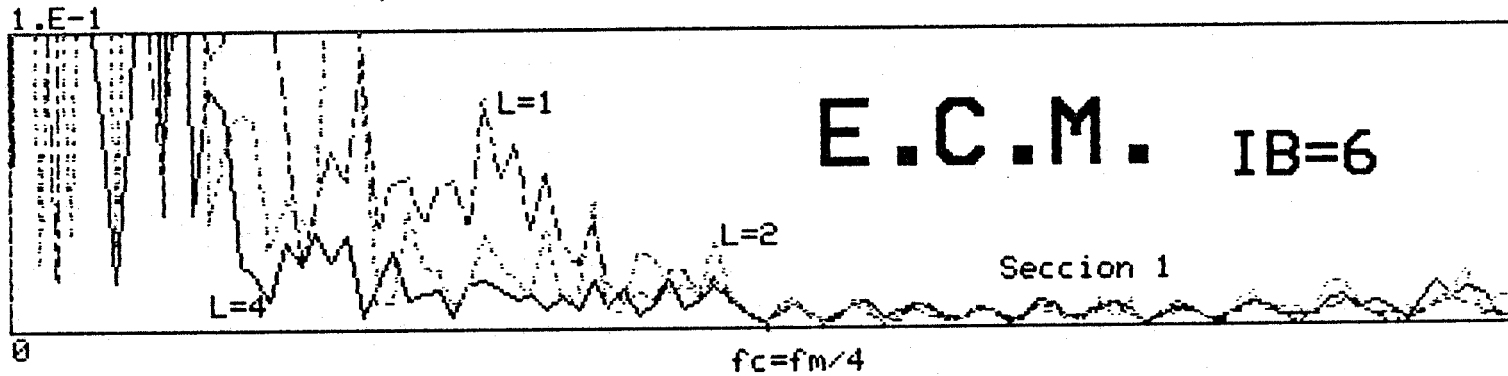
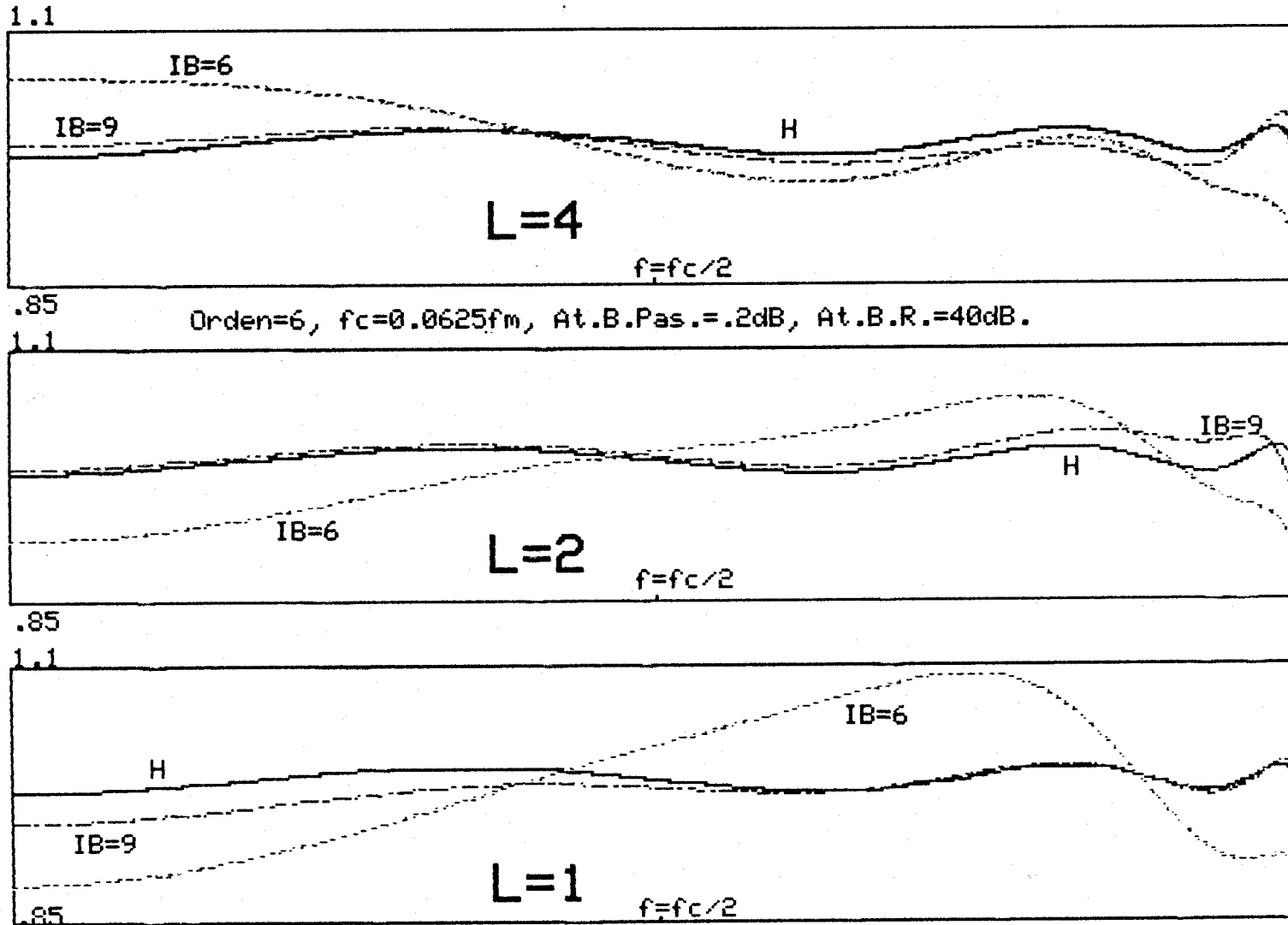


Figura II.14

Banda Pasante filtro paso bajo Eliptico.



2.52

Figura II.15

Respuesta en frecuencia de un filtro paso bajo Eliptico.

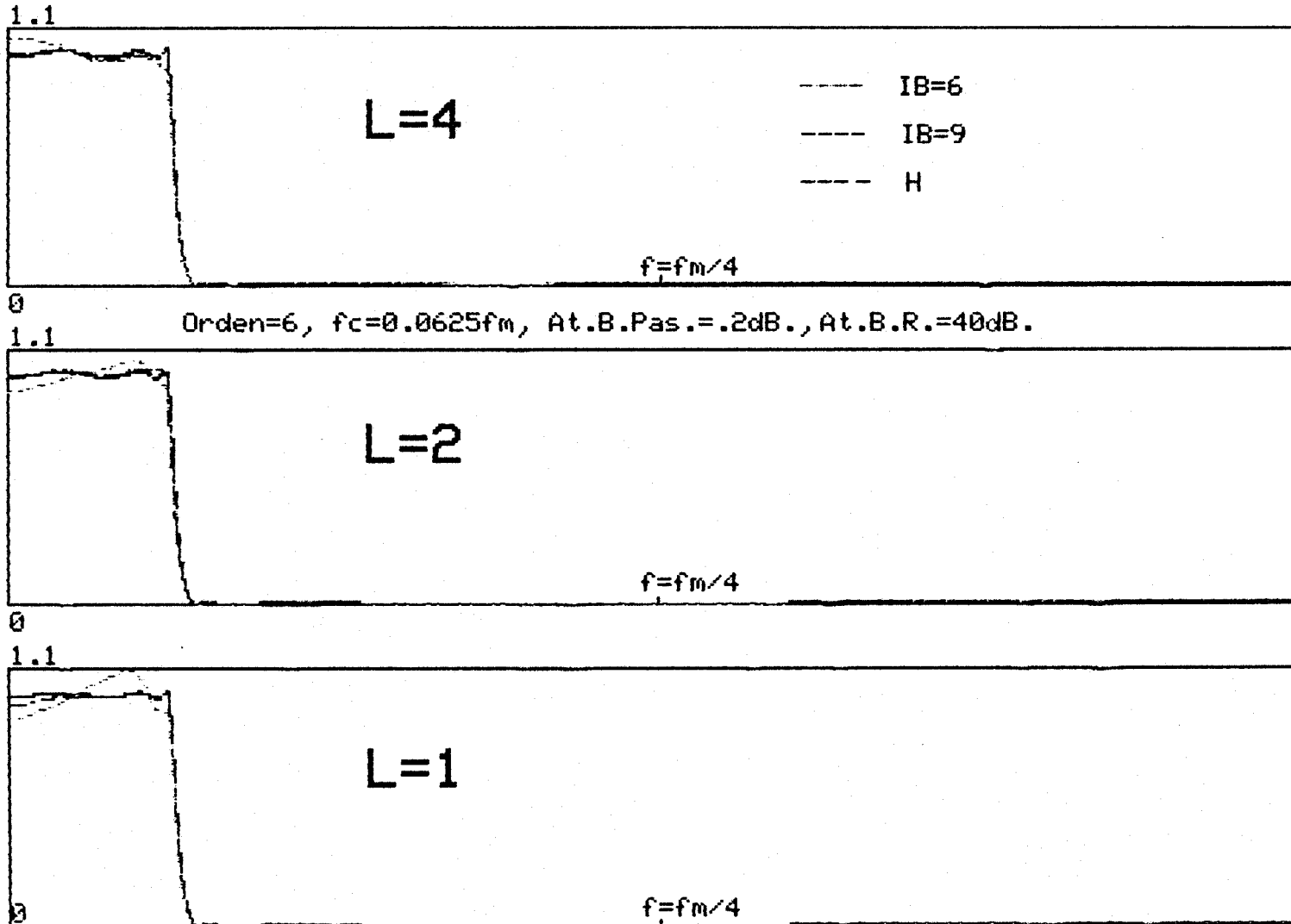
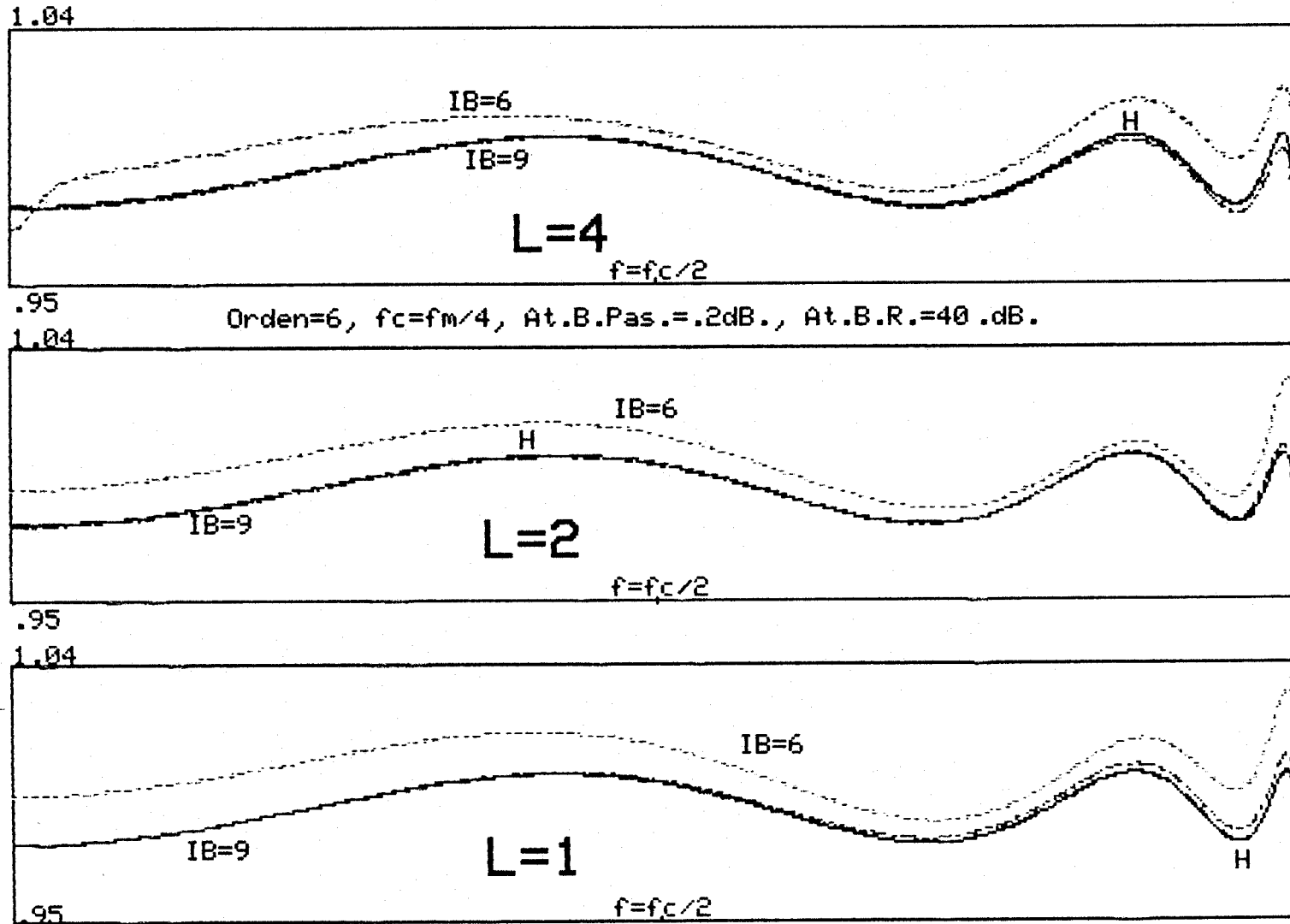


Figura II.16

Banda Pasante filtro paso bajo Eliptico



2.54

Figura II.17

Respuesta en frecuencia de un filtro paso bajo Eliptico.

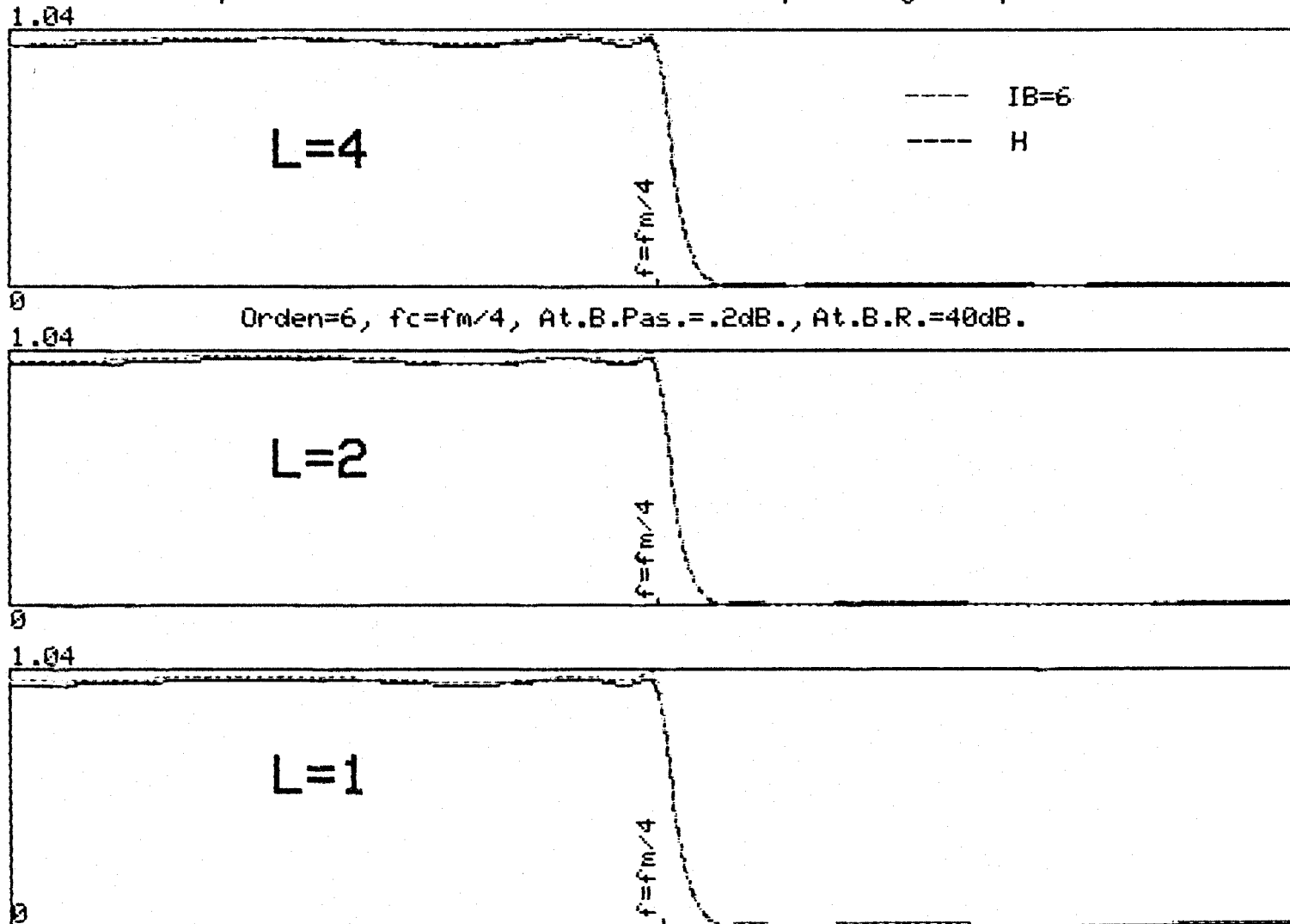


Figura II.18

```

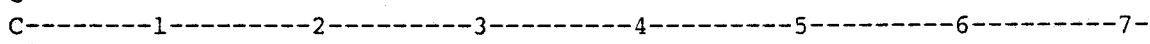
C *****
C *
C *   PROGRAMA : NUV
C *
C *   Este programa consta de cuatro partes:
C *
C *   1. En la primera disena filtros TAN paso-bajo segun las 4
C *   aproximaciones clasicas.
C *   2. En la segunda calcula los coeficientes de la estructura
C *   de bloques de orden L para la realizacion de cada seccion
C *   de segundo orden.
C *   3. La tercera parte calcula la sensibilidad de la funcion
C *   de transferencia de cada seccion de segundo orden respecto
C *   a la cuantizacion de los coeficientes de la estructura de
C *   bloques.
C *   4. La cuarta parte calcula el error cuadratico medio de la
C *   magnitud de la respuesta en frecuencia de cada una de las
C *   secciones de segundo orden cuando los coeficientes se
C *   redondean a IB bits.
C *
C *   AUTOR :      A. Ayerbe Garcia
C *
C *               C.T.N.E.
C *               DPTO. DE INVESTIGACION Y DESARROLLO
C *
C *   SECCION :   Investigacion Aplicada
C *
C *   Fecha de ultima revision : 24-Nov-1985
C *
C *****

```

PARTE 1.



PROGRAM NUV



```

COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)
COMMON/COEF/ L,IB
COMMON/SEN/ SEN1P,SEN1R,SEN2P,SEN2R,SEN3P,SEN3R
COMMON/SIM/ SIM1P,SIM1R,SIM2P,SIM2R,SIM3P,SIM3R
CHARACTER*10, NOMBRE
DIMENSION S1P(1000),S2P(1000),S3P(1000)
DIMENSION S1R(1000),S2R(1000),S3R(1000)
DIMENSION T1P(1000),T2P(1000),T3P(1000)
DIMENSION T1R(1000),T2R(1000),T3R(1000)

```

PI=4.*ATAN(1.0)

```

C N es el orden del filtro. Si N=1, se calcula, si N>1 es un dato
C F1 es el extremo de la banda de paso (Hz).
C F2 es el inicio de la banda eliminada (Hz).
C A1 es la atenuacion maxima en la banda de paso(dB)
C A2 es la atenuacion minima en la banda eliminada.

```

```

TYPE*, ' FILTRO DE ORDEN 6 '
N=6
TYPE*, ' Introduzca el orden de bloques:'
READ(5,*) L

```

```

TYPE*, ' Introduzca el no. de bits para el redondeo:'
READ(5,*) IB
30 TYPE*, ' Introduzca la frecuencia de muestreo(Hz):'
READ(5,*,ERR=40) IFM
IF(IFM.LT.1) THEN
40 CALL LECT3(*30)
END IF
TYPE*, ' Introduzca el rizado en la banda de paso,(dB)'
READ(5,*) A1
TYPE*, ' Introduzca la atenuacion minima en la banda el
* iminada'
READ(5,*) A2
90 TYPE 100
100 FORMAT(
* ' Introduzca el tipo de aproximacion: '/'
* '          1.Butterworth          '/'
* '          2.Chebyshev            '/'
* '          3.Chebyshev Inverso    '/'
* '          4.Eliptico             '/')
READ(5,*,ERR=110) IPX
IF(IPX.LT.1.OR.IPX.GT.4) THEN
110 CALL LECT3(*90)
END IF
C Se disenar 99 filtros paso bajo
DO IFEC=1,99
F1=(IFEC*.5/100.)*IFM
IF(IPX.EQ.1) CALL BUTTERWORTH
IF(IPX.EQ.2) CALL CHEBYSHEV
IF(IPX.EQ.3) CALL CHEBYSHEVI
IF(IPX.EQ.4) CALL ELIPTICO
CALL CAMBIO
S1P(IFEC)=SEN1P
S1R(IFEC)=SEN1R
S2P(IFEC)=SEN2P
S2R(IFEC)=SEN2R
S3P(IFEC)=SEN3P
S3R(IFEC)=SEN3R
T1P(IFEC)=SIM1P
T1R(IFEC)=SIM1R
T2P(IFEC)=SIM2P
T2R(IFEC)=SIM2R
T3P(IFEC)=SIM3P
T3R(IFEC)=SIM3R
END DO
C
C Escritura de ficheros
c
NOMBRE='D1P1'
OPEN(UNIT=3,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='D2P1'
OPEN(UNIT=7,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='D3P1'
OPEN(UNIT=8,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
WRITE(3) 99
WRITE(7) 99
WRITE(8) 99
WRITE(3) (S1P(I),I=1,99)
WRITE(7) (S2P(I),I=1,99)
WRITE(8) (S3P(I),I=1,99)
CLOSE(3)
CLOSE(7)
CLOSE(8)
NOMBRE='D1R1'
OPEN(UNIT=3,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='D2R1'
OPEN(UNIT=7,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')

```

```

NOMBRE='D3R1'
OPEN(UNIT=8,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
WRITE(3) 99
WRITE(7) 99
WRITE(8) 99
WRITE(3) (S1R(I),I=1,99)
WRITE(7) (S2R(I),I=1,99)
WRITE(8) (S3R(I),I=1,99)
CLOSE(3)
CLOSE(7)
CLOSE(8)
NOMBRE='F1P11'
OPEN(UNIT=3,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='F2P11'
OPEN(UNIT=7,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='F3P11'
OPEN(UNIT=8,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
WRITE(3) 99
WRITE(7) 99
WRITE(8) 99
WRITE(3) (T1P(I),I=1,99)
WRITE(7) (T2P(I),I=1,99)
WRITE(8) (T3P(I),I=1,99)
CLOSE(3)
CLOSE(7)
CLOSE(8)
NOMBRE='F1R11'
OPEN(UNIT=3,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='F2R11'
OPEN(UNIT=7,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
NOMBRE='F3R11'
OPEN(UNIT=8,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
WRITE(3) 99
WRITE(7) 99
WRITE(8) 99
WRITE(3) (T1R(I),I=1,99)
WRITE(7) (T2R(I),I=1,99)
WRITE(8) (T3R(I),I=1,99)
CLOSE(3)
CLOSE(7)
CLOSE(8)
STOP
END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      INTEGER FUNCTION ICEIL(X)
C-----1-----2-----3-----4-----5-----6-----7-
C

```

```

      IF(X.EQ.0.) THEN
          ICEIL=X
          GO TO 1000
      END IF
      IF(ABS(X-INT(X)).LT.1.E-12) THEN
          ICEIL=INT(X)
          GO TO 1000
      END IF
      IF(ABS(X-INT(X)).GT.(1.-1.E-12)) THEN
          A=ABS(X)+1.
          ICEIL=INT(A)
          IF(ICEIL.LT.0) THEN
              ICEIL=-ICEIL
          END IF
          GO TO 1000
      END IF
      IF(X.GT.0.)THEN
          ICEIL=INT(X+1)

```



```

GO TO 1000
END IF
IF(X.LT.0.) THEN
    ICEIL=-INT(ABS(X))
END IF
1000 RETURN
END

```

```

C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE LECT3(*)

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
TYPE*, ' Error en la entrada, introduzca dato de nuevo'
RETURN 1
END

```

```

C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE BUTTERWORTH

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

```

```

C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**((A1/10.)-1.))
AK1=E1/SQRT(10.**((A2/10.)-1.))
IF(N.GT.1) THEN
    F2=ATAN(T1/AK1**((1./N)))/(PI/IFM)
ELSE
    AK=T1/TAN(F2*PI/IFM)
    N=ICEIL(LOG10(AK1)/LOG10(AK))
END IF
AK=T1/TAN(F2*PI/IFM)
AL0=1./E1**((1./N))
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
    P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
    C(1)=(1.-P(1,1))/2.
END IF

```

```

C
DO I=2,INT(N/2.+1.)
    J=I-1
    Q=2.*AL0*SIN(PI*(2*J-1)/(2.*N))
    P(1,I)=2.*(1.-(AL0*T1)**2)/(1.+Q*T1+(AL0*T1)**2)
    P(2,I)=(1.-Q*T1+(AL0*T1)**2)/(1.+Q*T1+(AL0*T1)**2)
    C(I)=(1.-P(1,I)+P(2,I))/4.
END DO
RETURN
END

```

```

C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE CHEBYSHEV

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

```

```

C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**((A1/10.)-1.))
AK1=E1/SQRT(10.**((A2/10.)-1.))
IF(N.GT.1) THEN
    F2=ATAN(T1*COSH((1./N)*FNA2(1/AK1)))/(PI/IFM)

```

```

ELSE
    AK=T1/TAN(F2*PI/IFM)
    N=ICEIL(FNA2(1/AK1)/FNA2(1/AK))
END IF
AK=T1/TAN(F2*PI/IFM)
AF2=1./N*FNA1(1/E1)
AL0=SINH(AF2)
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
    P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
    C(1)=(1.-P(1,1))/2.
    ELSE
    C(1)=1./SQRT(1.+E1**2)
END IF

```

```

DO I=2,INT(N/2.+1.)
J=I-1
Q=2.*AL0*SIN(PI*(2*J-1)/(2.*N))
B=AL0**2+COS(PI*(2*J-1)/(2.*N))**2
P(1,I)=2.*(1.-B*T1**2)/(1.+Q*T1+B*T1**2)
P(2,I)=(1.-Q*T1+B*T1**2)/(1.+Q*T1+B*T1**2)
C(I)=(1.-P(1,I)+P(2,I))/4.
END DO
RETURN
END

```

C-----1-----2-----3-----4-----5-----6-----7-----

 SUBROUTINE CHEBYSHEVI

C-----1-----2-----3-----4-----5-----6-----7-----

```

COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

```

```

T1=TAN(F1*PI/IFM)
E1=SQRT(10.**((A1/10.)-1.))
AK1=E1/SQRT(10.**((A2/10.)-1.))
IF(N.GT.1) THEN
F2=ATAN(T1*COSH((1./N)*FNA2(1/AK1)))/(PI/IFM)
ELSE
    AK=T1/TAN(F2*PI/IFM)
    N=ICEIL(FNA2(1/AK1)/FNA2(1/AK))
END IF

```

```

AK=T1/TAN(F2*PI/IFM)
AF2=1./N*(FNA1(E1*(COSH(N*FNA2(1/AK)))))
AL0=1./(AK*SINH(AF2))
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
    P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
    C(1)=(1.-P(1,1))/2.
END IF

```

```

DO I=2,INT(N/2.+1.)
J=I-1
Q=2*AL0*SIN(PI*(2*J-1)/(2*N))/(1+(AL0*AK*COS(PI*
* (2*J-1)/(2*N)))**2)
B=AL0**2/(1+(AL0*AK*COS(PI*(2*J-1)/(2*N)))**2)
C1=(AK*COS(PI*(2*J-1)/(2*N)))**2
P(1,I)=2.*(1.-B*T1**2)/(1.+Q*T1+B*T1**2)
P(2,I)=(1.-Q*T1+B*T1**2)/(1.+Q*T1+B*T1**2)
P(3,I)=2*(C1-T1**2)/(C1+T1**2)
C(I)=(1.-P(1,I)+P(2,I))/(2.-P(3,I))
END DO
RETURN
END

```

```

C
C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE ELIPTICO
C-----1-----2-----3-----4-----5-----6-----7-
C
COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)
C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**(A1/10.)-1.)
AK1=E1/SQRT(10.**(A2/10.)-1.)
IF(N.GT.1) THEN
      Q=PI**2*N/(2*LOG(4.*10.**(A2/20.)/E1))
      F2=ATAN(T1*(8.*EXP(-Q)+1.))/(PI/IFM)
ELSE
      AK=T1/TAN(F2*PI/IFM)
AN=2*LOG(4.*10.**(A2/20.)/E1)*LOG(8./(1/AK-1.))/PI**2
N=ICEIL(AN)
END IF
AK=T1/TAN(F2*PI/IFM)
AF2=FNK(AK)/N/FNK(AK1)*LOG((1.+SQRT(1.+E1**2))/E1)
A0=AK
Q=FNS(AF2,A0)
A0=SQRT(1.-AK**2)
B1=FNC(AF2,A0)
AL0=Q/B1
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
      P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
      C(1)=(1.-P(1,1))/2.
ELSE
      C(1)=1./SQRT(1.+E1**2)
END IF
C
DO I1=2,INT(N/2.+1.)
L1=I1-1
AJ=(N-2.*L1+1.)/N*FNK(AK)
Q=2*AL0*FNC(AJ,AK)*FND(AJ,AK)/(1.+(AL0*AK*FNS(AJ,AK))**2)
B1=(AL0**2+FNS(AJ,AK)**2)/(1+(AL0*AK*FNS(AJ,AK))**2)
C1=(AK*FNS(AJ,AK))**2
P(1,I1)=2.*(1.-B1*T1**2)/(1.+Q*T1+B1*T1**2)
P(2,I1)=(1.-Q*T1+B1*T1**2)/(1.+Q*T1+B1*T1**2)
P(3,I1)=2.*(C1-T1**2)/(C1+T1**2)
C(I1)=(1.-P(1,I1)+P(2,I1))/(2.-P(3,I1))
END DO
RETURN
END
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNA1(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
      FNA1=LOG(X+SQRT(X**2+1.))
      RETURN
      END
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNA2(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
      FNA2=LOG(X+SQRT(X**2-1.))
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNK(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
      DIMENSION AEL(30),B(30),EE(30)
      PI=4.*ATAN(1.0)
      P2=.00001
      AEL(1)=1.
      B(1)=SQRT(1.-X**2)
      EE(1)=X
      DO I0=1,19
          AEL(I0+1)=.5*(AEL(I0)+B(I0))
          B(I0+1)=SQRT(AEL(I0)*B(I0))
          EE(I0+1)=.5*(AEL(I0)-B(I0))
      IF((EE(I0+1)-P2).LE.0.) GO TO 2000
      END DO
2000  J0=I0+1
      FNK=.5*PI/AEL(J0)
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      AX=X/2./FNK(A0)
      Y=EXP(-(PI*FNK(SQRT(1.-A0**2))/FNK(A0)))
      FNS=FNS4(AX,Y)/FNS0(AX,Y)/SQRT(A0)
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNC(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      AX=X/2./FNK(A0)
      Y=EXP(-(PI*FNK(SQRT(1.-A0**2))/FNK(A0)))
      FNC=SQRT(SQRT(1.-A0**2)/A0)*FNS2(AX,Y)/FNS0(AX,Y)
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FND(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      AX=X/2./FNK(A0)
      Y=EXP(-(PI*FNK(SQRT(1.-A0**2))/FNK(A0)))
      FND=SQRT(SQRT(1.-A0**2))*FNS3(AX,Y)/FNS0(AX,Y)
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS0(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S0=1.
      DO I=1,10
          S0=S0+2.*(-1)**I*Y**(I**2)*COS(2.*I*PI*X)
      END DO
      FNS0=S0
      RETURN

```

END

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS4(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S4=0.
      DO I=1,11
      J=I-1
      S4=S4+2.*(-1)**J*Y**(J*(J+1))*Y**(1./4.)*SIN((2*J+1)*PI*X)
      END DO
      FNS4=S4
      RETURN
      END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS2(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S2=0.
      DO I=1,11
      J=I-1
      S2=S2+2.*Y**(1./4.)*Y**(J*(J+1))*COS((2*J+1)*PI*X)
      END DO
      FNS2=S2
      RETURN
      END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS3(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S3=1.
      DO I=1,10
      S3=S3+2.*Y**(I**2.)*COS(2*I*PI*X)
      END DO
      FNS3=S3
      RETURN
      END
```

PARTE 2.

```
C
C
C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE CAMBIO
C-----1-----2-----3-----4-----5-----6-----7-
      COMMON/FILT/ P(3,20),C(20)
      COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
      COMMON/COEF/ L,IB
      COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
      COMMON/MATBR/ EAR(2,2),EBR(2,10),ECR(10,2),EDR(10,10)
      COMMON/MATES/ SA(2,2),SB(2,1),SC(1,2),SD
      COMMON/SEN/ SEN1P,SEN1R,SEN2P,SEN2R,SEN3P,SEN3R
      COMMON/SIM/ SIM1P,SIM1R,SIM2P,SIM2R,SIM3P,SIM3R
      COMMON/MED/ AMEDP,AMEDR,IBLO
      IF((IPX.EQ.1).OR.(IPX.EQ.2)) GO TO 16
      IF((IPX.EQ.3).OR.(IPX.EQ.4)) GO TO 17
      DO I=1,N/2
      SA(1,1)=P(1,I+1)
      SA(1,2)=-P(2,I+1)
      SA(2,1)=1.
      SA(2,2)=0.
      SB(1,1)=C(I+1)
      SB(2,1)=0.
```

```

SC(1,1)=2.+P(1,I+1)
SC(1,2)=1.-P(2,I+1)
SD=C(I+1)
IBLO=I
CALL MATRICES
CALL TRANS
IF(IBLO.EQ.1) THEN
    SEN1P=AMEDP
    SEN1R=AMEDR
ELSE IF(IBLO.EQ.2) THEN
    SEN2P=AMEDP
    SEN2R=AMEDR
ELSE IF(IBLO.EQ.3) THEN
    SEN3P=AMEDP
    SEN3R=AMEDR
END IF
CALL FUNH
IF(IBLO.EQ.1) THEN
    SIM1P=AMEDP
    SIM1R=AMEDR
ELSE IF(IBLO.EQ.2) THEN
    SIM2P=AMEDP
    SIM2R=AMEDR
ELSE IF(IBLO.EQ.3) THEN
    SIM3P=AMEDP
    SIM3R=AMEDR
END IF
END DO
GO TO 20
DO I=1,N/2
SA(1,1)=P(1,I+1)
SA(1,2)=-P(2,I+1)
SA(2,1)=1.
SA(2,2)=0.
SB(1,1)=C(I+1)
SB(2,1)=0.
SC(1,1)=-P(3,I+1)+P(1,I+1)
SC(1,2)=1-P(2,I+1)
SD=C(I+1)
IBLO=I
CALL MATRICES
CALL TRANS
IF(IBLO.EQ.1) THEN
    SEN1P=AMEDP
    SEN1R=AMEDR
ELSE IF(IBLO.EQ.2) THEN
    SEN2P=AMEDP
    SEN2R=AMEDR
ELSE IF(IBLO.EQ.3) THEN
    SEN3P=AMEDP
    SEN3R=AMEDR
END IF
CALL FUNH
IF(IBLO.EQ.1) THEN
    SIM1P=AMEDP
    SIM1R=AMEDR
ELSE IF(IBLO.EQ.2) THEN
    SIM2P=AMEDP
    SIM2R=AMEDR
ELSE IF(IBLO.EQ.3) THEN
    SIM3P=AMEDP
    SIM3R=AMEDR
END IF
END DO
RETURN
END

```

17

20

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE POTENCIA(SA,R,K)
C-----1-----2-----3-----4-----5-----6-----7-
C
      DIMENSION SA(2,2),R(2,2),R1(2,2)
      R(1,1)=1.
      R(1,2)=0.
      R(2,1)=0.
      R(2,2)=1.
      IF(K.EQ.0) GOTO 33
      DO IN=1,K
          DO I=1,2
              DO J=1,2
                  PAR=0.
                  DO M=1,2
                      PAR=PAR+R(I,M)*SA(M,J)
                  END DO
                  R1(I,J)=PAR
              END DO
          END DO
          DO I=1,2
              DO J=1,2
                  R(I,J)=R1(I,J)
              END DO
          END DO
      END DO
33  RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE PROD1(P1,SB,D1)
C-----1-----2-----3-----4-----5-----6-----7-
C
      DIMENSION P1(2,2),SB(2,1),D1(2,1)
      DO I=1,2
          D1(I,1)=0.
          DO J=1,2
              D1(I,1)=D1(I,1)+P1(I,J)*SB(J,1)
          END DO
      END DO
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE PROD2(P1,SC,D2)
C-----1-----2-----3-----4-----5-----6-----7-
C
      DIMENSION P1(2,2),SC(1,2),D2(1,2)
      DO I=1,2
          D2(1,I)=0.
          DO J=1,2
              D2(1,I)=D2(1,I)+SC(1,J)*P1(J,I)
          END DO
      END DO
      RETURN
      END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE MATRICES
C-----1-----2-----3-----4-----5-----6-----7-
C
      COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
      COMMON/COEF/ L,IB
      COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
      COMMON/MATBR/ EAR(2,2),EBR(2,10),ECR(10,2),EDR(10,10)

```

```
COMMON/MATES/ SA(2,2),SB(2,1),SC(1,2),SD  
DIMENSION P1(2,2),D1(2,1),D2(1,2),R(2,2)
```

```
C  
C  
C  
Calculo de EA(I,J) y de EAR(I,J)
```

```
K=L  
CALL POTENCIA(SA,R,K)  
DO I=1,2  
DO J=1,2  
    EA(I,J)=R(I,J)  
    EAR(I,J)=RED(EA(I,J),IB)  
END DO  
END DO
```

```
C  
C  
C  
Calculo de EB(I,J) y EBR(I,J)
```

```
DO M=1,L  
K=L-M  
CALL POTENCIA(SA,R,K)  
DO I=1,2  
DO J=1,2  
    P1(I,J)=R(I,J)  
END DO  
END DO  
CALL PROD1(P1,SB,D1)  
DO IN=1,2  
    EB(IN,M)=D1(IN,1)  
    EBR(IN,M) = RED(EB(IN,M),IB)  
END DO  
END DO
```

```
C  
C  
C  
Calculo de EC(I,J) y ECR(I,J)
```

```
DO M=1,L  
K=M-1  
CALL POTENCIA(SA,R,K)  
DO I=1,2  
DO J=1,2  
    P1(I,J)=R(I,J)  
END DO  
END DO  
CALL PROD2(P1,SC,D2)  
DO IN=1,2  
    EC(M,IN)=D2(1,IN)  
    ECR(M,IN) = RED(EC(M,IN),IB)  
END DO  
END DO
```

```
C  
C  
C  
Calculo de ED(I,J) y EDR(I,J)
```

```
DO I=1,L  
DO J=1,L  
IF(I.LT.J) THEN  
    ED(I,J)=0.  
    EDR(I,J)= RED(ED(I,J),IB)  
END IF  
IF(I.EQ.J) THEN  
    ED(I,J)=SD  
    EDR(I,J)=RED(ED(I,J),IB)  
END IF  
IF(I.GT.J) THEN  
K=I-J-1  
CALL POTENCIA(SA,R,K)  
DO IL=1,2  
DO IM=1,2  
    P1(IL,IM)=R(IL,IM)
```



```

END DO
END DO
CALL PROD1(P1,SB,D1)
ED(I,J)=SC(1,1)*D1(1,1)+SC(1,2)*D1(2,1)
EDR(I,J)=RED(ED(I,J),IB)
END IF
END DO
END DO
RETURN
END

```

C
C
C
C
C

PARTE 3.

C-----1-----2-----3-----4-----5-----6-----7
SUBROUTINE TRANS

C
C-----1-----2-----3-----4-----5-----6-----7-
C

```

COMMON/FILT/ P(3,20),C(20)
COMMON/CALEFIL/ N,F1,F2,IFM,IPX,PI
COMMON/COEF/ L
COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
COMMON/MED/ AMEDP,AMEDR,IBLO
DIMENSION A0(10,10),A1(10,10),A2(10,10),COF(144)
DIMENSION AL1(10,10,144),G1(144)
DIMENSION AL2(10,10,144),G2(144)
DIMENSION AL3(10,10,144),G3(144)
DIMENSION SEN(144),AMAG(1024)
DIMENSION AH(10,10),DH(10,10,144)
COMPLEX Z,AH,DH,DE1,DE2,SUM1
PI=4.*ATAN(1.0)

```

C
C Calculo de H(z)
C

```

DO I=1,L
DO J=1,L
A0(I,J)=ED(I,J)
A1(I,J)=EC(I,1)*EB(1,J)+EB(2,J)*EC(I,2)-ED(I,J)*(EA(1,1)+
* EA(2,2))
A2(I,J)=(EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2))*ED(I,J)+EA(2,1)*EB(1,J)
* *EC(I,2)+EA(1,2)*EB(2,J)*EC(I,1)-EA(2,2)*EB(1,J)*EC(I,1)-
* EA(1,1)*EB(2,J)*EC(I,2)
END DO
END DO
B0=1.
B1=-EA(1,1)-EA(2,2)
B2=EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2)

```

C
C Reordenacion de los coeficientes en COF(I)
C

```

COF(1)=EA(1,1)
COF(2)=EA(1,2)
COF(3)=EA(2,1)
COF(4)=EA(2,2)
DO I=1,L
COF(4+I)=EB(1,I)
END DO
DO I=1,L
COF(4+L+I)=EB(2,I)
END DO
DO I=1,L
COF(4+2*L+I)=EC(I,1)
END DO
DO I=1,L

```

```

COF(4+3*L+I)=EC(I,2)
END DO
DO I=1,L
DO M=1,L
COF(4+4*L+L*(I-1)+M)=ED(I,M)
END DO
END DO

```

C

C Calculo de [Lij]

c AL1(I,J,K) es la derivada del coef. A0 del elemento I,J de H(z) respecto
c al coeficiente K

C

```

DO I=1,L
DO J=1,L
DO K=1,4+4*L
    AL1(I,J,K)=0
END DO
DO K=1,L
DO M=1,L
IF(K.NE.I .OR. M.NE.J) THEN
    AL1(I,J,4+4*L+L*(K-1)+M)=0.
ELSE
    AL1(I,J,4+4*L+L*(K-1)+M)=1.
END IF
END DO
END DO
END DO
END DO

```

C

C AL2(I,J,K) es la derivada del coef. A1 del elemento I,J de H(z) con
C respecto al cof. K

C

```

DO I=1,L
DO J=1,L
AL2(I,J,1)=-ED(I,J)
AL2(I,J,2)=0
AL2(I,J,3)=0
AL2(I,J,4)=-ED(I,J)
DO K=1,L
IF(K.NE.J) THEN
    AL2(I,J,K+4)=0
ELSE
    AL2(I,J,K+4)=EC(I,1)
END IF
END DO
DO K=1,L
IF(K.NE.J) THEN
    AL2(I,J,K+L+4)=0
ELSE
    AL2(I,J,K+L+4)=EC(I,2)
END IF
END DO
DO K=1,L
IF(K.NE.I) THEN
    AL2(I,J,4+2*L+K)=0
ELSE
    AL2(I,J,4+2*L+K)=EB(1,J)
END IF
END DO
DO K=1,L
IF(K.NE.I) THEN
    AL2(I,J,4+3*L+K)=0
ELSE
    AL2(I,J,4+3*L+K)=EB(2,J)
END IF
END DO

```

```

DO K=1,L
DO M=1,L
IF(K.NE.I .OR. M.NE.J) THEN
    AL2(I,J,4+4*L+L*(K-1)+M)=0.
ELSE
    AL2(I,J,4+4*L+L*(K-1)+M)=- (EA(1,1)+EA(2,2))
END IF
END DO
END DO
END DO
END DO

```

C
C
C
C

AL3(I,J,K) es la derivada del coef. A2 del elto. I,J de H(z) respecto al coef. K.

```

DO I=1,L
DO J=1,L
AL3(I,J,1)=EA(2,2)*ED(I,J)-EB(2,J)*EC(I,2)
AL3(I,J,2)=EB(2,J)*EC(I,1)-EA(2,1)*ED(I,J)
AL3(I,J,3)=EB(1,J)*EC(I,2)-EA(1,2)*ED(I,J)
AL3(I,J,4)=EA(1,1)*ED(I,J)-EB(1,J)*EC(I,1)
DO K=1,L
IF(K.NE.J) THEN
    AL3(I,J,4+K)=0.
ELSE
    AL3(I,J,4+K)=EA(2,1)*EC(I,2)-EA(2,2)*EC(I,1)
END IF
END DO
DO K=1,L
IF(K.NE.J) THEN
    AL3(I,J,4+L+K)=0.
ELSE
    AL3(I,J,4+L+K)=EA(1,2)*EC(I,1)-EA(1,1)*EC(I,2)
END IF
END DO
DO K=1,L
IF(K.NE.I) THEN
    AL3(I,J,4+2*L+K)=0.
ELSE
    AL3(I,J,4+2*L+K)=EA(1,2)*EB(2,J)-EA(2,2)*EB(1,J)
END IF
END DO
DO K=1,L
IF(K.NE.I) THEN
    AL3(I,J,4+3*L+K)=0.
ELSE
    AL3(I,J,4+3*L+K)=EA(2,1)*EB(1,J)-EA(1,1)*EB(2,J)
END IF
END DO
DO K=1,L
DO M=1,L
IF(K.NE.I .OR.M.NE.J) THEN
    AL3(I,J,4+4*L+L*(K-1)+M)=0.
ELSE
    AL3(I,J,4+4*L+L*(K-1)+M)=EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2)
END IF
END DO
END DO
END DO

```

C
C
C
C
C
C

Calculo de [Gij], independiente de I,J.

Definicion de G1(K)

```
NCOF=4+4*L+L**2
```

```
DO K=1,NCOF
```

```
    G1(K)=0.
```

```
END DO
```

```
Definicion de G2(K)
```

```
G2(1)=-1.
```

```
G2(2)=0.
```

```
G2(3)=0.
```

```
G2(4)=-1.
```

```
DO K=1,NCOF-4
```

```
    G2(K+4)=0.
```

```
END DO
```

```
Definicion de G3(K)
```

```
G3(1)=EA(2,2)
```

```
G3(2)=-EA(2,1)
```

```
G3(3)=-EA(1,2)
```

```
G3(4)=EA(1,1)
```

```
DO K=1,NCOF-4
```

```
    G3(K+4)=0.
```

```
END DO
```

```
Comienza el calculo de la sensibilidad
```

```
DH(I,J,K) es la derivada del elemento I,J de H(z) respecto al  
coeficiente K. Es complejo y funcion de la frecuencia
```

```
F es la frecuencia normalizada FEQ/FM
```

```
Sensibilidad en la banda de paso
```

```
TB=F1/IFM
```

```
DO IF=1,49
```

```
F=(IF)*TB/50.
```

```
Z=CMPLX(COS(2*PI*L*F),SIN(2*PI*L*F))
```

```
DO I=1,L
```

```
DO J=1,L
```

```
DE1=A0(I,J)+A1(I,J)*Z**(-1)+A2(I,J)*Z**(-2)
```

```
DE2=B0+B1*Z**(-1)+B2*Z**(-2)
```

```
AH(I,J)=DE1/DE2
```

```
DO K=1,NCOF
```

```
DH(I,J,K)=AH(I,J)*((AL1(I,J,K)/DE1-G1(K)/DE2)+Z**(-1)*
```

```
* AL2(I,J,K)/DE1-G2(K)/DE2)+Z**(-2)*(AL3(I,J,K)/DE1-
```

```
* G3(K)/DE2))
```

```
END DO
```

```
END DO
```

```
END DO
```

```
Definicion del modulo de H escalar
```

```
SE TOMA LA COLUMNA IC
```

```
IC=1
```

```
SUM1=0.
```

```
Z=CMPLX(COS(2*PI*F),SIN(2*PI*F))
```

```
DO J=1,L
```

```
SUM1=SUM1+Z**(IC-J)*AH(J,IC)
```

```
END DO
```

```
SUM=SQRT(REAL(SUM1*CONJG(SUM1)))
```

```
Calculo de la sensibilidad del modulo de H(z) escalar.
```

```
DO I=1,NCOF
```

```

SEN(I)=0.
DO M=1,L
SEN(I)=SEN(I)+REAL(DH(M,IC,I)*CONJG(AH(M,IC)))
END DO
DO M=1,L
DO IP=1,L
IF(IP.NE.M) THEN
SEN(I)=SEN(I)+REAL(CONJG(AH(M,IC))*DH(IP,IC,I)*Z**
* (M-IP))
END IF
END DO
END DO
END DO

```

C
C
C
Calculo de la sensibilidad total

```

AMAG(IF)=0.
DO I=1,NCOF
AMAG(IF)=AMAG(IF)+COF(I)*SEN(I)/SUM
END DO
END DO
NP=IF-1

```

C
C
C
AMAG(I) es una funcion de la frecuencia que se quiere evaluar.

```

AMEDP=0.
DO I=1,NP
AMEDP=AMEDP+(AMAG(I))**2
END DO
AMEDP=SQRT(AMEDP/NP)

```

C
C
C
Sensibilidad en la banda eliminada

```

TB=(.5*IFM-F2)/IFM
DO IF=1,49
F=(IF)*TB/50.
Z=CMPLX(COS(2*PI*L*F),SIN(2*PI*L*F))
DO I=1,L
DO J=1,L
DE1=A0(I,J)+A1(I,J)*Z**(-1)+A2(I,J)*Z**(-2)
DE2=B0+B1*Z**(-1)+B2*Z**(-2)
AH(I,J)=DE1/DE2
DO K=1,NCOF
DH(I,J,K)=AH(I,J)*((AL1(I,J,K)/DE1-G1(K)/DE2)+Z**(-1))*
* AL2(I,J,K)/DE1-G2(K)/DE2)+Z**(-2)*(AL3(I,J,K)/DE1-
* G3(K)/DE2))
END DO
END DO
END DO

```

C
C
C
Definicion del modulo de H escalar

```

IC=1
SUM1=0.
Z=CMPLX(COS(2*PI*F),SIN(2*PI*F))
DO J=1,L
SUM1=SUM1+Z**(IC-J)*AH(J,IC)
END DO
SUM=SQRT(REAL(SUM1*CONJG(SUM1)))

```

C
C
C
Calculo de la sensibilidad del modulo de H(z) escalar.

```

DO I=1,NCOF
SEN(I)=0.
DO M=1,L
SEN(I)=SEN(I)+REAL(DH(M,IC,I)*CONJG(AH(M,IC)))

```

```

END DO
DO M=1,L
DO IP=1,L
IF(IP.NE.M) THEN
SEN(I)=SEN(I)+REAL(CONJG(AH(M,IC))*DH(IP,IC,I)*Z**
* (M-IP))
END IF
END DO
END DO
END DO

```

C
C Calculo de la sensibilidad total
C

```

AMAG(IF)=0.
DO I=1,NCOF
AMAG(IF)=AMAG(IF)+COF(I)*SEN(I)/SUM
END DO
END DO
NP=IF-1

```

C
C AMAG(I) es una funcion de la frecuencia que se quiere evaluar.
C

```

AMEDR=0.
DO I=1,NP
AMEDR=AMEDR+(AMAG(I))**2
END DO
AMEDR=SQRT(AMEDR/NP)
RETURN
END

```

C
C PARTE 4.
C
C

C-----1-----2-----3-----4-----5-----6-----7
C SUBROUTINE FUNH

C-----1-----2-----3-----4-----5-----6-----7-
C

```

COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/COEF/ L,IB
COMMON/MATBR/ EAR(2,2),EBR(2,10),ECR(10,2),EDR(10,10)
COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
COMMON/MED/ AMEDP,AMEDR,IBLO
DIMENSION A0(10,10),A1(10,10),A2(10,10)
DIMENSION A0R(10,10),A1R(10,10),A2R(10,10)
DIMENSION AH(10,10),AHR(10,10)
COMPLEX Z,AH,AHR,DE1,DE2,DE1R,DE2R,FUN,FUNR
PI =4.*ATAN(1.0)

```

C
C Calculo de H(z) y HR(z)
C

```

DO I=1,L
DO J=1,L
A0(I,J)=ED(I,J)
A1(I,J)=EC(I,1)*EB(1,J)+EB(2,J)*EC(I,2)-ED(I,J)*(EA(1,1)+
* EA(2,2))
A2(I,J)=(EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2))*ED(I,J)+EA(2,1)*EB(1,J)
* *EC(I,2)+EA(1,2)*EB(2,J)*EC(I,1)-EA(2,2)*EB(1,J)*EC(I,1)-
* EA(1,1)*EB(2,J)*EC(I,2)
END DO
END DO
B0=1.
B1=-EA(1,1)-EA(2,2)
B2=EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2)

```

DO I=1,L

```

DO J=1,L
A0R(I,J)=EDR(I,J)
A1R(I,J)=ECR(I,1)*EBR(1,J)+EBR(2,J)*ECR(I,2)-EDR(I,J)*(EAR(1,1)+
* EAR(2,2))
A2R(I,J)=(EAR(1,1)*EAR(2,2)-EAR(2,1)*EAR(1,2))*EDR(I,J)
* +EAR(2,1)*EBR(1,J)*ECR(I,2)+EAR(1,2)*EBR(2,J)*ECR(I,1)-
* EAR(2,2)*EBR(1,J)*ECR(I,1)-EAR(1,1)*EBR(2,J)*ECR(I,2)
END DO
END DO
B0R=1.
B1R=-EAR(1,1)-EAR(2,2)
B2R=EAR(1,1)*EAR(2,2)-EAR(2,1)*EAR(1,2)

```

C
C
C
C
C
C
Comienza el calculo de la sensibilidad

F es la frecuencia normalizada FEQ/FM

```

AMED=0.
TB=F1/IFM
DO IF=1,49
F=(IF)*TB/50.
Z=CMPLX(COS(2*PI*L*F),SIN(2*PI*L*F))
DO I=1,L
J=1
DE1=A0(I,J)+A1(I,J)*Z**(-1)+A2(I,J)*Z**(-2)
DE2=B0+B1*Z**(-1)+B2*Z**(-2)
AH(I,J)=DE1/DE2
DE1R=A0R(I,J)+A1R(I,J)*Z**(-1)+A2R(I,J)*Z**(-2)
DE2R=B0R+B1R*Z**(-1)+B2R*Z**(-2)
AHR(I,J)=DE1R/DE2R
END DO

```

C
C
C
C
Calculo de H(z) escalar a partir de la matriz H(z), con y sin redondeo

```

FUN=0.
FUNR=0.
Z=CMPLX(COS(2*PI*F),SIN(2*PI*F))
DO J=1,L
FUN=FUN+Z**(1.-J)*AH(J,1)
FUNR=FUNR+Z**(1.-J)*AHR(J,1)
END DO

```

C
C
C
Se define ahora el error cuadratico medio

```

AFUN=SQRT(REAL(FUN*CONJG(FUN)))
AFUNR=SQRT(REAL(FUNR*CONJG(FUNR)))
AMED=AMED+(AFUN-AFUNR)**2
END DO
NP=IF-1
AMEDP=SQRT(AMED/NP)

```

C
C
C
Sensibilidad en la banda eliminada

```

AMED=0.
TB=(.5*IFM-F2)/IFM
DO IF=1,49
F=(IF)*TB/50.
Z=CMPLX(COS(2*PI*L*F),SIN(2*PI*L*F))
DO I=1,L
J=1
DE1=A0(I,J)+A1(I,J)*Z**(-1)+A2(I,J)*Z**(-2)
DE2=B0+B1*Z**(-1)+B2*Z**(-2)
AH(I,J)=DE1/DE2
DE1R=A0R(I,J)+A1R(I,J)*Z**(-1)+A2R(I,J)*Z**(-2)

```

```

DE2R=B0R+B1R*Z**(-1)+B2R*Z**(-2)
AHR(I,J)=DE1R/DE2R
END DO

```

C
C
C
C

Calculo de H(z) escalar a partir de la matriz H(z), con y sin redondeo

```

FUN=0.
FUNR=0.
Z=CMPLX(COS(2*PI*F),SIN(2*PI*F))
DO J=1,L
FUN=FUN+Z**(1.-J)*AH(J,1)
FUNR=FUNR+Z**(1.-J)*AHR(J,1)
END DO

```

C
C
C

Se define ahora el error cuadratico medio

```

AFUN=SQRT(REAL(FUN*CONJG(FUN)))
AFUNR=SQRT(REAL(FUNR*CONJG(FUNR)))
AMED=AMED+(AFUN-AFUNR)**2
END DO
NP=IF-1
AMEDR=SQRT(AMED/NP)
RETURN
END

```

C
C
C
C
C

```

-----1-----2-----3-----4-----5-----6-----7-
REAL FUNCTION RED(X,IH)

```

```

-----1-----2-----3-----4-----5-----6-----7-
IF (X.EQ.0.) THEN
    RED=0.
ELSE
    IEX=ICEIL(ALOG10(ABS(X))/ALOG10(2.))
    XP=X*(2.**(-IEX))
    REDO=(IFLOOR((2.**IH)*XP+.5))/(2.**IH)
    RED=REDO*(2.**IEX)
END IF
RETURN
END

```

C
C
C
C

```

-----1-----2-----3-----4-----5-----6-----7-
FUNCTION IFLOOR(X)

```

```

-----1-----2-----3-----4-----5-----6-----7-
IF(X.GT.0.) THEN
    IFLOOR =INT(X)
    GO TO 10
END IF
IF(X.LT.0.) THEN
    IFLOOR=-INT(ABS(X)+.99)
END IF
RETURN
END

```



```

C *****
C *
C *   PROGRAMA   :   SIM
C *
C *           Este programa calcula la magnitud de la respuesta
C *           en frecuencia de un filtro paso de bajo determinado,
C *           realizado por conexion en cascada de secciones de segundo
C *           orden realizadas con estructuras de bloques de orden L,
C *           a) con los coeficientes en maxima precision
C *           b) con los coeficientes redondeados a IB bits.
C *
C *   AUTOR    :   A. Ayerbe Garcia
C *
C *           C.T.N.E.
C *           DPTO. DE INVESTIGACION Y DESARROLLO
C *
C *   SECCION  :   Investigacion Aplicada
C *
C *   Fecha de ultima revision : 8 Enero 1987
C *
C *****

```

```

C -----1-----2-----3-----4-----5-----6-----7-
C

```

```

PROGRAM SIM

```

```

C -----1-----2-----3-----4-----5-----6-----7-
C

```

```

COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)
COMMON/COEF/ L,IB,NPUN
COMMON/FUNCT/ TRF(1000),TRFR(1000)
CHARACTER*10, NOMBRE

```

```

PI=4.*ATAN(1.0)

```

```

C N es el orden del filtro. Si N=1, se calcula, si N>1 es un dato
C F1 es el extremo de la banda de paso (Hz).
C F2 es el inicio de la banda eliminada (Hz).
C A1 es la atenuacion maxima en la banda de paso(dB)
C A2 es la atenuacion minima en la banda eliminada.

```

```

TYPE*, '*****DISENO DE FILTROS TAN PASO BAJO*****'

```

```

10 TYPE*, ' Introduzca el orden del filtro:'

```

```

READ(5,*,ERR=20) N
IF(N.LT.1 .OR. N.GT.40) THEN

```

```

20 CALL LECT3(*10)

```

```

END IF
TYPE*, ' Introduzca el orden de bloques :'

```

```

READ(5,*) L
TYPE*, ' Introduzca el numero de bits para redondeo :'

```

```

READ(5,*) IB
NPUN = 999

```

```

30 TYPE*, ' Introduzca la frecuencia de muestreo(Hz):'

```

```

READ(5,*,ERR=40) IFM
IF(IFM.LT.1) THEN

```

```

40 CALL LECT3(*30)

```

```

END IF
TYPE*, ' Introduzca el extremo de la banda de paso(Hz):'

```

```

READ(5,*,ERR=60) F1
IF(F1.LT.1..OR.F1.GT.(IFM/2.)) THEN

```

```

60 CALL LECT3(*50)
   END IF
70 TYPE*, ' Introduzca el inicio de la b. eliminada (Hz)'
   READ(5,*,ERR=80) F2
   IF(F2.LT.1..OR.F2.LT.F1.OR.F2.GT.(IFM/2.)) THEN
80 CALL LECT3(*70)
   END IF
   TYPE*, ' Introduzca el rizado en la banda de paso,(dB)'
   READ(5,*) A1
   TYPE*, ' Introduzca la atenuacion minima en la banda el
* iminada'
   READ(5,*) A2
90 TYPE 100
100 FORMAT(
* ' Introduzca el tipo de aproximacion: '/
* '           1.Butterworth           '/
* '           2.Chebyshev              '/
* '           3.Chebyshev Inverso      '/
* '           4.Eliptico               '/')
   READ(5,*,ERR=110) IPX
   IF(IPX.LT.1.OR.IPX.GT.4) THEN
110 CALL LECT3(*90)
   END IF
   IF(IPX.EQ.1) CALL BUTTERWORTH
   IF(IPX.EQ.2) CALL CHEBYSHEV
   IF(IPX.EQ.3) CALL CHEBYSHEVI
   IF(IPX.EQ.4) CALL ELIPTICO
   CALL CAMBIO

```

C
C Apertura y escritura de ficheros
C

```

TYPE*, ' Escriba el nombre del fichero en maxima precision:'
ACCEPT4, NOMBRE
4 FORMAT(A10)
OPEN(UNIT=3,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')
TYPE*, ' Escriba el nombre del fichero cuantizado:'
ACCEPT4, NOMBRE
OPEN(UNIT=4,FILE=NOMBRE,FORM='UNFORMATTED',STATUS='NEW')

```

C
C Escritura
C

```

WRITE(3) NPUN
WRITE(4) NPUN
WRITE(3) (TRF(I), I=1,NPUN)
WRITE(4) (TRFR(I), I=1,NPUN)
CLOSE(3)
CLOSE(4)
STOP
END

```

C
C-----1-----2-----3-----4-----5-----6-----7-
C INTEGER FUNCTION ICEIL(X)

C-----1-----2-----3-----4-----5-----6-----7-
C

```

IF(X.EQ.0.) THEN
   ICEIL=X
   GO TO 1000
END IF
IF(ABS(X-INT(X)).LT.1.E-12) THEN
   ICEIL=INT(X)
   GO TO 1000
END IF
IF(ABS(X-INT(X)).GT.(1.-1.E-12)) THEN
   A=ABS(X)+1.
   ICEIL=INT(A)
   IF(ICEIL.LT.0) THEN

```

```

        ICEIL=-ICEIL
        END IF
        GO TO 1000
    END IF
    IF(X.GT.0.)THEN
        ICEIL=INT(X+1)
        GO TO 1000
    END IF
    IF(X.LT.0.) THEN
        ICEIL=-INT(ABS(X))
    END IF
1000  RETURN
    END

```

```

C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE LECT3(*)

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
    TYPE*, ' Error en la entrada, introduzca dato de nuevo'
    RETURN 1
    END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE BUTTERWORTH

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
    COMMON/FILT/ P(3,20),C(20)
    COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
    COMMON/DIS/ A1,A2
    COMMON/ESCR/ A(80)

```

```

C
    T1=TAN(F1*PI/IFM)
    E1=SQRT(10.**(A1/10.)-1.)
    AK1=E1/SQRT(10.**(A2/10.)-1.)
    IF(N.GT.1) THEN
        F2=ATAN(T1/AK1**(1./N))/(PI/IFM)
        TYPE*, 'F2=',F2
    ELSE
        AK=T1/TAN(F2*PI/IFM)
        N=ICEIL(LOG10(AK1)/LOG10(AK))
        TYPE*, ' El orden del filtro es ',N
    END IF
    AK=T1/TAN(F2*PI/IFM)
    AL0=1./E1**(1./N)
    X=FLOAT(N-2*INT(N/2.))
    IF(X.NE.0.) THEN
        P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
        C(1)=(1.-P(1,1))/2.
    END IF

```

```

C
    DO I=2,INT(N/2.+1.)
        J=I-1
        Q=2.*AL0*SIN(PI*(2*J-1)/(2.*N))
        P(1,I)=2.*(1.-(AL0*T1)**2)/(1.+Q*T1+(AL0*T1)**2)
        P(2,I)=(1.-Q*T1+(AL0*T1)**2)/(1.+Q*T1+(AL0*T1)**2)
        C(I)=(1.-P(1,I)+P(2,I))/4.
    END DO
    RETURN
    END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE CHEBYSHEV

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C
    COMMON/FILT/ P(3,20),C(20)
    COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI

```

COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**(A1/10.)-1.)
AK1=E1/SQRT(10.**(A2/10.)-1.)
IF(N.GT.1) THEN
F2=ATAN(T1*COSH((1./N)*FNA2(1/AK1)))/(PI/IFM)
TYPE*, 'F2=', F2
ELSE
AK=T1/TAN(F2*PI/IFM)
N=ICEIL(FNA2(1/AK1)/FNA2(1/AK))
TYPE*, ' El orden del filtro es ', N
END IF
AK=T1/TAN(F2*PI/IFM)
AF2=1./N*FNA1(1/E1)
AL0=SINH(AF2)
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
C(1)=(1.-P(1,1))/2.
ELSE
C(1)=1./SQRT(1.+E1**2)
END IF

C
DO I=2,INT(N/2.+1.)
J=I-1
Q=2.*AL0*SIN(PI*(2*J-1)/(2.*N))
B=AL0**2+COS(PI*(2*J-1)/(2.*N))**2
P(1,I)=2.*(1.-B*T1**2)/(1.+Q*T1+B*T1**2)
P(2,I)=(1.-Q*T1+B*T1**2)/(1.+Q*T1+B*T1**2)
C(I)=(1.-P(1,I)+P(2,I))/4.
END DO
RETURN
END

C
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE CHEBYSHEVI
C-----1-----2-----3-----4-----5-----6-----7-
C

COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**(A1/10.)-1.)
AK1=E1/SQRT(10.**(A2/10.)-1.)
IF(N.GT.1) THEN
F2=ATAN(T1*COSH((1./N)*FNA2(1/AK1)))/(PI/IFM)
TYPE*, 'F2=', F2
ELSE
AK=T1/TAN(F2*PI/IFM)
N=ICEIL(FNA2(1/AK1)/FNA2(1/AK))
TYPE*, ' El orden del filtro es ', N
END IF
AK=T1/TAN(F2*PI/IFM)
AF2=1./N*(FNA1(E1*(COSH(N*FNA2(1/AK)))))
AL0=1./(AK*SINH(AF2))
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
C(1)=(1.-P(1,1))/2.
END IF
C
DO I=2,INT(N/2.+1.)

```

J=I-1
Q=2*AL0*SIN(PI*(2*J-1)/(2*N))/(1+(AL0*AK*COS(PI*
* (2*J-1)/(2*N)))**2)
B=AL0**2/(1+(AL0*AK*COS(PI*(2*J-1)/(2*N)))**2)
C1=(AK*COS(PI*(2*J-1)/(2*N)))**2
P(1,I)=2.*(1.-B*T1**2)/(1.+Q*T1+B*T1**2)
P(2,I)=(1.-Q*T1+B*T1**2)/(1.+Q*T1+B*T1**2)
P(3,I)=2*(C1-T1**2)/(C1+T1**2)
C(I)=(1.-P(1,I)+P(2,I))/(2.-P(3,I))
END DO
RETURN
END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE ELIPTICO
C-----1-----2-----3-----4-----5-----6-----7-
C

```

```

COMMON/FILT/ P(3,20),C(20)
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
COMMON/DIS/ A1,A2
COMMON/ESCR/ A(80)

```

```

C
T1=TAN(F1*PI/IFM)
E1=SQRT(10.**(A1/10.)-1.)
AK1=E1/SQRT(10.**(A2/10.)-1.)
IF(N.GT.1) THEN
    Q=PI**2*N/(2*LOG(4.*10.**(A2/20.)/E1))
    F2=ATAN(T1*(8.*EXP(-Q)+1.))/(PI/IFM)
    TYPE*, 'F2=', F2
ELSE
    AK=T1/TAN(F2*PI/IFM)
    AN=2*LOG(4.*10.**(A2/20.)/E1)*LOG(8./(1/AK-1.))/PI**2
    N=ICEIL(AN)
    TYPE*, ' El orden del filtro es ',N

```

```

END IF
AK=T1/TAN(F2*PI/IFM)
AF2=FNK(AK)/N/FNK(AK1)*LOG((1.+SQRT(1.+E1**2))/E1)
A0=AK
Q=FNS(AF2,A0)
A0=SQRT(1.-AK**2)
B1=FNC(AF2,A0)
AL0=Q/B1
X=FLOAT(N-2*INT(N/2.))
IF(X.NE.0.) THEN
    P(1,1)=(1.-AL0*T1)/(1.+AL0*T1)
    C(1)=(1.-P(1,1))/2.
ELSE
    C(1)=1./SQRT(1.+E1**2)
END IF

```

```

C
DO I1=2,INT(N/2.+1.)
L=I1-1
AJ=(N-2.*L+1.)/N*FNK(AK)
Q=2*AL0*FNC(AJ,AK)*FND(AJ,AK)/(1+(AL0*AK*FNS(AJ,AK))**2)
B1=(AL0**2+FNS(AJ,AK)**2)/(1+(AL0*AK*FNS(AJ,AK))**2)
C1=(AK*FNS(AJ,AK))**2
P(1,I1)=2.*(1.-B1*T1**2)/(1.+Q*T1+B1*T1**2)
P(2,I1)=(1.-Q*T1+B1*T1**2)/(1.+Q*T1+B1*T1**2)
P(3,I1)=2.*(C1-T1**2)/(C1+T1**2)
C(I1)=(1.-P(1,I1)+P(2,I1))/(2.-P(3,I1))
END DO
RETURN
END

```

```

C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FNAL(X)

```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
FNA1=LOG(X+SQRT(X**2+1.))
RETURN
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FNA2(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
FNA2=LOG(X+SQRT(X**2-1.))
RETURN
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FNK(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
DIMENSION AEL(30),B(30),EE(30)
PI=4.*ATAN(1.0)
P2=.00001
AEL(1)=1.
B(1)=SQRT(1.-X**2)
EE(1)=X
DO I0=1,19
    AEL(I0+1)=.5*(AEL(I0)+B(I0))
    B(I0+1)=SQRT(AEL(I0)*B(I0))
    EE(I0+1)=.5*(AEL(I0)-B(I0))
IF((EE(I0+1)-P2).LE.0.) GO TO 2000
END DO
2000 J0=I0+1
FNK=.5*PI/AEL(J0)
RETURN
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FNS(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
PI=4.*ATAN(1.0)
AX=X/2./FNK(A0)
Y=EXP(-(PI*FNK(SQRT(1.-A0**2)))/FNK(A0)))
FNS=FNS4(AX,Y)/FNS0(AX,Y)/SQRT(A0)
RETURN
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FNC(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
PI=4.*ATAN(1.0)
AX=X/2./FNK(A0)
Y=EXP(-(PI*FNK(SQRT(1.-A0**2)))/FNK(A0)))
FNC=SQRT(SQRT(1.-A0**2)/A0)*FNS2(AX,Y)/FNS0(AX,Y)
RETURN
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION FND(X,A0)
C-----1-----2-----3-----4-----5-----6-----7-
C
PI=4.*ATAN(1.0)
AX=X/2./FNK(A0)
Y=EXP(-(PI*FNK(SQRT(1.-A0**2)))/FNK(A0)))
FND=SQRT(SQRT(1.-A0**2))*FNS3(AX,Y)/FNS0(AX,Y)
RETURN
```

END

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS0(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S0=1.
      DO I=1,10
      S0=S0+2.*(-1)**I*Y**(I**2)*COS(2.*I*PI*X)
      END DO
      FNS0=S0
      RETURN
      END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS4(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S4=0.
      DO I=1,11
      J=I-1
      S4=S4+2.*(-1)**J*Y**(J*(J+1))*Y**(1./4.)*SIN((2*J+1)*PI*X)
      END DO
      FNS4=S4
      RETURN
      END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS2(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S2=0.
      DO I=1,11
      J=I-1
      S2=S2+2.*Y**(1./4.)*Y**(J*(J+1))*COS((2*J+1)*PI*X)
      END DO
      FNS2=S2
      RETURN
      END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      FUNCTION FNS3(X,Y)
C-----1-----2-----3-----4-----5-----6-----7-
C
      PI=4.*ATAN(1.0)
      S3=1.
      DO I=1,10
      S3=S3+2.*Y**(I**2.)*COS(2*I*PI*X)
      END DO
      FNS3=S3
      RETURN
      END
```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
C
C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE CAMBIO
C-----1-----2-----3-----4-----5-----6-----7-
C
      COMMON/FILT/ P(3,20),C(20)
      COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
      COMMON/COEF/ L,IB,NPUN
```

```

COMMON/MATBR/ EAR(2,2),EBR(2,10),ECR(10,2),EDR(10,10)
COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
COMMON/MATES/ SA(2,2),SB(2,1),SC(1,2),SD
COMMON/MEDP/ AFUN(1000),AFUNR(1000)
COMMON/FUNCT/ TRF(1000),TRFR(1000)
DIMENSION TRAN(20,1000),TRANR(20,1000)

```

C

```

IF((IPX.EQ.1).OR.(IPX.EQ.2)) GO TO 16
IF((IPX.EQ.3).OR.(IPX.EQ.4)) GO TO 17
TYPE*, ' Frecuencia de corte,fc=',Fl

```

16

```

DO I=1,N/2
SA(1,1)=P(1,I+1)
SA(1,2)=-P(2,I+1)
SA(2,1)=1.
SA(2,2)=0.
SB(1,1)=C(I+1)
SB(2,1)=0.
SC(1,1)=2.+P(1,I+1)
SC(1,2)=1.-P(2,I+1)
SD=C(I+1)
CALL MATRICES
CALL FUNH
DO J=1, NPUN
    TRAN(I,J)=AFUN(J)
    TRANR(I,J)=AFUNR(J)

```

```

END DO
END DO
GO TO 20

```

17

```

DO I=1,N/2
SA(1,1)=P(1,I+1)
SA(1,2)=-P(2,I+1)
SA(2,1)=1.
SA(2,2)=0.
SB(1,1)=C(I+1)
SB(2,1)=0.
SC(1,1)=-P(3,I+1)+P(1,I+1)
SC(1,2)=1-P(2,I+1)
SD=C(I+1)
CALL MATRICES
CALL FUNH
DO J=1, NPUN
    TRAN(I,J)=AFUN(J)
    TRANR(I,J)=AFUNR(J)

```

```

END DO
END DO

```

20

```

DO J=1, NPUN
    IF(IPX.EQ.2.OR.IPX.EQ.4) THEN
        TRF(J)=C(1)
    ELSE
        TRF(J)=1.
    END IF
    DO I=1,N/2
        TRF(J)=TRF(J)*TRAN(I,J)
    END DO
END DO
DO J=1, NPUN
    IF(IPX.EQ.2.OR.IPX.EQ.4) THEN
        TRFR(J)=RED(C(1),IB)
    ELSE
        TRFR(J)=1.
    END IF
    DO I=1,N/2
        TRFR(J)=TRFR(J)*TRANR(I,J)
    END DO
END DO

```

C

C Se tienen asi TRF(J) y TRFR(J)

C

```
RETURN
END
```

C

```
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE POTENCIA(SA,R,K)
```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
```

```
DIMENSION SA(2,2),R(2,2),R1(2,2)
```

```
R(1,1)=1.
```

```
R(1,2)=0.
```

```
R(2,1)=0.
```

```
R(2,2)=1.
```

```
IF (K.EQ.0) GOTO 3
```

```
DO IN=1,K
```

```
    DO I=1,2
```

```
        DO J=1,2
```

```
            PAR=0.
```

```
            DO M=1,2
```

```
                PAR=PAR+R(I,M)*SA(M,J)
```

```
            END DO
```

```
            R1(I,J)=PAR
```

```
        END DO
```

```
    END DO
```

```
    DO I=1,2
```

```
        DO J=1,2
```

```
            R(I,J)=R1(I,J)
```

```
        END DO
```

```
    END DO
```

```
END DO
```

```
3 RETURN
```

```
END
```

C

```
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE PROD1(P1,SB,D1)
```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
```

```
DIMENSION P1(2,2),SB(2,1),D1(2,1)
```

```
DO I=1,2
```

```
D1(I,1)=0.
```

```
DO J=1,2
```

```
D1(I,1)=D1(I,1)+P1(I,J)*SB(J,1)
```

```
END DO
```

```
END DO
```

```
RETURN
```

```
END
```

C

```
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE PROD2(P1,SC,D2)
```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
```

```
DIMENSION P1(2,2),SC(1,2),D2(1,2)
```

```
DO I=1,2
```

```
D2(1,I)=0.
```

```
DO J=1,2
```

```
D2(1,I)=D2(1,I)+SC(1,J)*P1(J,I)
```

```
END DO
```

```
END DO
```

```
RETURN
```

```
END
```

C

```
C-----1-----2-----3-----4-----5-----6-----7-
SUBROUTINE MATRICES
```

```
C-----1-----2-----3-----4-----5-----6-----7-
C
```

C

```

COMMON/CALFIL/ N, F1, F2, IFM, IPX, PI
COMMON/COEF/ L, IB, NPUN
COMMON/MATBR/ EAR(2,2), EBR(2,10), ECR(10,2), EDR(10,10)
COMMON/MATB/ EA(2,2), EB(2,10), EC(10,2), ED(10,10)
COMMON/MATES/ SA(2,2), SB(2,1), SC(1,2), SD
DIMENSION P1(2,2), D1(2,1), D2(1,2), R(2,2)

```

```

C
C
C
Calculo de EA(J,I) y de EAR(J,I)

```

```

K=L
CALL POTENCIA(SA,R,K)
DO I=1,2
DO J=1,2
    EA(I,J)=R(I,J)
    EAR(I,J)=RED(EA(I,J),IB)
END DO
END DO

```

```

C
C
C
Calculo de EB(I,J) y EBR(I,J)

```

```

DO M=1,L
K=L-M
CALL POTENCIA(SA,R,K)
DO I=1,2
DO J=1,2
    P1(I,J)=R(I,J)
END DO
END DO
CALL PROD1(P1,SB,D1)
DO IN=1,2
    EB(IN,M)=D1(IN,1)
    EBR(IN,M)=RED(EB(IN,M),IB)
END DO
END DO

```

```

C
C
C
Calculo de EC(I,J) y ECR(I,J)

```

```

DO M=1,L
K=M-1
CALL POTENCIA(SA,R,K)
DO I=1,2
DO J=1,2
    P1(I,J)=R(I,J)
END DO
END DO
CALL PROD2(P1,SC,D2)
DO IN=1,2
    EC(M,IN)=D2(1,IN)
    ECR(M,IN)=RED(EC(M,IN),IB)
END DO
END DO

```

```

C
C
C
Calculo de ED(I,J) y EDR(I,J)

```

```

DO I=1,L
DO J=1,L
IF(I.LT.J) THEN
    ED(I,J)=0.
END IF
IF(I.EQ.J) THEN
    ED(I,J)=SD
    EDR(I,J)=RED(ED(I,J),IB)
END IF
IF(I.GT.J) THEN
K=I-J-1
CALL POTENCIA(SA,R,K)

```

```
DO IL=1,2
DO IM=1,2
```

```
    P1(IL,IM)=R(IL,IM)
```

```
END DO
```

```
END DO
```

```
CALL PROD1(P1,SB,D1)
```

```
ED(I,J)=SC(1,1)*D1(1,1)+SC(1,2)*D1(2,1)
```

```
EDR(I,J)=RED(ED(I,J),IB)
```

```
END IF
```

```
END DO
```

```
END DO
```

```
RETURN
```

```
END
```

```
C
C
C
C
C
C
C-----1-----2-----3-----4-----5-----6-----7
C
C
C
C
C
C-----1-----2-----3-----4-----5-----6-----7-
C
```

PARTE 3.

```

SUBROUTINE FUNH
```

```

Calcula la funcion de transferencia de una seccion
```

```
COMMON/CALFIL/ N,F1,F2,IFM,IPX,PI
```

```
COMMON/COEF/ L,IB,NPUN
```

```
COMMON/MATBR/ EAR(2,2),EBR(2,10),ECR(10,2),EDR(10,10)
```

```
COMMON/MATB/ EA(2,2),EB(2,10),EC(10,2),ED(10,10)
```

```
COMMON/MEDP/ AFUN(1000),AFUNR(1000)
```

```
DIMENSION A0(10,10),A1(10,10),A2(10,10)
```

```
DIMENSION A0R(10,10),A1R(10,10),A2R(10,10)
```

```
DIMENSION AH(10,10),AHR(10,10)
```

```
COMPLEX Z,AH,AHR,DE1,DE2,DE1R,DE2R,FUN,FUNR
```

```
PI =4.*ATAN(1.0)
```

```
C
C Calculo de H(z) y HR(z)
```

```
DO I=1,L
```

```
DO J=1,L
```

```
A0(I,J)=ED(I,J)
```

```
A1(I,J)=EC(I,1)*EB(1,J)+EB(2,J)*EC(I,2)-ED(I,J)*(EA(1,1)+
```

```
* EA(2,2))
```

```
A2(I,J)=(EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2))*ED(I,J)+EA(2,1)*EB(1,J)
```

```
* *EC(I,2)+EA(1,2)*EB(2,J)*EC(I,1)-EA(2,2)*EB(1,J)*EC(I,1)-
```

```
* EA(1,1)*EB(2,J)*EC(I,2)
```

```
END DO
```

```
END DO
```

```
B0=1.
```

```
B1=-EA(1,1)-EA(2,2)
```

```
B2=EA(1,1)*EA(2,2)-EA(2,1)*EA(1,2)
```

```
DO I=1,L
```

```
DO J=1,L
```

```
A0R(I,J)=EDR(I,J)
```

```
A1R(I,J)=ECR(I,1)*EBR(1,J)+EBR(2,J)*ECR(I,2)-EDR(I,J)*(EAR(1,1)+
```

```
* EAR(2,2))
```

```
A2R(I,J)=(EAR(1,1)*EAR(2,2)-EAR(2,1)*EAR(1,2))*EDR(I,J)
```

```
* +EAR(2,1)*EBR(1,J)*ECR(I,2)+EAR(1,2)*EBR(2,J)*ECR(I,1)-
```

```
* EAR(2,2)*EBR(1,J)*ECR(I,1)-EAR(1,1)*EBR(2,J)*ECR(I,2)
```

```
END DO
```

```
END DO
```

```
B0R=1.
```

```
B1R=-EAR(1,1)-EAR(2,2)
```

```
B2R=EAR(1,1)*EAR(2,2)-EAR(2,1)*EAR(1,2)
```

C
C Comienza el calculo de la funcion de transferencia
C
C

C F es la frecuencia normalizada FEQ/FM
C

AMED=0.
TB=F1/IFM
C TB=.5
DO IF=1, NPUN
F=(IF*TB/(NPUN+1.))
Z=CMPLX(COS(2*PI*L*F), SIN(2*PI*L*F))
DO I=1, L
J=1
DE1=A0(I, J)+A1(I, J)*Z**(-1.)+A2(I, J)*Z**(-2.)
DE2=B0+B1*Z**(-1.)+B2*Z**(-2.)
AH(I, J)=DE1/DE2
DE1R=A0R(I, J)+A1R(I, J)*Z**(-1.)+A2R(I, J)*Z**(-2.)
DE2R=B0R+B1R*Z**(-1.)+B2R*Z**(-2.)
AHR(I, J)=DE1R/DE2R
END DO

C
C Calculo de H(z) escalar a partir de la matriz H(z), con y sin
C redondeo
C

FUN=CMPLX(0., 0.)
FUNR=CMPLX(0., 0.)
DO J=1, L
Z=CMPLX(COS(2*PI*F), SIN(2*PI*F))
FUN=FUN+Z**(1.-J)*AH(J, 1)
FUNR=FUNR+Z**(1.-J)*AHR(J, 1)
END DO

C
C
C AFUN(IF)=SQRT(REAL(FUN*CONJG(FUN)))
AFUNR(IF)=SQRT(REAL(FUNR*CONJG(FUNR)))
END DO
RETURN
END

C
C-----1-----2-----3-----4-----5-----6-----7-
REAL FUNCTION RED(X, IH)

C-----1-----2-----3-----4-----5-----6-----7-
C

IF(X.EQ.0.) THEN
RED=0.
ELSE
IEX=ICEIL(ALOG10(ABS(X))/ALOG10(2.))
XP=X*(2.**(-IEX))
REDO=(IFLOOR((2.**IH)*XP+.5))/(2.**IH)
RED=REDO*(2.**IEX)
END IF
RETURN
END

C
C-----1-----2-----3-----4-----5-----6-----7-
FUNCTION IFLOOR(X)

C-----1-----2-----3-----4-----5-----6-----7-
C

IF(X.GT.0.) THEN
IFLOOR =INT(X)
GO TO 10
END IF
IF(X.LT.0.) THEN
IFLOOR=-INT(ABS(X)+.99)
END IF

CAPITULO 3.

OSCILADOR DE BLOQUES.

3.0. INTRODUCCION

Trataremos en este Capítulo de dar una aplicación concreta a todo lo desarrollado en los Capítulos anteriores.

Adoptaremos para la descripción del OSCILADOR DE BLOQUES la descripción en variables de estado, puesto que partiremos del oscilador escalar descrito también en dicha forma.

Un OSCILADOR DIGITAL se puede definir como un sistema digital autónomo capaz de generar la secuencia

$$\{y(n)\} = \{A \operatorname{sen}(nw_0 T_s + \emptyset)\} \quad (3.1)$$

n =entero

$w = 2\pi f_0$, f_0 es la frecuencia de la señal generada

T_s es el periodo de muestreo

\emptyset es el desfase.

Dicha secuencia puede considerarse como la versión muestreada de una onda seno de amplitud A , desfase \varnothing y frecuencia w_0 . Para que sea periódica, ya que podría no serlo, se tendrá que cumplir:

$$w_0 = 2\pi \cdot [M/(N \cdot T_s)] \quad (3.2)$$

donde M, N son enteros, primos entre sí y además $2M < N$. Según lo cual $y(n) = A \sin[2\pi n(M/N) + \varnothing]$ es una señal periódica de periodo NT_s y la frecuencia es $f_0 = M/(NT_s)$, M veces mayor que la frecuencia de la secuencia.

A partir de ahora consideraremos $M=1$ y llamaremos

$$\alpha = w_0 T_s = 2\pi/N \quad (3.3)$$

frecuencia del oscilador normalizada a la frecuencia de muestreo.

Se han propuesto en la Bibliografía [1-6] una gran multitud de osciladores digitales, entre los cuales el método "table look-up" [1,2] ofrece uno de los caminos más flexibles para la generación de señales sinusoidales con un mínimo de carga computacional.

De cualquier forma, cuando se necesita una señal sinusoidal altamente precisa, lo cual significa muchos bits para la representación de los datos, de frecuencia $f_0 \ll f_s$, siendo f_s la frecuencia de muestreo, este método no es práctico debido fundamentalmente al tamaño de la memoria.

Una posible solución es evaluar la función seno usando una aproximación polinómica conveniente, por ejemplo, el desarrollo en serie de Mc.Laurin [2].

Recientemente [7] se ha propuesto un método para generar secuencias sinusoidales basado en el método de recursión lineal. Las estructuras resultantes presentan baja sensibilidad de la frecuencia de oscilación a la cuantización de los coeficientes cuando $f_0 \ll f_s$. Y además de entre todas las estructuras generadas de esa forma, dos de ellas presentan el error cuadrático medio mínimo en un ciclo, es decir, $q^2/12$, siendo q el escalón de cuantización. El ruido de redondeo de la salida para estas estructuras no depende, pues, del número de muestras en un ciclo, o lo que es lo mismo, de la frecuencia de oscilación f_0 . Además de esto, el número de operaciones aritméticas a realizar es: una multiplicación, tres operaciones de "shiftado" y tres sumas por muestra, lo cual representa una carga computacional menor que la que proporciona el método de McLaurin: cuatro multiplicaciones y tres sumas por muestra, además de una decisión lógica.

El propósito de este Capítulo es hacer una extensión del trabajo anterior al caso de realización de osciladores digitales de baja frecuencia empleando estructuras de bloques de estado.

Para ello, nos basaremos en los resultados obtenidos por un gran número de autores sobre técnicas de realización en bloques de estado [8-12] y que han sido resumidas en un Capítulo anterior. No obstante, tendremos que tener en cuenta que un oscilador digital es un

sistema marginalmente estable y que por tanto el problema de establecer restricciones de rango dinámico sobre las variables de estado para controlar el "overflow" y los efectos de redondeo después de las multiplicaciones de dos números se tendrán que estudiar en el dominio del tiempo en vez de en el de la frecuencia.

En el apartado 1, describiremos los principios básicos y un método para la obtención de la matriz escalar de estado a partir de la de bloques.

En el apartado 2, veremos las ecuaciones para determinar las condiciones necesarias de oscilación, la frecuencia de oscilación y los valores iniciales de las variables de estado de un oscilador digital de segundo orden.

En el apartado 3, presentaremos una estructura de bloques de mínima sensibilidad para el caso de interés: $f_0 \ll f_s$.

En el apartado 4, analizaremos el problema de escalado y las formas prácticas de implementación del oscilador digital anteriormente presentado.

En el apartado 5, y después de hacer un estudio del ruido de la estructura, obtendremos una expresión explícita para el error cuadrático medio y veremos que no depende de la frecuencia de oscilación cuando $f_0 \ll f_s$.

En el apartado 6, haremos el mismo estudio anterior para el caso de que las estructuras empleadas para la realización del oscilador sean la de bloques directa y acoplada.

En el apartado 7, daremos los resultados de la simulación de todas estas estructuras poniendo de manifiesto la concordancia entre resultados teóricos y experimentales.

Por último, en el apartado 8, se incluye una comparación de cada oscilador de bloques con su correspondiente escalar en la que quedan patentes las ventajas computacionales de los primeros frente a los últimos.

3.1.- CONSIDERACIONES GENERALES

Un sistema autónomo lineal de segundo orden se puede representar por medio de sus ecuaciones de estado de la siguiente forma:

$$\begin{aligned}\tilde{x}(n+1) &= \underline{A} \cdot \tilde{x}(n) \\ y(n) &= \underline{C} \cdot \tilde{x}(n)\end{aligned}\tag{3.4}$$

donde \underline{A} y \underline{C} son matrices $2 \cdot 2$ y $1 \cdot 2$ respectivamente, $\tilde{x}(n)$ es el vector de estado de dimensión 2, y la salida $y(n)$ tendrá la forma dada por (3.1), siendo $\alpha = \omega_0 T_s$ la frecuencia del oscilador. La realización de dicho sistema se muestra en la Fig. III.1.

Asimismo, y según se ha visto en un Capítulo anterior, dicho sistema podrá ser descrito en términos de sus ecuaciones en bloque de estados de la forma:

$$\begin{aligned}\hat{\tilde{x}}(k+1) &= \hat{\underline{A}} \cdot \hat{\tilde{x}}(k) \\ \hat{\tilde{y}}(k) &= \hat{\underline{C}} \cdot \hat{\tilde{x}}(k)\end{aligned}\quad (3.5)$$

donde, y según [9]

$$\begin{aligned}\hat{\tilde{x}}(k) &= \underline{x}(kL) \\ \hat{\underline{A}} &= \underline{A}^L \\ \hat{\underline{C}} &= \begin{bmatrix} \underline{C} \\ \underline{C} \cdot \underline{A} \\ \cdot \\ \cdot \\ \underline{C} \cdot \underline{A}^{L-1} \end{bmatrix}\end{aligned}\quad (3.6)$$

e $\hat{\tilde{y}}(k)$ es el vector de salida de dimensión L, tal que:

$$\begin{aligned}\hat{\tilde{y}}(k) &= [\hat{y}_1(k) \hat{y}_2(k) \dots \hat{y}_L(k)]^t = \\ &= [y(kL) y(kL+1) \dots y(kL+L-1)]^t\end{aligned}\quad (3.7)$$

con lo cual resulta evidente que la frecuencia característica de esta estructura de bloques será $\hat{\alpha} = \alpha L$.

La realización del oscilador para esta estructura de bloques de estado tiene la forma dada en la Figura III.2.

Teniendo presente que lo que se pretende encontrar es una estructura de bloques de estado de la forma de la Figura III.2 que implemente un oscilador de frecuencia α , tal que se minimice en lo posible la sensibilidad de la misma respecto a la cuantización de los coeficientes empleados en la realización de la estructura, se nos ofrecen dos posibilidades de determinación de los elementos \hat{a}_{ij} de la matriz \hat{A} :

a).- Elegir previamente la estructura escalar para el sistema, determinando, por tanto, los a_{ij} de \underline{A} y luego mediante la expresión (3.6) determinar los \hat{a}_{ij} . Este método presenta el inconveniente de que al producirse la cuantización (de los \hat{a}_{ij} , por supuesto) en general, no sólo se alterará la frecuencia de oscilación sino que además los polos del sistema dejarán de estar sobre el círculo unidad y dejará de ser marginalmente estable.

b).- Elegir directamente la parte recursiva de la estructura de bloques, y por tanto los \hat{a}_{ij} de \hat{A} , con lo cual podrá conseguirse que los polos del oscilador de bloques sigan sobre el círculo unidad al producirse la cuantización, si se ha tomado este criterio para la elección de los \hat{a}_{ij} .

Dadas las ventajas que aporta esta segunda posibilidad, será la que adoptaremos.

Ahora bien, en la determinación de la estructura de bloques y teniendo en cuenta la expresión (3.6), que nos da la matriz \hat{C} , es necesario determinar previamente la matriz \underline{A} , supuesta conocida \underline{C} .

Veamos un procedimiento general de obtención de la matriz de estado escalar \underline{A} a partir de la de bloques $\hat{\underline{A}}$.

Se nos plantea, pues, el problema de, una vez determinada $\hat{\underline{A}}$, determinar la matriz \underline{A} de la estructura escalar asociada.

Como ya hemos visto, si λ es un autovalor de la estructura escalar de estados, λ^L lo es de la de bloques de estado [10], y podemos escribir [13],

$$\hat{\underline{A}} = \underline{A}^L = \underline{Q} \cdot \underline{\Lambda}^L \cdot \underline{Q}^{-1} \quad (3.8)$$

siendo $\underline{\Lambda}$ la matriz diagonal:

$$\underline{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (3.9)$$

λ_1 y λ_2 son los autovalores de la estructura escalar de estados y $\underline{Q} = [\underline{q}_1 \quad \underline{q}_2]$, donde \underline{q}_i es el autovector asociado al autovalor λ_i ($i=1,2$).

Podemos expresar (3.8) de la forma:

$$\hat{\underline{A}} \cdot \underline{Q} = \underline{Q} \cdot \underline{\Lambda}^L \quad (3.10)$$

$$\begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} \\ \hat{a}_{21} & \hat{a}_{22} \end{bmatrix} \cdot \begin{bmatrix} \tilde{q}_1 & \tilde{q}_2 \end{bmatrix} = \begin{bmatrix} \tilde{q}_1 & \tilde{q}_2 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1^L & 0 \\ 0 & \lambda_2^L \end{bmatrix} \quad (3.11)$$

Haciendo operaciones se obtiene para \underline{Q}

$$\underline{Q} = \begin{bmatrix} 1 & 1 \\ \frac{\lambda_1^L - \hat{a}_{11}}{\hat{a}_{12}} & \frac{\lambda_2^L - \hat{a}_{11}}{\hat{a}_{12}} \end{bmatrix} \quad (3.12)$$

Y en general, tendremos

$$\underline{A}^i = \underline{Q} \cdot \underline{\Lambda}^i \cdot \underline{Q}^{-1} \quad (3.13)$$

en donde sustituyendo (3.9) y (3.12) se obtiene:

$$\underline{A}^i = (1/|Q|) \cdot \begin{bmatrix} \frac{\lambda_1^i (\lambda_2^L - \hat{a}_{11})}{\hat{a}_{12}} - \frac{\lambda_2^i (\lambda_1^L - \hat{a}_{11})}{\hat{a}_{12}} \\ \frac{\lambda_1^i (\lambda_1^L - \hat{a}_{11}) (\lambda_2^L - \hat{a}_{11})}{\hat{a}_{12}^2} - \frac{\lambda_2^i (\lambda_2^L - \hat{a}_{11}) (\lambda_1^L - \hat{a}_{11})}{\hat{a}_{12}^2} \end{bmatrix}$$

$$\left. \begin{aligned} & - \lambda_1^i + \lambda_2^i \\ & \frac{- \lambda_1^i (\lambda_1^L - \hat{a}_{11})}{\hat{a}_{12}} + \frac{\lambda_2^i (\lambda_2^L - \hat{a}_{11})}{\hat{a}_{12}} \end{aligned} \right] \quad (3.14)$$

y

$$|Q| = (1/\hat{a}_{12}) \cdot (\lambda_2^L - \lambda_1^L) \quad (3.15)$$

En este caso particular, el sistema es un oscilador y por tanto :

$$\lambda_1^L = e^{jL\alpha} \quad , \quad \lambda_2^L = e^{-jL\alpha}$$

sustituyendo en (3.14) y teniendo en cuenta que:

$$\begin{aligned} \hat{a}_{11} + \hat{a}_{22} &= 2 \cdot \cos(L\alpha) \\ \hat{a}_{11}\hat{a}_{22} - \hat{a}_{12}\hat{a}_{21} &= 1 \end{aligned} \quad (3.16)$$

se tiene

$$\underline{A}^i = (1/\text{sen}(L\alpha)) \begin{bmatrix} \text{sen}[(L-i)\alpha] + \hat{a}_{11}\text{sen}(i\alpha) & \hat{a}_{12}\text{sen}(i\alpha) \\ \hat{a}_{21}\text{sen}(i\alpha) & \text{sen}[(L-i)\alpha] + \hat{a}_{22}\text{sen}(i\alpha) \end{bmatrix} \quad (3.17)$$

Evidentemente para $i=1$, obtendremos la matriz \underline{A} de la estructura de estados escalar:

$$\underline{A} = (1/\text{sen}(L\alpha)) \cdot \begin{bmatrix} \text{sen}[(L-1)\alpha] + \hat{a}_{11} \cdot \text{sen}(\alpha) & \hat{a}_{12} \cdot \text{sen}(\alpha) \\ \hat{a}_{21} \cdot \text{sen}(\alpha) & \text{sen}[(L-1)\alpha] + \hat{a}_{22} \cdot \text{sen}(\alpha) \end{bmatrix} \quad (3.18)$$

Conocida \underline{A} , si suponemos que de alguna forma determinamos $\underline{C} = [c_1 \ c_2]$, la matriz $\hat{\underline{C}}$ característica de la estructura de bloques será:

$$\hat{\underline{C}} = [\underline{C} : \underline{C} \cdot \underline{A} : \underline{C} \cdot \underline{A}^2 \cdot \dots \cdot \underline{C} \cdot \underline{A}^{L-1}]^t = [\hat{c}_{ij}] \quad (3.19)$$

donde:

$$\begin{aligned} \hat{c}_{i1} &= (c_1/\text{sen}(L\alpha)) \{ \text{sen}[(L-i+1)\alpha] + \hat{a}_{11} \cdot \text{sen}[(i-1)\alpha] \} + \\ &\quad + (c_2/\text{sen}(L\alpha)) \cdot \hat{a}_{21} \cdot \text{sen}[(i-1)\alpha] \\ \hat{c}_{i2} &= (c_1 \hat{a}_{12} / \text{sen}(L\alpha)) \cdot \text{sen}[(i-1)\alpha] + \\ &\quad + (c_2/\text{sen}(L\alpha)) \{ \text{sen}[(L-i+1)\alpha] + \hat{a}_{22} \cdot \text{sen}[(i-1)\alpha] \} \\ &\quad i=1, 2, \dots, L \end{aligned} \quad (3.20)$$

Resumiendo, en primer lugar se determinará la matriz \underline{A} de bloques de estado siguiendo algún criterio, como por ejemplo en el caso del oscilador, el criterio de minimización de la sensibilidad de la fre-

cuencia de oscilación respecto a la cuantización de los coeficientes. A continuación, y a través de (3.18) se tendrá la matriz \underline{A} de la estructura escalar correspondiente y por último mediante la elección de las constantes c_1 y c_2 y por medio de (3.20) se tendrá la matriz \hat{C} y con ello la estructura de bloques quedará perfectamente determinada.

3.2.- OSCILADOR DIGITAL DE BLOQUES.

Lógicamente, la generación de una secuencia sinusoidal implicará la imposición de unas ciertas condiciones sobre los elementos de $\hat{\underline{A}}$ y $\hat{\underline{C}}$.

Si aplicamos la transformada-z a las ecuaciones (3.5) tendremos:

$$z \cdot \begin{bmatrix} \hat{X}_1(z) \\ \hat{X}_2(z) \end{bmatrix} - z \cdot \begin{bmatrix} \hat{x}_1(0) \\ \hat{x}_2(0) \end{bmatrix} = \hat{\underline{A}} \cdot \begin{bmatrix} \hat{X}_1(z) \\ \hat{X}_2(z) \end{bmatrix} \quad (3.21)$$

siendo:

$$\hat{X}_i(z) = \sum_{k=0}^{\infty} x_i(k) \cdot z^{-k} = \sum_{k=0}^{\infty} x_i(kL) \cdot z^{-k} \quad i=1,2$$

y

$$\hat{Y}(z) = \hat{C} \begin{bmatrix} \hat{X}_1(z) \\ \hat{X}_2(z) \end{bmatrix} \quad (3.22)$$

donde

$$\begin{aligned} \hat{Y}(z) &= [\hat{Y}_1(z) \quad \hat{Y}_2(z) \quad \dots \quad \hat{Y}_L(z)]^t \\ \hat{Y}_i(z) &= \sum_{k=0}^{\infty} y(kL+i-1) \cdot z^{-k} \end{aligned} \quad (3.23)$$

Operando en (3.21), tendremos para las transformadas z de las variables de estado:

$$\hat{X}_1(z) = \frac{z^2 \cdot \hat{x}_1(0) + z[-\hat{a}_{22}\hat{x}_1(0) + \hat{a}_{12}\hat{x}_2(0)]}{D(z)} \quad (3.24)$$

$$\hat{X}_2(z) = \frac{z^2 \cdot \hat{x}_2(0) + z[\hat{a}_{21}\hat{x}_1(0) - \hat{a}_{11}\hat{x}_2(0)]}{D(z)}$$

$$\text{con } D(z) = z^2 - z(\hat{a}_{11} + \hat{a}_{22}) + (\hat{a}_{11}\hat{a}_{22} - \hat{a}_{12}\hat{a}_{21}) \quad (3.25)$$

Identificando estas expresiones con la transformada z de una señal sinusoidal [14]:

$$\frac{z^2 A \cdot \text{sen}(\psi) + zA \cdot \text{sen}(\hat{\alpha} - \psi)}{z^2 - 2z \cdot \cos(\hat{\alpha}) + 1} \longleftrightarrow A \cdot \text{sen}(k\hat{\alpha} + \psi) \quad (3.26)$$

se tendrá para las variables de estado:

$$\begin{aligned}\hat{x}_1(k) &= A[\hat{x}_1(k)] \cdot \text{sen}(k\hat{\alpha} + \psi_1) \\ \hat{x}_2(k) &= A[\hat{x}_2(k)] \cdot \text{sen}(k\hat{\alpha} + \psi_2)\end{aligned}\tag{3.27}$$

siendo

$$\begin{aligned}A[\hat{x}_1(k)] &= \left[\hat{x}_1^2(0) + \frac{[(\cos(\hat{\alpha}) - \hat{a}_{22})\hat{x}_1(0) + \hat{a}_{12}\hat{x}_2(0)]^2}{\text{sen}^2(\hat{\alpha})} \right]^{\frac{1}{2}} \\ A[\hat{x}_2(k)] &= \left[\hat{x}_2^2(0) + \frac{[\hat{a}_{21}\hat{x}_1(0) + (\cos(\hat{\alpha}) - \hat{a}_{11})\hat{x}_2(0)]^2}{\text{sen}^2(\hat{\alpha})} \right]^{\frac{1}{2}}\end{aligned}$$

$$\begin{aligned}\psi_1 &= \text{sen}^{-1} [\hat{x}_1(0)/A[\hat{x}_1(k)]] \\ \psi_2 &= \text{sen}^{-1} [\hat{x}_2(0)/A[\hat{x}_2(k)]]\end{aligned}\tag{3.28}$$

Análogamente, se tiene para la componente i -ésima del vector de salida:

$$\begin{aligned}\hat{Y}_i(z) &= \frac{z^2 \cdot [\hat{c}_{i1}\hat{x}_1(0) + \hat{c}_{i2}\hat{x}_2(0)] + z \cdot [(\hat{c}_{i2}\hat{a}_{21} - \hat{a}_{22}\hat{c}_{i1})\hat{x}_1(0) + (\hat{c}_{i1}\hat{a}_{12} - \hat{c}_{i2}\hat{a}_{11})\hat{x}_2(0)]}{D(z)} \\ & \quad i=1, \dots, L\end{aligned}\tag{3.29}$$

Identificando esta expresión con la transformada z de una señal sinusoidal tendremos:

$$\begin{aligned}
 \hat{a}_{11} + \hat{a}_{22} &= 2\cos(\hat{\alpha}) \\
 \hat{a}_{11}\hat{a}_{22} - \hat{a}_{12}\hat{a}_{21} &= 1 \\
 \hat{c}_{i1}\hat{x}_1(0) + \hat{c}_{i2}\hat{x}_2(0) &= A_i \cdot \text{sen}(\varphi_i) \\
 (\hat{c}_{i2}\hat{a}_{21} - \hat{a}_{22}\hat{c}_{i1})\hat{x}_1(0) + (\hat{c}_{i1}\hat{a}_{12} - \hat{c}_{i2}\hat{a}_{11})\hat{x}_2(0) &= A_i \cdot \text{sen}(\hat{\alpha} - \varphi_i)
 \end{aligned} \tag{3.30}$$

donde $\hat{\alpha} = L\alpha$, e $\hat{y}_i(k) = A_i \cdot \text{sen}(k\hat{\alpha} + \varphi_i)$

Las dos primeras ecuaciones determinan la frecuencia de oscilación y la condición necesaria para oscilación respectivamente y las dos últimas ecuaciones dan la amplitud y la fase de cada una de las L secuencias de salida del oscilador.

Despejando A_i se tiene:

$$\begin{aligned}
 A_i &= [[\hat{c}_{i1}\hat{x}_1(0) + \hat{c}_{i2}\hat{x}_2(0)]^2 + \\
 &+ \frac{(\hat{x}_1(0) \cdot [\hat{c}_{i2}\hat{a}_{21} - \hat{a}_{22}\hat{c}_{i1}] + \hat{c}_{i1}\cos(\hat{\alpha}) + \hat{x}_2(0) \cdot [\hat{c}_{i1}\hat{a}_{12} - \hat{c}_{i2}\hat{a}_{11}] + \hat{c}_{i2}\cos(\hat{\alpha}))^2}{\text{sen}^2\hat{\alpha}}]^{1/2}
 \end{aligned}$$

$$\varphi_i = \text{sen}^{-1} \left(\frac{\hat{c}_{i1}\hat{x}_1(0) + \hat{c}_{i2}\hat{x}_2(0)}{A_i} \right) \quad i=1, \dots, L \tag{3.31}$$

Puesto que lo que se pretende con la estructura de bloques es la generación de una secuencia sinusoidal $y(n)=A \text{ sen}(n\alpha+\varnothing)$, resultará que teniendo en cuenta (3.7)

$$\begin{aligned} \hat{y}_i(k) &= y(kL+i-1) = A \cdot \text{sen}[(kL+i-1)\alpha + \varnothing] = \\ &= A \cdot \text{sen}[k\hat{\alpha} + (i-1)\alpha + \varnothing] \quad i=1, \dots, L \end{aligned} \quad (3.32)$$

Identificando (3.30) con (3.32) se tendrá:

$$\begin{aligned} A_i &= A \\ \varnothing_i &= (i-1)\alpha + \varnothing \quad i=1, \dots, L \end{aligned} \quad (3.33)$$

Sin pérdida de generalidad podemos considerar $A=1$, $\varnothing=0$, con lo que

$$\begin{aligned} A_i &= 1 \\ \varnothing_i &= (i-1)\alpha \end{aligned} \quad (3.34)$$

Imponiendo estas condiciones sobre las dos últimas ecuaciones de (3.30) se tiene para cada i un sistema de ecuaciones con las condiciones iniciales de las variables de estado como incógnitas

$$\begin{aligned} \hat{c}_{i1} \hat{x}_1(0) + \hat{c}_{i2} \hat{x}_2(0) &= \text{sen}[(i-1)\alpha] \\ (\hat{c}_{i2} \hat{a}_{21} - \hat{a}_{22} \hat{c}_{i1}) \hat{x}_1(0) + (\hat{c}_{i1} \hat{a}_{12} - \hat{c}_{i2} \hat{a}_{11}) \hat{x}_2(0) &= \\ &= \text{sen}(L\alpha) \cdot \cos[(i-1)\alpha] - \text{sen}[(i-1)\alpha] \cdot \cos(L\alpha) \end{aligned} \quad (3.35)$$

$i=1, \dots, L$

Particularizando para $i=1$, y tomando c_1 y c_2 cualesquiera, se tendrá para las condiciones iniciales:

$$\hat{x}_1(0) = \frac{-c_2 \cdot \text{sen}(L\alpha)}{\Delta} \quad (3.36)$$

$$\hat{x}_2(0) = \frac{c_1 \cdot \text{sen}(L\alpha)}{\Delta}$$

con $\Delta = c_1[c_1\hat{a}_{12} - c_2\hat{a}_{11}] - c_2[c_2\hat{a}_{21} - c_1\hat{a}_{22}]$, $c_1 = \hat{c}_{11}$, $c_2 = \hat{c}_{12}$ valores que satisfacen los $(L-1)$ pares de ecuaciones restantes, como se puede demostrar fácilmente.

En resumen, si los elementos de la matriz de estados \hat{A} se eligen de forma que satisfagan las dos primeras ecuaciones en (3.30) y se eligen los valores iniciales de las variables de estado dados por (3.36), entonces la salida de la estructura de bloques dada por la Figura III.2 es la secuencia sinusoidal $y = \text{sen}(n\alpha)$.

De todo lo expuesto queda patente que existen dos grados de libertad dados por el vector \underline{c} , con lo cual podrán obtenerse distintas estructuras, resultado análogo al ya obtenido en [15].

3.3.- ESTRUCTURA DE OSCILADOR EN BLOQUE DE ESTADO CON SENSIBILIDAD MINIMA.

En la búsqueda de una estructura de bloques para el oscilador con sensibilidad mínima, se pretende que los polos del sistema se mantengan sobre el círculo unidad aún después de producida la cuantización y además que ésta afecte lo menos posible a la frecuencia de oscilación α . Esto se puede expresar:

$$\hat{a}_{11}|_q \cdot \hat{a}_{22}|_q - \hat{a}_{12}|_q \cdot \hat{a}_{21}|_q = 1 \quad (3.37)$$

$$\Delta\alpha = \sum S_{\hat{a}_{ij}}^\alpha \cdot \Delta\hat{a}_{ij} \quad \text{minima}$$

siendo $S_{\hat{a}_{ij}}^\alpha = \partial\alpha/\partial\hat{a}_{ij}$

y $x|_q$ = coeficiente x sometido a cuantización.

Si siguiendo este procedimiento para estructuras escalares, Acha et Al.[7] han propuesto varias estructuras para osciladores que reproducimos a continuación:

$$\begin{bmatrix} 2^{-\gamma} + \beta & 2^{\gamma} \\ \beta & 2^{\gamma} \end{bmatrix} \quad \text{Estructura 1} \quad (3.38)$$

$$\begin{bmatrix} \beta & 2^{\gamma} \\ -2^{-\gamma} + \beta & 2^{\gamma} \end{bmatrix} \quad \text{Estructura 2} \quad (3.39)$$

En ellas existe un único coeficiente sometido a cuantización que es β .

Puesto que la parte recursiva de la estructura de bloques del oscilador es la que determina la frecuencia de oscilación $\hat{\alpha} = \alpha L$, conjeturamos que estructuras topológicamente idénticas a las escalares pueden conducir a buenos resultados sobre la sensibilidad, que es lo que en definitiva deseamos.

Así pues, podemos proponer como posibles estructuras de bloque para el oscilador las correspondientes a la matriz \hat{A} con la forma:

$$\begin{bmatrix} 2^{-\hat{\gamma}} + \hat{\beta} & 2^{\hat{\gamma}} \\ \hat{\beta} & 2^{\hat{\gamma}} \end{bmatrix} \quad (3.40)$$

$$Y \quad \begin{bmatrix} \hat{B} & 2^{\hat{Y}} \\ -2^{-\hat{Y}} + \hat{B} & 2^{\hat{Y}} \end{bmatrix} \quad (3.41)$$

En el primer caso, $\hat{B} = 2\cos(\hat{\alpha}) - [2^{\hat{Y}} + 2^{-\hat{Y}}]$ (3.42)

y en el segundo, $\hat{B} = 2\cos(\hat{\alpha}) - 2^{\hat{Y}}$ (3.43)

Sólo existe un coeficiente sometido a cuantización en ambas estructuras, \hat{B} , y puede encontrarse fácilmente [7] que la sensibilidad normalizada de la frecuencia de oscilación respecto a la cuantización del mismo es, en ambos casos, :

$$S_{NB}^{\alpha} = (\hat{B}/\alpha) \cdot (\partial\alpha/\partial\hat{B}) = [1/|2\text{sen}(\hat{\alpha})|] \cdot |\hat{B}/\hat{\alpha}| \quad (3.44)$$

Lógicamente esa expresión será mínima para $|\hat{B}|$ mínima, con lo cual tenemos un criterio para la elección de \hat{Y} , es decir, tomaremos \hat{Y} en cada estructura de manera que se tenga el \hat{B} menor.

Así, para la estructura (3.40) y según [7] se toma $\hat{Y}=0$, con lo que $|\hat{B}|=2-2\cos(\hat{\alpha})$, y para (3.41) se toma \hat{Y} de forma que $2^{\hat{Y}}$ se acerque lo más posible a $2\cos(\hat{\alpha})$, es decir $\hat{Y}=1$. De esta forma, las estructuras (3.40) y (3.41) se transforman respectivamente en:

$$\begin{bmatrix} 1+\hat{\beta} & 1 \\ \hat{\beta} & 1 \end{bmatrix} \quad (3.45)$$

$$\begin{bmatrix} \hat{\beta} & 2 \\ -2^{-1}+\hat{\beta} & 2 \end{bmatrix} \quad (3.46)$$

Aprovechando los resultados obtenidos en [7] para ambas estructuras, parece que la primera conduce a mejores resultados sobre ruido y sensibilidad, con lo cual será ésta la que estudiaremos.

La parte recursiva de dicha estructura puede observarse en la Figura III.3.

Así pues, la matriz de bloque de estados \hat{A} vendrá dada por (3.45) y la sensibilidad de la frecuencia de oscilación por:

$$S^{\alpha}_{NB\hat{A}} = |\cos(L\alpha)-1| / |L\alpha \cdot \text{sen}(L\alpha)| \quad (3.47)$$

que para osciladores de baja frecuencia ($\alpha \rightarrow 0$) es:

$$S^{\alpha}_{NB\hat{A}} \approx 1/2 \quad (3.48)$$

valor obtenido también en [7].

Además, la sensibilidad radial es nula, y por tanto, el oscilador con los coeficientes cuantizados seguirá teniendo sus polos sobre el círculo unidad, es decir, seguirá siendo marginalmente estable.

3.4.- ESCALADO DE LA SEÑAL

Los factores de escalado se pueden obtener a partir de las expresiones en el dominio del tiempo de las variables de estado $\hat{x}_1(k)$ y $\hat{x}_2(k)$ dados en (3.27).

Dichos factores, y para el caso general de una estructura cualquiera, vienen dados en (3.28), donde $A[\hat{x}_i(k)]$ es la amplitud o valor de pico de la secuencia $\hat{x}_i(k)$, y los estados iniciales $\hat{x}_1(0)$ y $\hat{x}_2(0)$ vienen dados por (3.36).

Ahora bien, para el caso que nos interesa, es decir, osciladores de baja frecuencia, $\alpha \rightarrow 0$, con lo que $\hat{B} = 2\cos(\hat{\alpha}) - 2 = -\hat{\alpha}^2$. Simplificando (3.36) puede obtenerse:

$$\begin{aligned}\hat{x}_1(0) &= -c_2 \cdot \hat{\alpha} / d(\hat{\alpha}) \\ \hat{x}_2(0) &= c_1 \cdot \hat{\alpha} / d(\hat{\alpha})\end{aligned}\tag{3.49}$$

donde $d(\hat{\alpha}) = c_1^2 \hat{a}_{12} - c_1 c_2 (\hat{a}_{11} - \hat{a}_{22}) - c_2^2 \hat{a}_{21} = c_1^2 + c_2^2 \hat{\alpha}^2 +$

$c_1 c_2 \hat{\alpha}^2$ y sustituyendo en (3.28), (3.45) y (3.49) se tendrá:

$$A[\hat{x}_1(k)] = [1/d(\hat{\alpha})]^{\frac{1}{2}}$$

$$A[\hat{x}_2(k)] = \frac{\hat{\alpha}[c_1^2 + c_2^2 \hat{\alpha}^2 + c_1 c_2 \hat{\alpha}^2 + (c_1^2/4) \hat{\alpha}^2]^{\frac{1}{2}}}{d(\hat{\alpha})} \quad (3.50)$$

Hemos de volver a hacer constar ahora que se podrán generar distintas estructuras para el oscilador de bloques con la misma matriz \hat{A} , simplemente a través de la elección de las constantes c_1 y c_2 . En cada caso, se tendrán unas condiciones iniciales distintas y unos valores para el escalado, además de unas constantes \hat{c}_{ij} , distintas también.

Para sacar toda la ventaja posible de este grado de libertad de que disponemos, conviene tener en cuenta las siguientes ideas:

1. Para polos próximos a $z=1$, las estructuras que tienen acumuladores y registros largos reducen considerablemente los errores debidos a la cuantización de los productos, aunque comparativamente la longitud de las señales y de las multiplicaciones se reduce fuertemente [16,17].

2. El pleno uso de registros y acumuladores largos es útil para aumentar la relación señal/ruido.

3. La reducción del número de multiplicadores es útil para la reducción de la carga computacional.

La consideración de estas ideas nos llevará posteriormente a tomar $c_1=0$ y $c_2=1$, lo cual dará lugar a una estructura concreta para el oscilador de bloque, que será la que simularemos.

Definimos ahora las variables s_1 y s_2 como:

$$\begin{aligned} s_1 &= \lceil \log_2 |A[\hat{x}_1(k)]| \rceil \\ s_2 &= \lceil \log_2 |A[\hat{x}_2(k)]| \rceil \end{aligned} \quad (3.51)$$

donde $\lceil x \rceil$ es el menor entero mayor o igual que x . s_1 y s_2 no son más que el número de bits necesarios para representar la parte entera de las variables de estado y por tanto dan la longitud de los acumuladores en las variables de estado.

Considerando $s = \max(s_1, s_2)$, y utilizando la misma longitud para los acumuladores, el circuito escalado será el de la Figura III.4, de la cual podemos tener los casos particulares:

$$\begin{aligned} s &= s_1, s_2 = 0 \\ s &= s_2, s_1 = 0 \end{aligned}$$

representados en la Figura III.5.

En estos circuitos se supone una longitud de $(2+B)$ bits (uno para la parte entera) para la señal en cualquier punto del oscilador, excepto para el acumulador, para el que se suponen $(2+s+B)$ bits.

En el caso particular de que la elección de constantes c_i sea $c_1=0$, $c_2=1$, sustituyendo en (3.51), se obtiene:

$$\begin{aligned} s_1 &= \lceil \log_2 |1/\hat{\alpha}| \rceil \\ s_2 &= 0 \end{aligned} \quad (3.52)$$

estructura que corresponde a la Figura III.4(a). Para la cual la matriz \underline{A}^i dada en (3.17), para $L\alpha \rightarrow 0$, se reduce a :

$$\underline{A}^i \approx \begin{bmatrix} 1+(i\hat{\beta}/L) & (i/L) \\ (i\hat{\beta}/L) & 1 \end{bmatrix} \quad i=1, \dots, L-1 \quad (3.53)$$

y los elementos \hat{c}_{ij} de \hat{C} , sustituyendo (3.45) en (3.20) y haciendo $L\alpha \rightarrow 0$, vienen dados por:

$$\begin{aligned} \hat{c}_{i1} &\approx [(i-1)\hat{\beta}]/L \\ \hat{c}_{i2} &\approx 1 \end{aligned} \quad (3.54)$$

3.5.- EFFECTOS DE LA CUANTIZACION DE PRODUCTOS:RUIDO.

Nos interesa ahora considerar los efectos de la cuantización de los productos en la estructura propuesta sobre la secuencia de salida.

Haremos un estudio, pues, del ruido de la estructura de la Figura III.4 producido por la cuantización de productos, particularizando posteriormente al caso de $c_1=0$, $c_2=1$ (Fig.III.5a).

El método estocástico da una medida razonable de los efectos de la cuantización de productos sobre un ciclo de la secuencia de salida cuando $f_0 \ll f_s$ [7]. Así pues, emplearemos este modelo para hacer una evaluación de los errores de cuantización.

El modelo de ruido para el oscilador de bloques de la Figura III.4 es el de la Figura III.6, cuyas ecuaciones de estado son:

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 1+\hat{\beta} & 2^{s2-s1} \\ \hat{\beta}2^{-(s2-s1)} & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \\
+ \begin{bmatrix} 1 & 2^{-s} & \hat{\beta}2^s & 2^{s+s2-s1} & 0 & 1 & 0 \\ 0 & 2^{s1-s2-s} & \hat{\beta}2^{s1-s2+s} & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \\ \vdots \\ e_6(k) \\ e_0(k) \end{bmatrix}$$

(3.55)

y la salida $\hat{y}_i(k)$ viene dada por:

$$\hat{y}_i(k) = \begin{bmatrix} \hat{c}_{i1} \cdot 2^{s1-s} & \hat{c}_{i2} \cdot 2^{s2-s} \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \\
+ \begin{bmatrix} 0 & 0 & \hat{c}_{i1} \cdot 2^{s1} & \hat{c}_{i2} \cdot 2^{s2} & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \\ \vdots \\ e_6(k) \\ e_0(k) \end{bmatrix}$$

(3.56)

donde $e_k(k)$ son las fuentes de ruido.

Suponiendo condiciones iniciales nulas y tomando transformadas a ambos lados de las ecuaciones (3.55) y (3.56), podemos obtener la matriz de transferencia, $\hat{G}(z)$, cuyos elementos $\hat{G}_{ik}(z)$ son la función de

transferencia de la fuente de error k a la salida i-ésima del oscilador.

$$\hat{\tilde{x}}(k+1) = \hat{\underline{A}}_r \cdot \hat{\tilde{x}}(k) + \hat{\underline{B}}_r \cdot \tilde{e}(k) \quad (3.57)$$

$$\hat{y}_i(k) = \hat{\underline{C}}_{ri} \cdot \hat{\tilde{x}}(k) + \hat{\underline{D}}_{ri} \cdot \tilde{e}(k)$$

$$\hat{G}_i(z) = \hat{\underline{C}}_{ri} \cdot (z\underline{I} - \hat{\underline{A}}_r)^{-1} \cdot \hat{\underline{B}}_r + \hat{\underline{D}}_{ri} \quad (3.58)$$

Sustituyendo y haciendo operaciones tenemos:

$$\hat{G}_{i1}(z) = (2^{s1-s}/D(z)) \cdot (\hat{c}_{i1}(z-1) + \hat{B}\hat{c}_{i2})$$

$$\hat{G}_{i2}(z) = (2^{s1-2s}/D(z)) \cdot (z(\hat{c}_{i1} + \hat{c}_{i2}) - \hat{c}_{i2})$$

$$\hat{G}_{i3}(z) = (2^{s1}/D(z)) \cdot (c_{i1}z^2 - z(2\hat{c}_{i1} - \hat{B}\hat{c}_{i2}) + \hat{c}_{i1} - \hat{B}\hat{c}_{i2})$$

$$\hat{G}_{i4}(z) = (2^{s2}/D(z)) \cdot (\hat{c}_{i2}z^2 - z(2\hat{c}_{i2} + \hat{B}\hat{c}_{i2} - \hat{c}_{i1}) + \hat{c}_{i2} + \hat{B}\hat{c}_{i2} - \hat{c}_{i1})$$

$$\hat{G}_{i5}(z) = (2^{s2-s}/D(z)) \cdot (\hat{c}_{i2}z + \hat{c}_{i1} - \hat{c}_{i2}(1 + \hat{B}))$$

$$\hat{G}_{i6}(z) = \hat{G}_{i1}(z) \quad i=1 \dots L$$

$$\hat{G}_{i0}(z) = 1 \quad (3.59)$$

donde $D(z) = z^2 - 2z \cdot \cos(\hat{\alpha}) + 1 \quad (3.60)$

En el caso particular, $c_1=0$, $c_2=1$,
 $\hat{G}_{i5}=\hat{G}_{i6}=\hat{G}_{i10}=0$.

Suponiendo $(2+B)$ bits para la señal en cualquier punto del oscilador excepto para los acumuladores y registros siguientes, donde se emplean $(2+s+B)$ bits, el error cuadrático medio de la salida i del oscilador en un ciclo viene dado por:

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \cdot \sum_{k=1}^6 \sum_{m=0}^{\hat{N}} h_{ik}^2(m) + (q^2/12) \cdot 2 \quad (3.61)$$

Y en el caso particular, $c_1=0, c_2=1$,

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \cdot \sum_{k=1}^4 \sum_{m=0}^{\hat{N}} h_{ik}^2(m) + (q^2/12)(1-\delta_{i1}) \quad (3.62)$$

donde, $\hat{N}=N/L$, $q=2^{-B}$ =escalón de cuantización y δ_{ij} es la delta de Kronecker:

$$\delta_{ij} = 1, \text{ si } i=j, \quad \delta_{ij}=0, \text{ si } i \neq j$$

Tenemos, pues, que hallar $\hat{h}_{ik}(n)$

$$\hat{h}_{ik}(n) \longleftrightarrow \hat{G}_{ik}(z)$$

para lo cual tendremos en cuenta los pares de transformadas de la Tabla 3.I.

Una vez calculados los $\hat{h}_{ij}(n)$, habrá que calcular $\sum_{n=0}^{\hat{N}} \hat{h}_I^2(n)$ y $\sum_{n=0}^{\hat{N}} \hat{h}_{II}^2(n)$. Para lo cual emplearemos

las relaciones de la Tabla 3.II, así como el hecho de que \hat{N} es el valor del índice de la secuencia $\text{sen}(n\hat{\alpha})$ para la cual se recorre un periodo completo de la señal, luego

$$\begin{aligned} \text{sen}[(\hat{N}+1)\hat{\alpha}] &= \text{sen}(\hat{\alpha}) \\ \text{sen}[(\hat{N}-1)\hat{\alpha}] &= -\text{sen}(\hat{\alpha}) \\ \text{sen}(2\hat{N}\hat{\alpha}) &= 0 \\ \text{sen}[(2\hat{N}+3)\hat{\alpha}] &= \text{sen}(3\hat{\alpha}) \end{aligned} \tag{3.63}$$

Calculemos en primer lugar $\sum_{n=0}^{\hat{N}} \hat{h}_I^2(n)$

$$\begin{aligned} \sum_{n=0}^{\hat{N}} \hat{h}_I^2(n) &= \sum_{n=1}^{\hat{N}} [\hat{h}_I(n) + b \cdot \delta(n)]^2 = \\ &= \sum_{n=1}^{\hat{N}} \{ [a \cdot \text{sen}(n\hat{\alpha}) - b \cdot \text{sen}[(n-1)\hat{\alpha}]] / \text{sen}(\hat{\alpha}) \}^2 = \\ &= (1/\text{sen}^2\hat{\alpha}) \{ \sum_{n=1}^{\hat{N}} a^2 \text{sen}^2 n\hat{\alpha} + b^2 \text{sen}^2((n-1)\hat{\alpha}) - 2ab \cdot \text{sen}(n\hat{\alpha}) \cdot \text{sen}[(n-1)\hat{\alpha}] \} \end{aligned}$$

Luego,

$$\sum_{n=0}^{\hat{N}} \hat{h}_I^2(n) = [\hat{N}/(2 \cdot \text{sen}^2\hat{\alpha})] \cdot (a^2 + b^2 - 2ab \cdot \cos(\hat{\alpha})) \tag{3.64}$$

y particularizando para el caso de que $\hat{\alpha}$ tienda a cero, entonces $\cos \hat{\alpha} \approx 1$, $\sin \hat{\alpha} \approx \hat{\alpha}$, y

$$\sum_{n=0}^{\hat{N}} \hat{h}_I^2(n) \approx [\hat{N}/(2 \cdot \alpha^2)] \cdot (a-b)^2, \quad \text{si } a \neq b$$

$$\sum_{n=0}^{\hat{N}} \hat{h}_I^2(n) \approx (\hat{N}/2) a^2, \quad \text{si } a=b$$

Y si además hacemos $\hat{N}\hat{\alpha} = 2\pi$ tendremos los resultados de la tabla 3.IIIa.

Análogamente, para la función $\hat{h}_{II}^2(n)$ se tendrá:

$$\sum_{n=0}^{\hat{N}} \hat{h}_{II}^2(n) = \hat{h}_{II}^2(0) + \sum_{n=1}^{\hat{N}} \hat{h}_{II}^2(n) = a^2 + \sum_{n=1}^{\hat{N}} \hat{h}_{II}^2(n) \quad (3.65)$$

$$\sum_{n=1}^{\hat{N}} \hat{h}_{II}^2(n) = \sum_{n=1}^{\hat{N}} (1/\sin^2 \hat{\alpha}) \{a \cdot \sin[(n+1)\hat{\alpha}] - b \cdot \sin(n\hat{\alpha}) + c \cdot \sin[(n-1)\hat{\alpha}]\}^2$$

$$= [1/(2 \cdot \sin^2 \hat{\alpha})] \{ \hat{N} [a^2 + b^2 + c^2 + 2ac \cdot \cos(2\hat{\alpha}) - 2bc \cdot \cos(\hat{\alpha}) - 2ab \cdot \cos(\hat{\alpha})] + (3a^2/2) - 2a^2 \cdot \sin^2 \hat{\alpha} - [(a^2 \cdot \sin(3\hat{\alpha})) / (2 \cdot \sin(\hat{\alpha}))] \}$$

Igual que antes, si resolvemos para el caso $\hat{\alpha} \rightarrow 0$, tendremos los valores de la tabla 3.IIIb.

Estamos ya en condiciones de, mediante las tablas 3.III, calcular la expresión dada por (3.61), que por comodidad vamos a repetir:

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \sum_{k=1}^6 \sum_{m=0}^{\hat{N}} \hat{h}_{ik}^2(m) + (q^2/6) \quad (3.61)$$

Para $k=1$,

$$\begin{aligned} \sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m) &\approx (\pi/\hat{\alpha}^3) \cdot 2^{2(s1-s)} \cdot \hat{c}_{i2}^2 \cdot \hat{\beta}^2, \quad \text{si } \hat{c}_{i2} \neq 0 \\ &\approx (\pi/\hat{\alpha}) \cdot 2^{2(s1-s)} \cdot \hat{c}_{i1}^2, \quad \text{si } \hat{c}_{i2} = 0 \end{aligned} \quad (3.66)$$

Para $k=2$,

$$\begin{aligned} \sum_{m=0}^{\hat{N}} \hat{h}_{i2}^2(m) &\approx (\pi/\hat{\alpha}^3) \cdot 2^{2(s1-2 \cdot s)} \cdot \hat{c}_{i1}^2, \quad \text{si } \hat{c}_{i1} \neq 0 \\ &\approx (\pi/\hat{\alpha}) \cdot 2^{2(s1-2 \cdot s)} \cdot \hat{c}_{i2}^2, \quad \text{si } \hat{c}_{i1} = 0 \end{aligned} \quad (3.67)$$

Para $k=3$

$$\begin{aligned} \sum_{m=0}^{\hat{N}} \hat{h}_{i3}^2(m) &\approx 2^2 \cdot s1 \cdot \hat{c}_{i1}^2 + (\pi/\hat{\alpha}) \cdot \hat{\beta}^2 \cdot \hat{c}_{i2}^2 \cdot 2^2 \cdot s1, \quad \text{si } \hat{c}_{i2} \neq 0 \\ &\approx 2^2 \cdot s1 \cdot \hat{c}_{i1}^2 + \pi \hat{\alpha} \cdot 2^2 \cdot s1 \cdot \hat{c}_{i1}^2, \quad \text{si } \hat{c}_{i2} = 0 \end{aligned} \quad (3.68)$$

Para k=4,

$$\begin{aligned} \sum_{m=0}^{\hat{N}} \hat{h}_{i4}^2(m) &\approx 2^2 \cdot s^2 \cdot \hat{c}_{i2}^2 + (\pi/\hat{\alpha}) \cdot (\hat{B}\hat{c}_{i2} - \hat{c}_{i1})^2 \cdot 2^2 \cdot s^2, \quad \hat{B}\hat{c}_{i2} - \hat{c}_{i1} \neq 0 \\ &\approx 2^2 \cdot s^2 \cdot \hat{c}_{i2}^2 + \pi \hat{\alpha} \cdot 2^2 \cdot s^2 \cdot \hat{c}_{i2}^2, \quad \hat{B}\hat{c}_{i2} - \hat{c}_{i1} = 0 \end{aligned} \quad (3.69)$$

Para k=5,

$$\begin{aligned} \sum_{m=0}^{\hat{N}} \hat{h}_{i5}^2(m) &\approx (\pi/\hat{\alpha}^3) \cdot 2^{2(s-2)} \cdot (\hat{c}_{i1} - \hat{B}\hat{c}_{i2})^2, \quad \hat{B}\hat{c}_{i2} - \hat{c}_{i1} \neq 0 \\ &\approx (\pi/\hat{\alpha}) \cdot 2^{2(s-2)} \cdot \hat{c}_{i2}^2, \quad \hat{B}\hat{c}_{i2} - \hat{c}_{i1} = 0 \end{aligned} \quad (3.70)$$

Para k=6,

$$\sum_{m=0}^{\hat{N}} \hat{h}_{i6}^2(m) = \sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m) \quad (3.71)$$

En la Tabla 3.IV se especifican los valores de a, b, y c correspondientes a cada valor de k.

Por último, y para evaluar (3.61), no nos resta nada más que tener en cuenta que :

$$\begin{aligned}
\hat{\beta} &\approx -\hat{\alpha}^2, \\
\hat{c}_{i1} &\approx (c_1/L) \cdot (L + \hat{\beta} \cdot (i-1)) + (c_2/L) \cdot \hat{\beta} \cdot (i-1) \\
\hat{c}_{i2} &\approx (c_1/L) \cdot (i-1) + c_2
\end{aligned} \tag{3.72}$$

s1 y s2 vienen dados en (3.51).

Con todo ello, llegamos a :

$$\begin{aligned}
\hat{\sigma}_{oi}^2(\hat{N}) &\approx q^2/6 + (q^2/12) \cdot [(2\pi/\hat{\alpha}^3) \cdot 2^{2(s1-s)} \cdot \hat{c}_{i2}^2 \cdot \hat{\beta}^2 + \\
&\quad + (\pi/\hat{\alpha}^3) 2^{2(s1-2s)} \cdot \hat{c}_{i1}^2 + 2^{2s1} \hat{c}_{i1}^2 + (\pi/\hat{\alpha}) \cdot \hat{\beta}^2 \hat{c}_{i2}^2 \cdot 2^{2s1} + \\
&\quad + 2^{2s2} \cdot \hat{c}_{i2}^2 + (\pi/\hat{\alpha}) \cdot (\hat{\beta} \hat{c}_{i2} - \hat{c}_{i1})^2 \cdot 2^{2s2} + \\
&\quad + (\pi/\hat{\alpha}^3) \cdot 2^{2(s2-s)} \cdot (\hat{c}_{i1} - \hat{\beta} \hat{c}_{i2})^2]
\end{aligned} \tag{3.73}$$

En el caso particular de que $c_1=0$, $c_2=1$, desaparecen el quinto y el sexto sumando.

$$\begin{aligned}
\hat{c}_{i1} &\approx -\hat{\alpha}^2 \cdot (i-1)/L, & \hat{c}_{i2} &\approx 1 \\
s2 &= 0, & s &= s1, & 2^{-s1} &\approx \hat{\alpha}
\end{aligned} \tag{3.74}$$

y sustituyendo en (3.62) tendremos:

$$\begin{aligned}
\hat{\sigma}_{oi}^2(\hat{N}) &\approx (q^2/12) \cdot (1 - \delta_{i1}) + (q^2/12) \cdot [\pi \hat{\beta}^2 / \hat{\alpha}^3 + \pi 2^{-2s} \hat{c}_{i1}^2 / \hat{\alpha}^3 + \\
&\quad + 2^{2s} \hat{c}_{i1}^2 + \pi \hat{\beta}^2 \cdot 2^{2s} / \hat{\alpha} + 1 + \pi (\hat{\beta} - \hat{c}_{i1})^2 / \hat{\alpha}]
\end{aligned}$$

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx (q^2/12) \cdot [2 - \delta_{i1} + 2\pi\hat{\alpha} + (i-1)^2\hat{\alpha}^2/L^2 + \dots], \quad i \neq 1$$

$$\approx (q^2/12) [1 + 3\pi\hat{\alpha}], \quad i=1$$

En cualquier caso, como $\hat{\alpha} \rightarrow 0$, podemos escribir:

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx q^2/12, \quad i=1$$

$$\approx q^2/6, \quad i=2, \dots, L$$
(3.75)

Y teniendo en cuenta la definición [9] para el error cuadrático medio de redondeo de la salida escalar en un oscilador de bloques,

$$\sigma_o^2(N) \approx (1/L) \cdot \sum_{i=1}^L \hat{\sigma}_{oi}^2(\hat{N})$$
(3.76)

quedará:

$$\sigma_o^2(N) \approx (q^2/12) \cdot (2L-1)/L$$
(3.77)

con lo que se pone de manifiesto que la varianza del ruido de salida no depende de la frecuencia de oscilación.

3.6.- OTROS OSCILADORES DE BLOQUES.

Como ya se ha visto, el primer grado de libertad existente a la hora de implementar un oscilador escalar empleando una estructura de bloques se encuentra en la elección de la matriz de estado \hat{A} . Una vez fijada ésta, la elección de la matriz C nos determinará las condiciones iniciales de las variables de estado así como los factores de escalado y las secuencias de salida. En la sección 3.3 se ha propuesto una estructura determinada para \hat{A} con el ánimo de minimizar la sensibilidad. Veremos a continuación, y antes de presentar los resultados de simulación, dos estructuras más, obtenidas directamente de las estructuras escalares directa y acoplada [4], y que, como veremos, presentan la ventaja de estar autoescaladas, si bien la varianza del ruido dependerá de la frecuencia de oscilación.

Seguiremos, pues, para el análisis de cada una de ellas, un proceso similar al seguido en las secciones 3.3, 3.4 y 3.5 para la estructura allí propuesta.

3.6.1.- OSCILADOR DIRECTO DE BLOQUES.

Este oscilador se obtendrá tomando como matrices $\hat{\underline{A}}$ y \underline{C} las siguientes:

$$\hat{\underline{A}} = \begin{bmatrix} 2 \cdot \cos(\hat{\alpha}) & -1 \\ 1 & 0 \end{bmatrix} \quad \underline{C} = [2\cos(\hat{\alpha}) \quad -1] \quad (3.78)$$

A partir de dichas matrices y a través de (3.20) pueden obtenerse fácilmente los coeficientes \hat{c}_{ij} característicos de la estructura.

$$\begin{aligned} \hat{c}_{i1} &= [2 \cdot \cos(L\alpha) / \text{sen}(L\alpha)] \{ \text{sen}[(L-i+1)\alpha] + \\ &\quad + 2 \cdot \cos(L\alpha) \cdot \text{sen}[(i-1)\alpha] \} - \{ \text{sen}[(i-1)\alpha] / \text{sen}(L\alpha) \} \\ \hat{c}_{i2} &= [-2 \cdot \cos(L\alpha) / \text{sen}(L\alpha)] \{ \text{sen}[(i-1)\alpha] \} - \{ \text{sen}[(L-i+1)\alpha] / \text{sen}(L\alpha) \} \\ &\quad i=1, \dots, L \quad (3.79) \end{aligned}$$

La Figura III.7 nos muestra la forma general de dicha estructura.

Y las condiciones iniciales de las variables de estado se tendrán sin más que sustituir los valores dados en (3.78) en las ecuaciones (3.36) :

$$\begin{aligned}\hat{x}_1(0) &= -\text{sen}(L\alpha) \\ \hat{x}_2(0) &= -\text{sen}(2L\alpha)\end{aligned}\tag{3.80}$$

ESCALADO:

Los factores de escalado vienen dados por las ecuaciones presentadas en (3.28) y teniendo en cuenta (3.78) y (3.80), se comprueba que esta estructura está autoescalada, es decir,

$$A[\hat{x}_1(k)] = A[\hat{x}_2(k)] = 1\tag{3.81}$$

RUIDO:

En nuestro estudio de la estructura nos queda considerar los efectos de la cuantización de los productos sobre la secuencia de salida.

El modelo de ruido para el oscilador de la Figura III.7 viene dado en la Figura III.8., en la cual :

$$\sigma_{oe_0}^2 = q^2/6$$

$$\sigma_{oe_1}^2 = q^2/12$$

Las ecuaciones de estado para dicha estructura son :

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 2 \cos(\hat{\alpha}) & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_0(k) \end{bmatrix}$$

$$\hat{y}_i(k) = \begin{bmatrix} \hat{c}_{i1} & \hat{c}_{i2} \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_0(k) \end{bmatrix}$$

$i=2, \dots, L$

$$\hat{y}_1(k) = \begin{bmatrix} 2 \cos(\hat{\alpha}) & -1 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_0(k) \end{bmatrix}$$

(3.82)

Suponiendo condiciones iniciales nulas y tomando transformadas a ambos lados de las ecuaciones (3.82) podemos tener las expresiones para los elementos $\hat{G}_{ik}(z)$ de la matriz de transferencia $\hat{G}(z)$.

$\hat{G}_{ik}(z)$ representa la función de transferencia de la fuente de error k a la salida i -ésima del oscilador .

$$\begin{aligned}
\hat{G}_i(z) &= \hat{C}_{ri} \cdot (zI - \hat{A}_r)^{-1} \cdot \hat{B}_r + \hat{D}_{ri} \\
\hat{G}_{i1}(z) &= (\hat{c}_{i1}z + \hat{c}_{i2})/D(z), & i=2, \dots, L \\
\hat{G}_{i0}(z) &= 1 \\
\hat{G}_{11}(z) &= z^2/D(z) & (3.83) \\
\hat{G}_{10}(z) &= 0
\end{aligned}$$

donde $D(z) = z^2 - 2z \cdot \cos(\hat{\alpha}) + 1$

Suponiendo $(2+B)$ bits para la señal en cualquier punto del oscilador, el error cuadrático medio de la salida i -ésima del oscilador en un ciclo viene dado por :

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \cdot \sum_{m=0}^{\hat{N}} h_{i1}^2(m) + (q^2/6) \cdot \sum_{m=0}^{\hat{N}} \hat{h}_{i0}^2(m)$$

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \cdot [2 + \sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m)] \quad i=2, \dots, L$$

$$\hat{\sigma}_{o1}^2(\hat{N}) = (q^2/12) \cdot (\sum_{m=0}^{\hat{N}} \hat{h}_{11}^2(m))$$

Y reagrupando:

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/12) \cdot [2 - 2\delta_{i1} + \sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m)] \quad i=1, \dots, L \quad (3.84)$$

Y por último, el error cuadrático medio de redondeo de la salida escalar en el oscilador se tendrá sustituyendo (3.84) en (3.76) :

$$\sigma_o^2(N) = (1/L) \cdot \left(\sum_{i=1}^L \hat{\sigma}_{oi}^2(\hat{N}) \right) \quad (3.76)$$

Para el cálculo de las $\hat{h}_{i1}(m)$, así como de $\sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m)$ tendremos en cuenta las tablas 3.I, 3.II, y 3.III, con lo que podremos escribir:

$$\sum_{m=0}^{\hat{N}} \hat{h}_{11}^2(m) \approx 1 + (\pi/\hat{\alpha}^3)$$

$$\sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m) \approx (\pi/\hat{\alpha}^3) \cdot (\hat{c}_{i1} + \hat{c}_{i2})^2, \quad \hat{c}_{i1} \neq -\hat{c}_{i2} \quad i=2, \dots, L$$

con lo que:

$$\hat{\sigma}_{o1}^2(N) \approx (q^2/12) \cdot [1 + \pi/\hat{\alpha}^3] \quad (3.85)$$

$$\hat{\sigma}_{oi}^2(N) \approx (q^2/12) \cdot [2 + (\pi/\hat{\alpha}^3) \cdot (\hat{c}_{i1} + \hat{c}_{i2})^2], \quad i=2, \dots, L$$

Y teniendo en cuenta las ecuaciones (3.79) para el caso $\hat{\alpha} \rightarrow 0$,

$$\hat{c}_{i1} \approx (2L+i-1)/L \quad (3.86)$$

$$\hat{c}_{i2} \approx (-L-i+1)/L$$

Luego :

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx (q^2/12) \cdot [2 - \delta_{i1} + \pi/\hat{\alpha}^3] \approx (q^2/12) \cdot (\pi/\hat{\alpha}^3) \quad i=1, \dots, L \quad (3.87)$$

Poniéndose de manifiesto, la dependencia de $\hat{\sigma}_{oi}$ con la frecuencia. Y sustituyendo en (3.76),

$$\sigma_o^2 \approx (1/L) \sum_{i=1}^L (\pi \cdot q^2) / (12 \cdot \hat{\alpha}^3) = (\pi \cdot q^2) / (12 \cdot \hat{\alpha}^3) \quad (3.88)$$

3.6.2.- OSCILADOR ACOPLADO DE BLOQUES.

Las matrices características, $\hat{\underline{A}}$ y \underline{C} , para el oscilador acoplado de bloques que se representa en la Figura III.9 , vienen dadas por :

$$\hat{\underline{A}} = \begin{bmatrix} \cos(\hat{\alpha}) & \text{sen}(\hat{\alpha}) \\ -\text{sen}(\hat{\alpha}) & \cos(\hat{\alpha}) \end{bmatrix} \quad \underline{C} = [\cos(\hat{\alpha}) \quad \text{sen}(\hat{\alpha})] \quad (3.89)$$

De forma similar al apartado anterior, de las ecuaciones (3.20) y para (3.89) se tendrán los coeficientes \hat{C}_{ij} característicos de la estructura:

$$\begin{aligned} \hat{c}_{i1} &= [\cos(L\alpha)/\text{sen}(L\alpha)]\{\text{sen}[(L-i+1)\alpha] + \\ &\quad + \cos(L\alpha)\cdot\text{sen}[(i-1)\alpha]\} - \text{sen}(L\alpha)\cdot\text{sen}[(i-1)\alpha] \\ \hat{c}_{i2} &= \cos(L\alpha)\cdot\text{sen}[(i-1)\alpha] + \text{sen}[(L-i+1)\alpha] + \\ &\quad + \cos(L\alpha)\cdot\text{sen}[(i-1)\alpha] \end{aligned}$$

$i=1, \dots, L$ (3.90)

Y las condiciones iniciales de serán, de (3.36) :

$$\begin{aligned} \hat{x}_1(0) &= -\text{sen}(L\alpha) \\ \hat{x}_2(0) &= \cos(L\alpha) \end{aligned}$$

(3.91)

ESCALADO :

Los factores de escalado se obtienen sustituyendo (3.89) y (3.91) en (3.28) con lo que comprobamos que la estructura está autoescalada:

$$A[\hat{x}_1(k)] = A[\hat{x}_2(k)] = 1 \quad (3.92)$$

RUIDO :

En este caso, el modelo de ruido para la estructura viene dado en la Figura III.10, donde

$$\sigma_{oe_0}^2 = q^2/6 \quad , \quad \sigma_{oe_i}^2 = q^2/6$$

Y las ecuaciones de estado para esta estructura son:

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} \cos(\hat{\alpha}) & \text{sen}(\hat{\alpha}) \\ -\text{sen}(\hat{\alpha}) & \cos(\hat{\alpha}) \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_0(k) \end{bmatrix}$$

$$\hat{y}_i(k) = [\hat{c}_{i1} \quad \hat{c}_{i2}] \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + [0 \quad 0 \quad 1] \cdot \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_0(k) \end{bmatrix}$$

$i=2, \dots, L$

$$\hat{y}_1(k) = [\cos(\hat{\alpha}) \quad \text{sen}(\hat{\alpha})] \cdot \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + [1 \quad 0 \quad 0] \cdot \begin{bmatrix} e_0(k) \\ e_2(k) \\ e_0(k) \end{bmatrix}$$

(3.93)

Suponiendo condiciones iniciales nulas y tomando transformadas a ambos lados de las ecuaciones (3.93) podemos obtener la función de transferencia $\hat{G}_{ik}(z)$ de la fuente de error k a la salida i -ésima del oscilador.

$$\hat{G}_i(z) = \hat{C}_{ri} \cdot (zI - \hat{A}_r)^{-1} \cdot \hat{E}_r + \hat{D}_{ri}$$

$$\begin{aligned} \hat{G}_{i1}(z) &= (1/D(z)) \cdot [z \cdot \hat{c}_{i1} - \hat{c}_{i1} \cdot \cos(\hat{\alpha}) - \hat{c}_{i2} \cdot \text{sen}(\hat{\alpha})] \\ \hat{G}_{i2}(z) &= (1/D(z)) \cdot [z \cdot \hat{c}_{i2} - \hat{c}_{i2} \cdot \cos(\hat{\alpha}) + \hat{c}_{i1} \cdot \text{sen}(\hat{\alpha})] \quad i=2, \dots, L \\ \hat{G}_{i0}(z) &= 1 \end{aligned} \tag{3.94}$$

$$\hat{G}_{11}(z) = [z^2 - z \cdot \cos(\hat{\alpha})] / D(z)$$

$$\hat{G}_{12}(z) = [z \cdot \text{sen}(\hat{\alpha})] / D(z)$$

$$\hat{G}_{10}(z) = 0$$

donde, $D(z) = z^2 - 2z \cos(\hat{\alpha}) + 1$

Suponiendo $(2+B)$ bits para la señal en cualquier punto del oscilador, el error cuadrático medio de la salida $-i$ del oscilador en un ciclo viene dado por:

$$\hat{\sigma}_{oi}^2(\hat{N}) = (q^2/6) [1 - \delta_{i1} + \sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m) + \sum_{m=0}^{\hat{N}} \hat{h}_{i2}^2(m)] \quad i=1, \dots, L \tag{3.95}$$

y el error cuadrático medio de redondeo de la salida escalar en el oscilador vendrá dado por (3.76).

Calculamos $\sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m)$ y $\sum_{m=0}^{\hat{N}} \hat{h}_{i2}^2(m)$ teniendo

en cuenta las tablas 3.I, 3.II, y 3.III. Así,

$$\sum_{m=0}^{\hat{N}} \hat{h}_{11}^2(m) \approx 1 + (\pi/\hat{\alpha})$$

$$\sum_{m=0}^{\hat{N}} \hat{h}_{12}^2(m) \approx \pi/\hat{\alpha}$$

$$\sum_{m=0}^{\hat{N}} \hat{h}_{i1}^2(m) \approx (\pi/\hat{\alpha}^3) \cdot [(\hat{c}_{i1} \hat{\alpha}^2)/2 - \hat{c}_{i2} \hat{\alpha}]^2 \approx (\pi/\hat{\alpha}) \cdot \hat{c}_{i2}^2$$

$$\sum_{m=0}^{\hat{N}} \hat{h}_{i2}^2(m) \approx (\pi/\hat{\alpha}) \cdot \hat{c}_{i1}^2$$

con lo que :

$$\hat{\sigma}_{01}^2(\hat{N}) \approx (q^2/6) \cdot [1 + (2\pi/\hat{\alpha})] \approx (q^2/6) \hat{N} \quad (3.96)$$

$$\hat{\sigma}_{0i}^2(\hat{N}) \approx (q^2/6) \cdot [1 + (\pi/\hat{\alpha}) \cdot (\hat{c}_{i1}^2 + \hat{c}_{i2}^2)] \quad i=2, \dots, L$$

Y si tenemos en cuenta que $\hat{\alpha} \rightarrow 0$, las ecuaciones (3.90) se transforman en :

$$\hat{c}_{i1} \approx 1 - [(i-1) \hat{\alpha}^2]/L \quad (3.97)$$

$$\hat{c}_{i2} \approx \hat{\alpha} \cdot [(L+i-1)/L]$$

Y sustituyendo en (3.96), despreciando términos de segundo orden, se tiene:

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx (q^2/6) \cdot [1 + (\pi/\hat{\alpha}) \cdot \{1 - ((2(i-1)\hat{\alpha}^2)/L) + (\hat{\alpha}(L+i-1)/L)^2\}]$$

$$i=2, \dots, L \quad (3.98)$$

y volviendo a despreciar términos de segundo orden:

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx (q^2/6) \cdot [1 + (\pi/\hat{\alpha})] \approx (q^2/12) \cdot \hat{N} \quad i=2, \dots, L$$

con lo que:

$$\hat{\sigma}_{oi}^2(\hat{N}) \approx (q^2/12) \cdot \hat{N} \cdot [1 + \delta_{i1}] \quad i=1, \dots, L$$

$$(3.99)$$

expresión que pone de manifiesto la dependencia de $\hat{\sigma}_{oi}$ con la frecuencia. Si sustituimos (3.99) en (3.76) tendremos:

$$\sigma_o^2(N) \approx (1/L) \cdot [(\hat{N} \cdot (L-1)q^2)/12 + (q^2/6) \cdot \hat{N}] =$$

$$\approx (q^2/12) \cdot \hat{N} \cdot [(L+1)/L] \quad (3.100)$$

3.7.- RESULTADOS EXPERIMENTALES DE SIMULACION.

Con el objetivo de ilustrar los resultados teóricos anteriormente expuestos se han empleado las tres estructuras propuestas para diseñar una serie de osciladores digitales.

Se han realizado programas de simulación de las mismas sobre una HP-9826, empleando aritmética de punto fijo para la representación de los coeficientes. Asimismo, los números se han expresado en complemento de 2 y la cuantización se ha realizado por redondeo.

Para cada estructura se han simulado dos osciladores, ambos con coeficientes ideales, con la intención de poner únicamente de manifiesto los efectos de cuantización de las operaciones producto sobre la salida. En uno de dichos osciladores las operaciones aritméticas se realizan con la máxima precisión dada por el ordenador, y en el otro los resultados de las operacio-

nes se someten a redondeo a un número fijo de bits. La diferencia entre las secuencias de salida de cada uno de ellos nos da la secuencia de error, cuyo valor cuadrático medio en un ciclo será una aproximación del error cuadrático medio de redondeo de la salida del oscilador, que hemos determinado teóricamente.

Las Tablas 3.V, 3.VI, y 3.VII muestran los resultados obtenidos para distintas longitudes de palabra en los tres casos, para la misma frecuencia y el mismo orden de la estructura de bloques.

Las Tablas 3.VIII, 3.IX, y 3.X, muestran los resultados para distintas frecuencias y valores del orden de la estructura de bloques y ponen de manifiesto cómo para la primera estructura el error cuadrático medio de la secuencia de error no depende de la frecuencia de oscilación, mientras que para las otras estructuras sí depende.

3.8.- COMPARACION CON EL OSCILADOR ESCALAR.

Nos proponemos ahora, comparar el oscilador de bloques propuesto con el correspondiente escalar [7].

3.8.I. EFICIENCIA COMPUTACIONAL.

Uno de los criterios que se escogen para evaluar la carga computacional de las estructuras digitales es el número de operaciones aritméticas necesarias por muestra de salida.

Así pues, emplearemos éstas, es decir, el número de multiplicaciones, (m), el número de operaciones de "shiftado", (s), y el número de sumas, (a), por muestra de salida, para comparar el oscilador de bloques con el oscilador escalar.

Para el oscilador de bloques se tiene fácilmente de la Figura III.5a, con $\hat{c}_{i2}=1$,

$$m_L = 1 \quad (3.101a)$$

$$s_L = 3/L \quad (3.101b)$$

$$a_L = (2+L)/L \quad (3.101c)$$

y para el oscilador escalar :

$$m_1 = 1 \quad (3.102a)$$

$$s_1 = 3 \quad (3.102b)$$

$$a_1 = 3 \quad (3.102c)$$

Comparando las expresiones (3.101) con las (3.102) puede verse en primer lugar, que el número de multiplicaciones no depende de la longitud de la estructura de bloques, lo que implica que el número de multiplicaciones a realizar es el mínimo posible tanto para los osciladores de bloque como para el escalar, con lo que no es posible mejorar este aspecto con la estructura de bloques.

Por otro lado, el número de operaciones de "shiftado" decrece al aumentar L, como se deduce observando (3.101c) y (3.102c).

Así pues, si se considera que la multiplicación es la operación más compleja en el sistema digital, la realización por bloques del oscilador no aporta grandes beneficios. No obstante, ha sido señalado por varios autores [18-20], que el motivo primordial del uso de las estructuras de bloques para la realización de sistemas digitales es el elevado incremento de la velocidad del flujo de datos que se consigue si se dispone de facilidades de procesado en paralelo. En este caso, el oscilador de bloques permitirá una mayor rapidez en la obtención de las muestras de salida que el correspondiente escalar.

3.8.II. TIEMPO DE COMPUTACION DE LAS MUESTRAS DE SALIDA.

Usaremos ahora el concepto de "precedencia" [21] para analizar la relación entre el número de facilidades de procesado empleado en la realización de cada una de las estructuras (de bloques y escalar) y el tiempo mínimo de computación para cada muestra de salida en cada una de ellas.

Los diagramas de flujo de señal para la computación de cada muestra de salida para cada oscilador y sus correspondientes "formas de precedencia" se muestran en la Figura III.11.

La "forma de precedencia" indica en primer lugar el orden en el que deben evaluarse los nodos y en segundo lugar , el número de operaciones aritméticas realizadas concurrentemente en la estructura.

Como es sabido, las operaciones que intervienen en la realización de los osciladores, tanto escalar como de bloques, son: multiplicaciones, sumas y "shiftados".

Una operación de "shiftado" implica en cierto modo un redondeo, por tanto, el tiempo necesario para realizarla (t_s) se puede considerar igual que el necesario para realizar una suma (t_a).

Por otro lado, el tiempo necesario para realizar un producto (t_m) depende del método empleado para implementar el multiplicador [22]. Consideraremos, en lo que sigue, que $t_m \geq 2t_a$ con la intención de cubrir el mayor número de casos posible.

Para el oscilador escalar, se obtiene fácilmente de la Figura III.11(b) que el tiempo necesario para obtener una muestra de salida es:

$$t = t_m + 2t_s + t_a \quad (3.103)$$

siempre que se supongan disponibles un "shiftador", un multiplicador y un sumador. Obsérvese que es imposible disminuir t aumentando el número de facilidades en el procesado.

Para el oscilador de bloques sea τ el tiempo

necesario para la obtención de una muestra en cada una de las L salidas del mismo. Como \hat{Y}_1 ya se ha obtenido, necesitaremos las $(L-1)$ salidas restantes $\hat{Y}_2, \hat{Y}_3, \dots, \hat{Y}_L$. Si suponemos que se dispone de M multiplicadores y M sumadores que trabajan en paralelo, además de las facilidades del oscilador escalar, podemos escribir :

$$\tau = \lceil L/(M+1) \rceil \cdot t_m \quad (3.104)$$

Normalmente son los multiplicadores los elementos más caros a la hora de su implementación en un sistema digital. Nótese que si $\lceil L/(M+1) \rceil$ es un entero los multiplicadores se emplean al máximo, en otro caso los multiplicadores no se usan con una eficiencia del 100%, es decir, existen multiplicadores durante la obtención de las muestras de salida que no se encuentran operando. Esto se muestra en la Figura III.12 donde se supone $L=12, t_m=4t_a$. En dicha Figura se han tomado dos valores de M , $M=3$, en la Figura III.12(a) y $M=4$, en la Figura III.12(b).

Teniendo en cuenta que el tiempo necesario para obtener el nuevo estado en el oscilador está dado por (3.103), el tiempo necesario para la obtención de una muestra en el oscilador de bloques será:

$$t_L = (1/L) \text{ máx} \{ \tau, t_m + 2t_s + t_a \} \quad (3.105)$$

Hay que hacer notar, que si las facilidades de procesamiento para el oscilador de bloques son las mismas que las del oscilador escalar, es decir $M=0$,

$$t_L = t_m \quad (3.106)$$

que es menor que el tiempo obtenido para este último.

Si comparamos (3.103) con (3.105) y suponemos que t_m es mucho mayor que t_s y t_a , vemos que en la realización de bloques el tiempo de computación de cada muestra de salida es reducido por un factor $(M+1)$ en relación con el tiempo necesario en la realización escalar, siendo M el número de facilidades de procesado (sumadores y multiplicadores) añadidos a la realización escalar. Si t_m es del mismo orden que t_s y t_a entonces se obtiene todavía más un mejor comportamiento.

En resumen, en aquellas situaciones en las que interesa la realización de un oscilador digital de baja frecuencia comparada con la frecuencia de muestreo, el oscilador de bloques permite mediante la elección adecuada de L y M un grado adecuado de paralelismo.

3.8.III. RESULTADOS DE SIMULACION.

De las Tablas 3.V puede deducirse que el oscilador de bloques presenta un error cuadrático medio de redondeo ligeramente mayor que el escalar. De cualquier forma, este último necesita un acumulador mayor que el primero, de aproximadamente $(\log_2 L)$ bits más.

Y de la Tabla 3.VIII puede verse que el error cuadrático medio de ambos osciladores no depende de la frecuencia de oscilación, f_0 .

3.9. REFERENCIAS.

- [1]. J.Tierney, C.M.Rader and B.Gold, "A digital frequency synthesizer". IEEE Trans. Audio Electroacoust. Vol.AU-19, pp.48-56. 1971.
- [2]. D.L.Favin, "Tone generator". Bell Sys. Tech. J. Vol-60, pp.1655-1671. 1981.
- [3]. T.S.Kinsel and J.H.Whorinen, "A digital signal generation". IEEE Micro, Vol.1, pp.6-15. 1981.
- [4]. B.Gold and C.M.Rader. "Digital Processing of Signals". Mc.Graw-Hill, New York. 1969.
- [5]. D.Mitra and H.G.Montgomery, "On a digital oscillator". Proc. 1972 Annual Allerton Conf. on Circuit and Syst. Theory. pp.709-723.
- [6]. K.Furano, S.K.Mitra, K.Hirano and Y.Ito, "Design of digital sinusoidal oscillators with absolute periodicity". IEEE Trans Aerospace and Electronic Systems, Vol.AES-11, pp.1286-1299 1975.

- [7]. J.I.Acha, J.Payán-Somet and A.Civit, "Design of very-low-frequency digital oscillators". IEE Proc. Pt.G, Electronic Circuits and Systems, Vol.131, pp.93-102. 1984.
- [8]. S.K.Mitra and R. Gnanasekaran, "Block implementation of recursive digital filters: New structures and properties", IEEE Trans. Circuits and Syst. Vol.CAS-25, pp.200-207. 1978.
- [9]. C.W.Barnes and S.Shinnanka, "Finite word effects in block-state realizations of fixed-point digital filters". IEEE Trans. Circuits and Syst. Vol.CAS-27, pp.345-349. 1980.
- [10]. C.W.Barnes and S.Shinnanka, "Block-shift invariance and block implementation of discrete time filters". IEEE Trans. Circuits and Syst. Vol.CAS-27, pp.667-672. 1980.
- [11]. P.Ananthakrishna and S.K.Mitra, "A general theory of block-state digital filters". Proc. IEEE Int. Symp. Circuits and Syst., pp.1025-1028
- [12]. P. Ananthakrishna and S.K.Mitra, "Minimum round-off noise realization of block-state recursive digital filters". Circuits, Systems and Signal Processing, Vol.1, No.2, pp.217-231. 1982

- [13]. B.Noble and J.W.Daniel, "Applied Linear Algebra" (2nd.Edition). Prentice Hall, Englewood Cliffs. 1977.
- [14]. E.I.Jury, "Theory and Application of the z-transform Method". Robere Krieger Publishing Co., Huntington. 1973.
- [15]. J.Payán-Somet, "Análisis y diseño de osciladores Sevilla. 1982.
- [16]. R.C.Agarwal and C.S.Burrus, "New recursive digital filter structures having very low sensitivity and roundoff noise". IEEE Trans. on 1975.
- [17]. D.C.Munson and B.Liu, "ROM/ACC realisation of digital filters for poles near the unit circle". IEEE Trans. Circuits and Systems, Vol.CAS-27, pp.147-151. 1980.
- [18]. C.S.Burrus, "Block implementation of digital filters". IEEE Trans. Circuit Theory. Vol.CT-18, pp. 697-701. 1971.
- [19]. J.Zeman and A.G.Lindgren, "Fast digital filters with low round-off noise". IEEE Trans. Circuits and Syst. Vol.CAS-28, pp. 716-722. 1981.

- [20]. Ch.L.Nikias, "Fast block data processing via a new IIR digital filter structure". IEEE Trans. Acoustics, Speech and signal processing. Vol. ASSP-32, pp.770-779. 1984.
- [21]. R.E.Crochiere and A.V. Oppenheim, "Analysis of linear digital networks". Proc. IEEE, vol.63, pp.581-595. 1975.
- [22]. K.Hwang, "Computer Arithmetic". Wiley, New York. 1979.

$$\hat{G}_I(z) = \frac{az-b}{D(z)}$$

$$\hat{h}_I(n) = a \cdot \frac{\text{sen}(n\hat{\alpha})}{\text{sen}(\hat{\alpha})} - b \cdot \frac{\text{sen}[(n-1)\hat{\alpha}]}{\text{sen}(\hat{\alpha})} - b\delta(n)$$

$$\hat{G}_{II}(z) = \frac{az^2-bz+c}{D(z)}$$

$$\hat{h}_{II}(n) = a \cdot \frac{\text{sen}[(n+1)\hat{\alpha}]}{\text{sen}(\hat{\alpha})} - b \cdot \frac{\text{sen}(n\hat{\alpha})}{\text{sen}(\hat{\alpha})} + c \cdot \frac{\text{sen}[(n-1)\hat{\alpha}]}{\text{sen}(\hat{\alpha})} + c\delta(n)$$

Tabla 3.I. Pares de transformadas de interés.

$$\sum_{n=1}^N \text{sen}^2(n\alpha) = (1/4) \cdot \{ (2N+1) \cdot \text{sen}(\alpha) - \text{sen}[(2N+1)\alpha] \} \cdot \text{cosec}(\alpha)$$

$$\sum_{n=1}^N \text{sen}(n\alpha) \cdot \text{sen}[(n+1)\alpha] = (1/4) \cdot \{ (N+1) \text{sen}(2\alpha) - \text{sen}[(2N+2)\alpha] \} \cdot \text{cosec}(\alpha)$$

$$\sum_{n=1}^N \text{sen}(n\alpha) \cdot \text{sen}[(n+2)\alpha] = (N/2) \cdot \cos(2\alpha) - (1/2) \cdot \cos[(N+3)\alpha] \cdot \text{sen}(N\alpha) \cdot \text{cosec}(\alpha)$$

Tabla 3.II. Relaciones Trigonómicas.

$\sum_{n=1}^{\hat{N}} \hat{h}_I^2(n) \approx \begin{cases} (\pi/\hat{\alpha}^3) \cdot (a-b)^2 & \text{si } a \neq b \\ (\pi/\hat{\alpha}) \cdot a^2 & \text{si } a=b \end{cases}$
$\sum_{n=1}^{\hat{N}} \hat{h}_{II}^2(n) \approx \begin{cases} (\pi/\hat{\alpha}^3) \cdot (a-b+c)^2 & \text{si } a+c \neq b \\ (\pi/\hat{\alpha}) \cdot (a-c)^2 & \text{si } a+c=b, a \neq c \\ \pi a^2 \hat{\alpha} & \text{si } a=c=b/2 \end{cases}$

Tabla 3.III. Aproximaciones de $\sum_{n=1}^{\hat{N}} \hat{h}_j^2(n)$ ($j=I, II$)
para a pequeño. ($\hat{N}=N/L$ $2\pi/\hat{\alpha}$)

k	a	b	c
1	$2^{s1-s} \hat{c}_{i1}$	$2^{s1-s}(\hat{c}_{i1} - \hat{\beta}\hat{c}_{i2})$	-
2	$2^{s1-2s} (\hat{c}_{i1} + \hat{c}_{i2})$	$2^{s1-2s} \cdot \hat{c}_{i2}$	-
3	$2^{s1} \cdot \hat{c}_{i1}$	$2^{s1}[2\hat{c}_{i1} - \hat{\beta}\hat{c}_{i2}]$	$2^{s1}[\hat{c}_{i1} - \hat{\beta}\hat{c}_{i2}]$
4	$2^{s2} \cdot \hat{c}_{i2}$	$2^{s2}(2\hat{c}_{i2} + \hat{\beta}\hat{c}_{i2} - \hat{c}_{i1})$	$2^{s2}(\hat{c}_{i2} + \hat{\beta}\hat{c}_{i2} - \hat{c}_{i1})$
5	$2^{s2-s} \cdot \hat{c}_{i2}$	$2^{s2-s}[\hat{c}_{i2} + \hat{\beta}\hat{c}_{i2} - \hat{c}_{i1}]$	-

Tabla 3.IV. Valores de a,b,c, para k=1,..5.

	B+2	$E\{e^2(n)\}$	$\sigma_o^2(N)$
L=1 s=8	11	3.630E-7	3.179E-7
	12	9.770E-8	7.947E-8
	13	2.081E-8	1.987E-8
	14	5.371E-9	4.967E-9
	15	1.369E-9	1.242E-9
	16	3.441E-10	3.104E-10
L=4 s=6	11	2.161E-6	5.563E-7
	12	5.220E-7	1.391E-7
	13	1.409E-7	3.477E-8
	14	2.477E-8	8.692E-9
	15	5.880E-9	2.173E-9
	16	1.483E-9	5.433E-10

Tabla 3.V. Oscilador de bloques propuesto

Error cuadrático medio de redondeo ($\sigma_o^2(N)$) y
 Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
 frente a diferentes valores de longitud de la señal, B.
 Periodo de oscilación, $T_o=1000 T_s$.

	B+2	$E\{e^2(n)\}$	$\sigma_o^2(N)$
L=1	11	5.074	4.026
	12	7.645	1.006
	13	6.866	0.2516
	14	6.098	6.290E-2
	15	0.3418	1.573E-2
	16	0.1089	3.932E-3
L=4	11	0.3557	6.290E-2
	12	0.1087	1.573E-2
	13	3.464E-2	3.931E-3
	14	5.184E-4	9.829E-4
	15	1.339E-5	2.457E-4
	16	2.436E-5	6.143E-5

Tabla 3.VI. Oscilador de bloques directo

Error cuadrático medio de redondeo ($\sigma_o^2(N)$) y
 Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
 frente a diferentes valores de longitud de la señal, B.
 Periodo de oscilación, $T_o=1000 T_s$.

	B+2	$E\{e^2(n)\}$	$\sigma_o^2(N)$
L=1	11	3.839E-3	6.358E-4
	12	2.611E-3	1.589E-4
	13	8.492E-5	3.974E-5
	14	6.837E-5	9.934E-6
	15	6.255E-5	2.483E-6
	16	6.814E-5	6.209E-7
L=4	11	1.093E-3	9.934E-5
	12	1.206E-3	2.484E-5
	13	7.144E-5	6.209E-6
	14	8.019E-6	1.552E-6
	15	2.854E-7	3.880E-7
	16	7.435E-8	9.701E-8

Tabla 3.VII. Oscilador de bloques acoplado

Error cuadrático medio de redondeo ($\sigma_o^2(N)$) y
 Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
 frente a diferentes valores de longitud de la señal, B.
 Periodo de oscilación, $T_o=1000 T_s$.

	N	$E\{e^2(n)\}$	Long. Acc.	
L=1	1000	3.441E-10	24	$\sigma_o^2(N) _{L=1} = 3.104E-10$
	1200	4.099E-10	24	
	1600	3.667E-10	24	
	2000	3.364E-10	25	
	2500	3.251E-10	25	
	3000	3.294E-10	25	
L=4	1000	1.483E-9	22	$\sigma_o^2(N) _{L=4} = 5.433E-10$
	1200	1.514E-9	22	
	1600	1.377E-9	22	
	2000	1.483E-9	23	
	2500	1.494E-9	23	
	3000	1.636E-9	23	
L=10	1000	1.799E-9	20	$\sigma_o^2(N) _{L=10} = 5.898E-10$
	1200	2.346E-9	21	
	1600	1.978E-9	21	
	2000	2.188E-9	21	
	2500	1.742E-9	22	
	3000	1.797E-9	22	

Tabla 3.VIII. Oscilador de bloques propuesto

Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
y longitud del acumulador
frente a distintos valores de la relación T_0/T_s .
Longitud de la señal. $B+2=16$

	N	$E\{e^2(n)\}$	$\sigma_o^2(N)$
L=1	1000	0.1089	3.932E-3
	1200	3.169E-2	6.794E-3
	1600	6.196	1.610E-2
	2000	6.87	3.145E-2
	2500	6.61	6.143E-2
	3000	7.022	0.106
L=4	1000	2.436E-5	6.143E-5
	1200	1.015E-5	1.061E-4
	1600	1.180E-4	2.516E-4
	2000	6.503E-4	4.915E-4
	2500	6.423E-3	9.599E-4
	3000	7.580E-4	1.659E-3
L=10	1000	2.418E-6	3.932E-6
	1200	8.627E-6	6.794E-6
	1600	2.903E-6	1.610E-5
	2000	1.342E-6	3.145E-5
	2500	2.432E-5	6.143E-5
	3000	1.106E-5	1.061E-4

Tabla 3.IX. Oscilador de bloques directo

Error cuadrático medio de redondeo ($\sigma_o^2(N)$) y
 Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
 frente a distintos valores de la relación T_0/T_s .
 Longitud de la señal. $B+2=16$

	N	$E\{e^2(n)\}$	$\sigma_o^2(N)$
L=1	1000	6.814E-5	6.209E-7
	1200	4.535E-5	7.450E-7
	1600	2.241E-5	9.934E-7
	2000	1.641E-5	1.242E-6
	2500	1.586E-5	1.552E-6
	3000	2.964E-5	1.863E-6
L=4	1000	7.435E-8	9.701E-8
	1200	2.511E-7	1.164E-7
	1600	5.842E-7	1.552E-7
	2000	2.500E-6	1.940E-7
	2500	8.174E-6	2.425E-7
	3000	2.636E-6	2.910E-7
L=10	1000	1.602E-7	3.415E-8
	1200	1.114E-7	4.098E-8
	1600	2.084E-8	5.464E-8
	2000	5.567E-8	6.830E-8
	2500	7.437E-8	8.537E-8
	3000	2.496E-7	1.024E-7

Tabla 3.X. Oscilador de bloques acoplado

Error cuadrático medio de redondeo ($\sigma_o^2(N)$) y
 Error cuadrático medio de la secuencia de error ($E\{e^2(n)\}$)
 frente a distintos valores de la relación T_0/T_s .
 Longitud de la señal. $B+2=16$

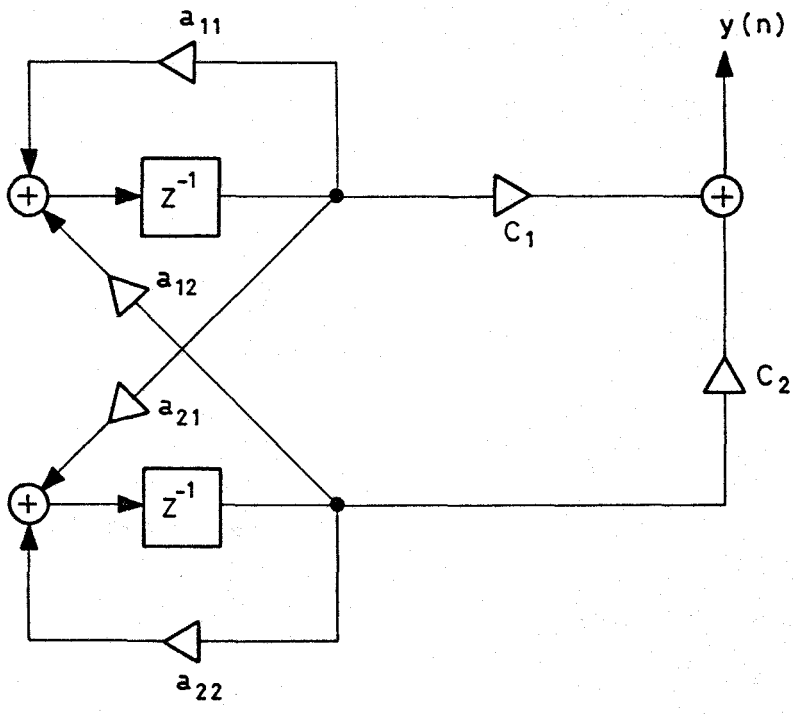


FIGURA III. 1.- OSCILADOR ESCALAR DE ESTADOS DE SEGUNDO ORDEN

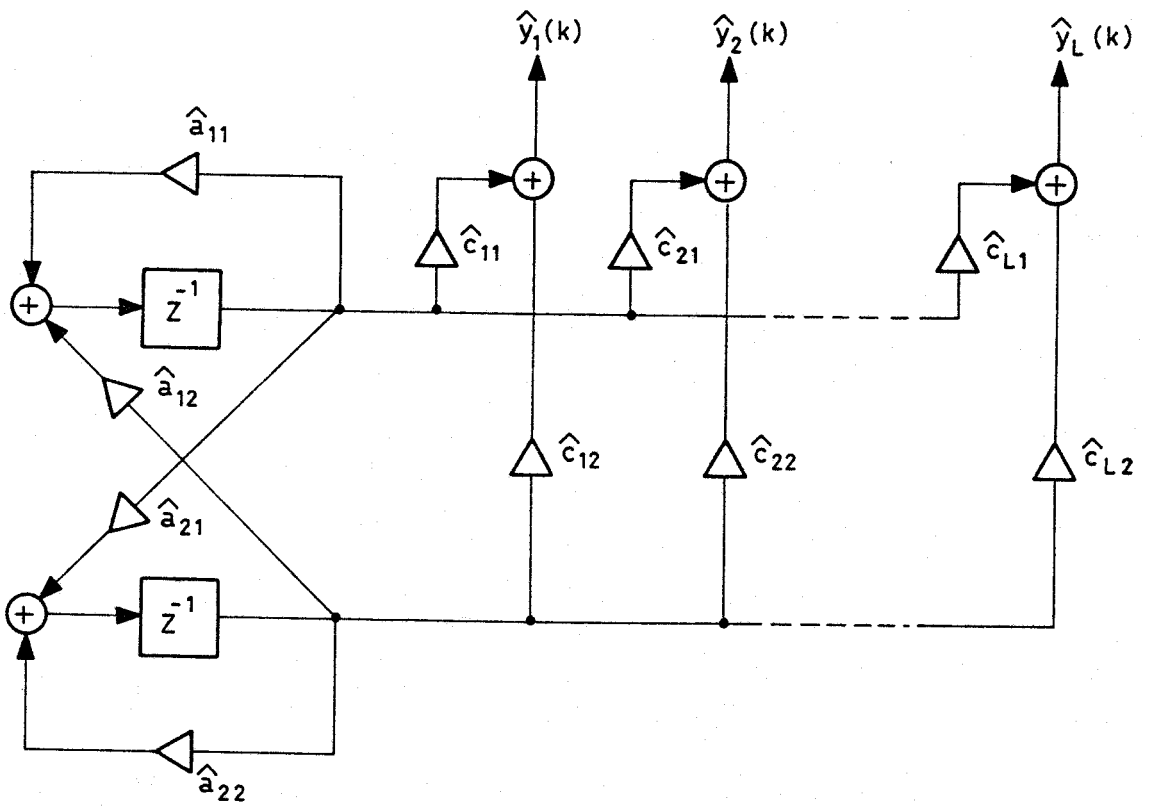


FIGURA III 2 - ESTRUCTURA GENERAL DE UN OSCILADOR EN BLOQUE DE ESTADOS

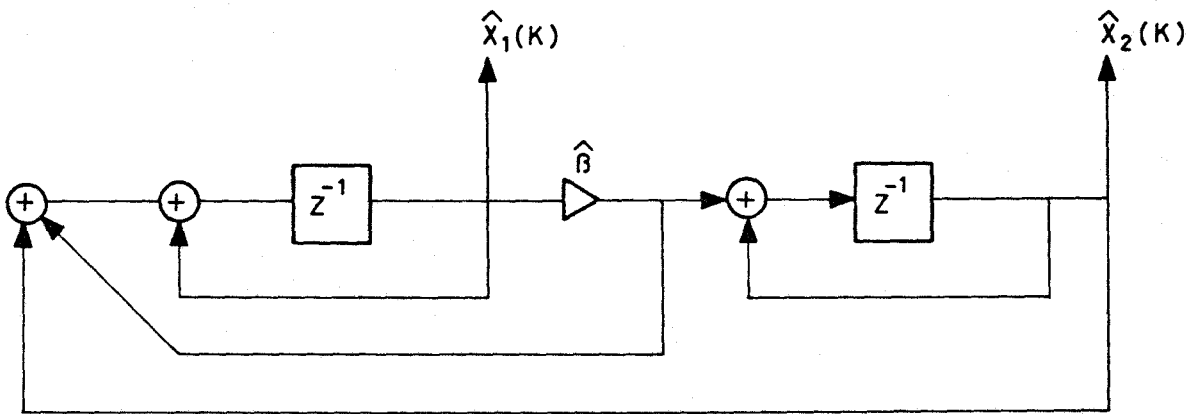


FIGURA III 3.- PARTE RECURSIVA DEL OSCILADOR DE BLOQUES PROPUESTO

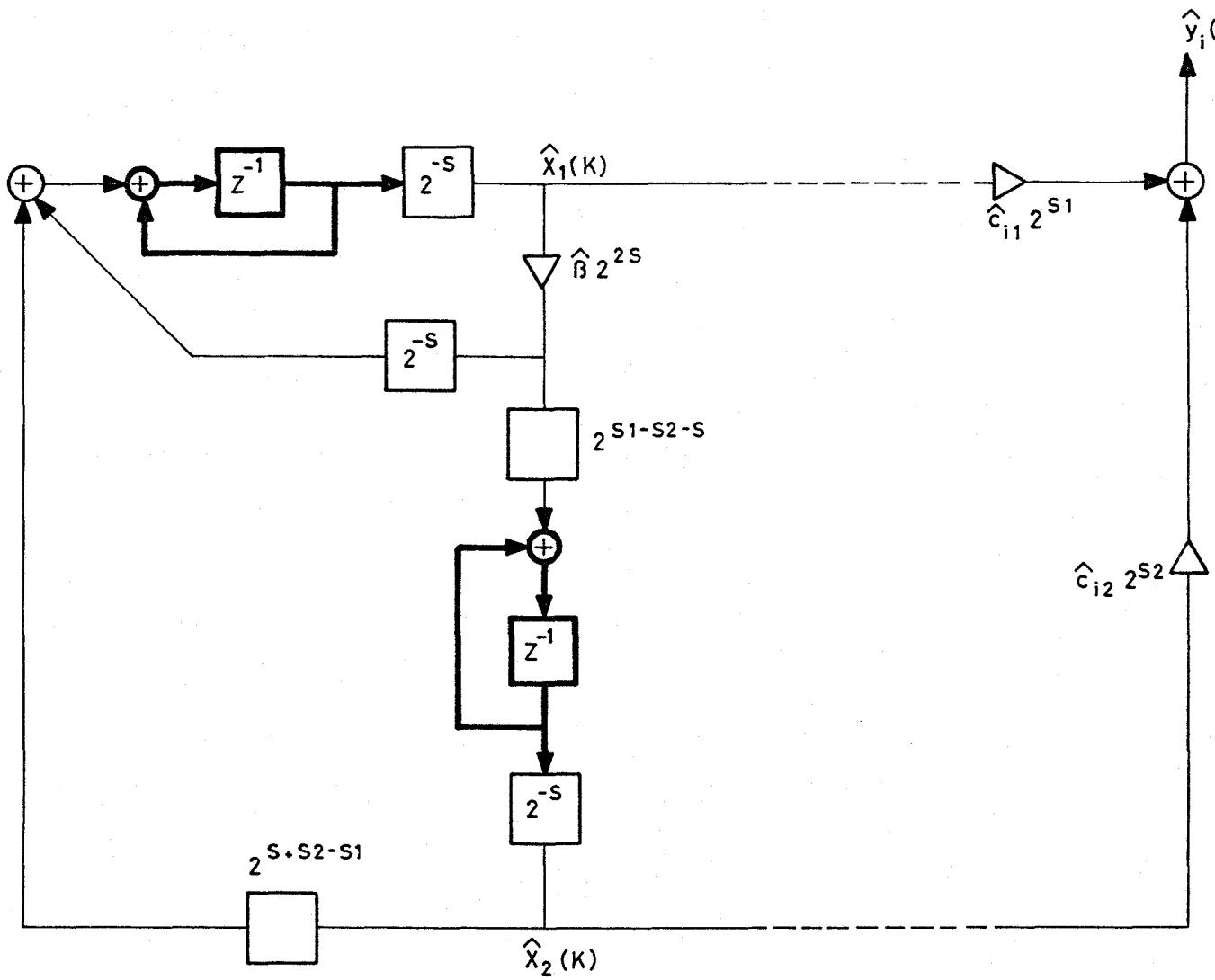


FIGURA III.4 - OSCILADOR DE BLOQUES ESCALADO

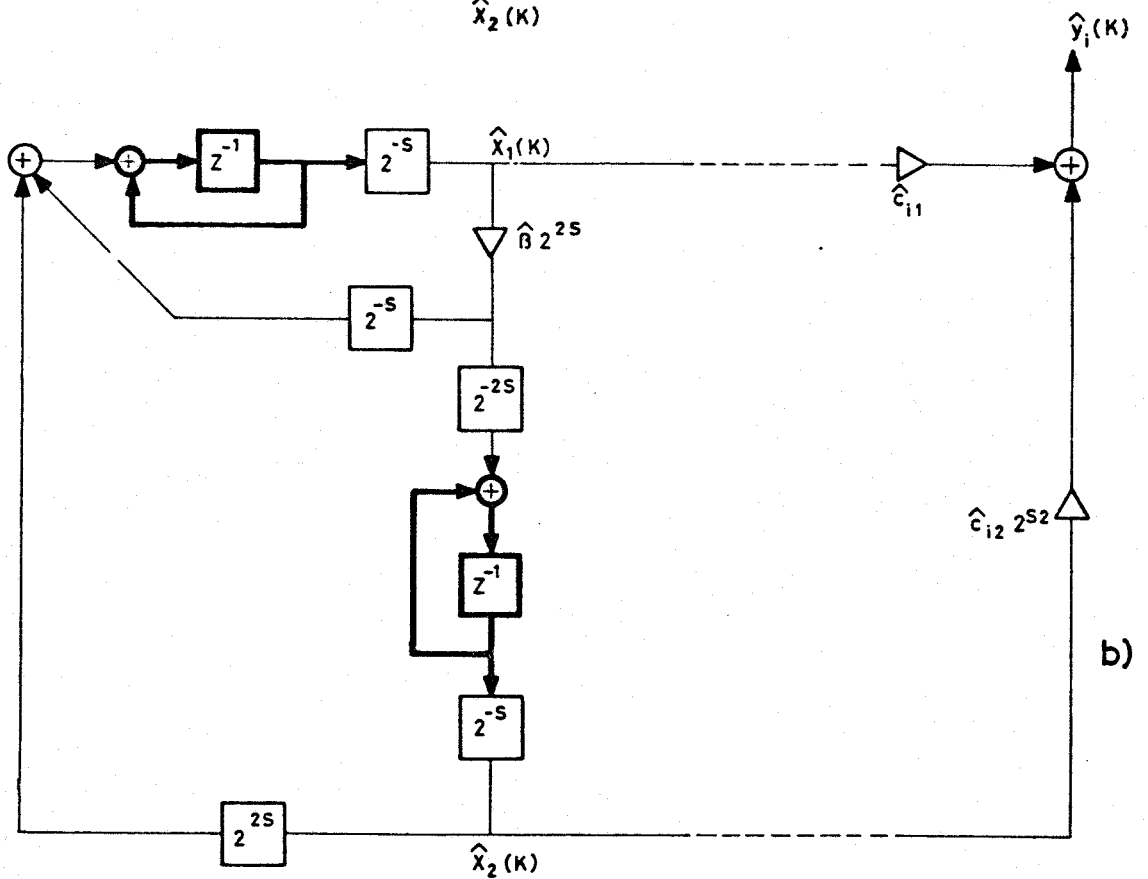
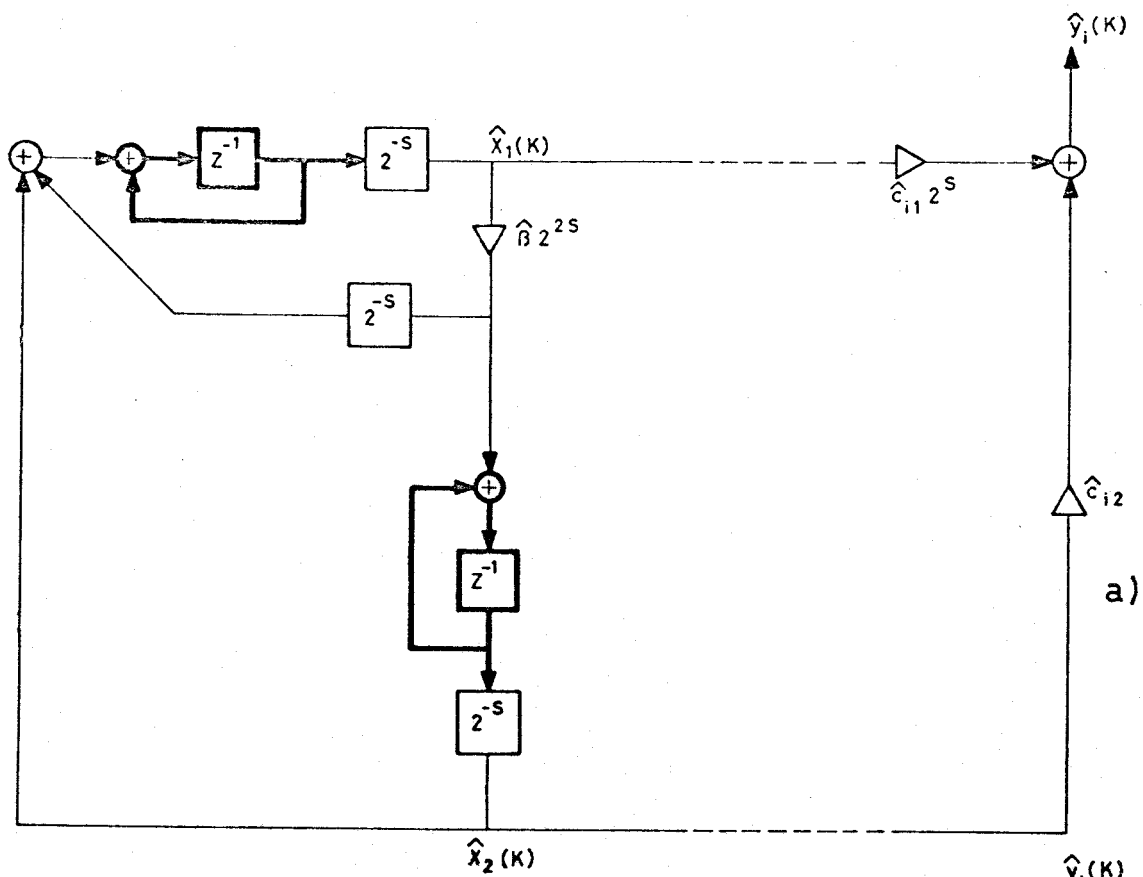


FIGURA III. 5 - CASOS PARTICULARES DE ESCALADO

- a) $S=S_1, S_2=0$
- b) $S=S_2, S_1=0$

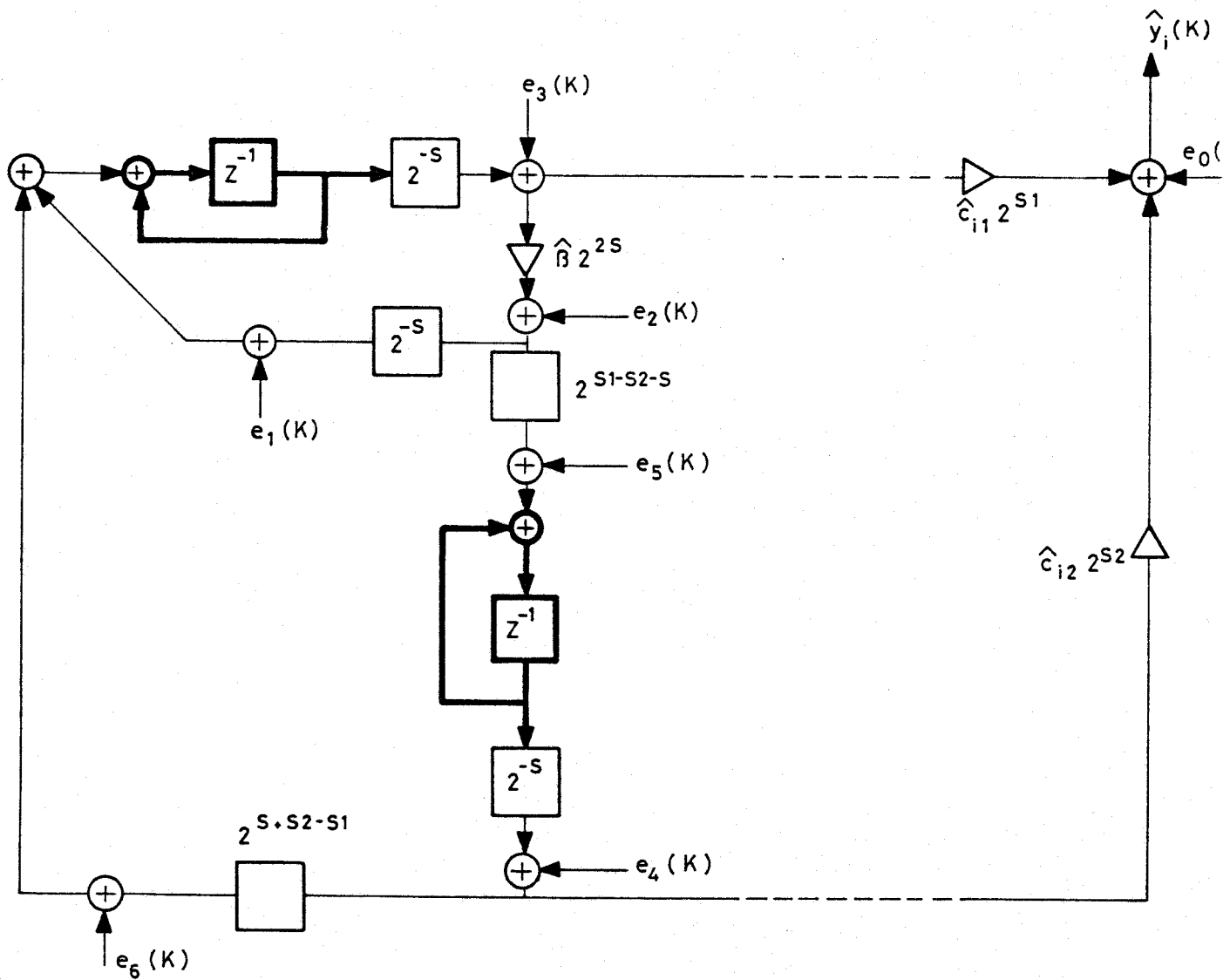


FIGURA III.6 - MODELO DE RUIDO PARA EL OSCILADOR DE BLOQUES

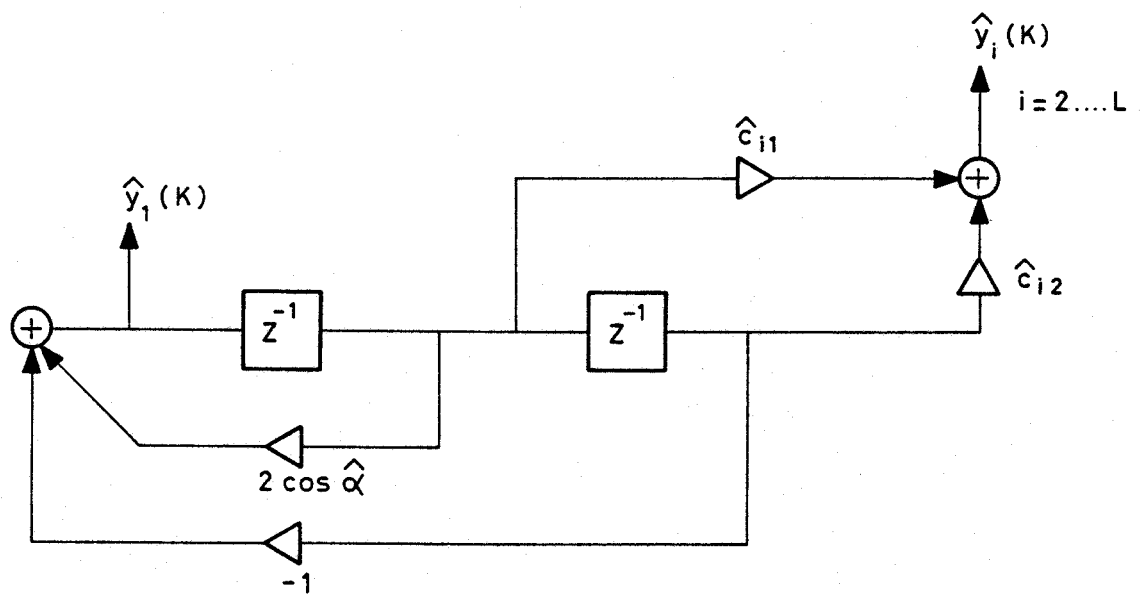


FIGURA III. 7 - OSCILADOR DIRECTO DE BLOQUES

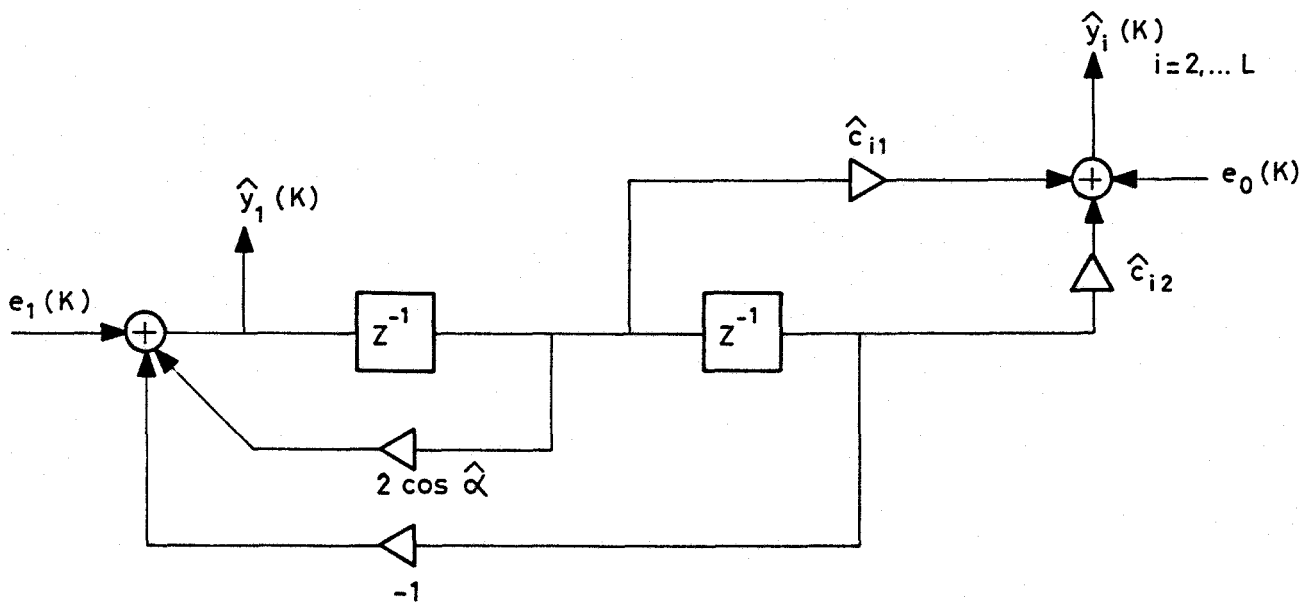


FIGURA III. 8 - MODELO DE RUIDO PARA EL OSCILADOR DIRECTO DE BLOQUES

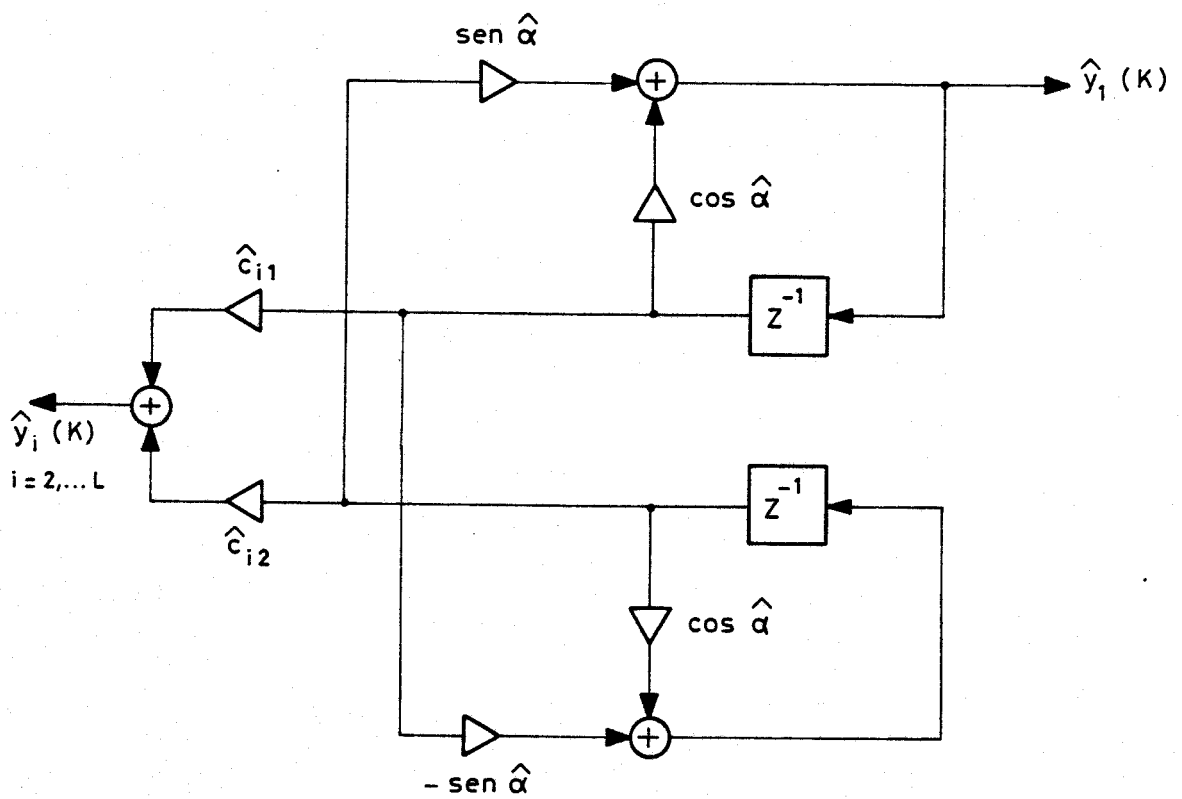


FIGURA III. 9 - OSCILADOR ACOPLADO DE BLOQUES

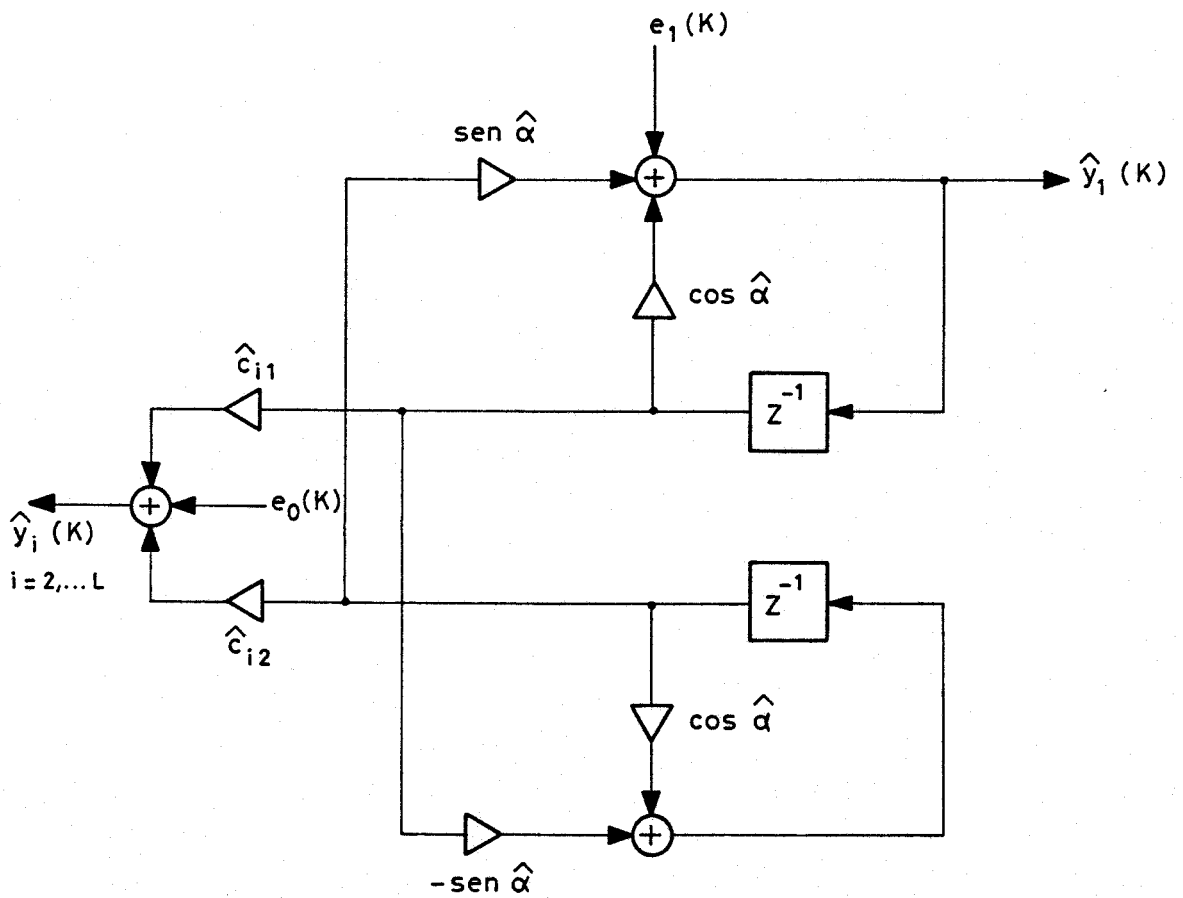


FIGURA III. 10 .- MODELO DE RUIDO PARA EL OSCILADOR ACOPLADO DE BLOQUES

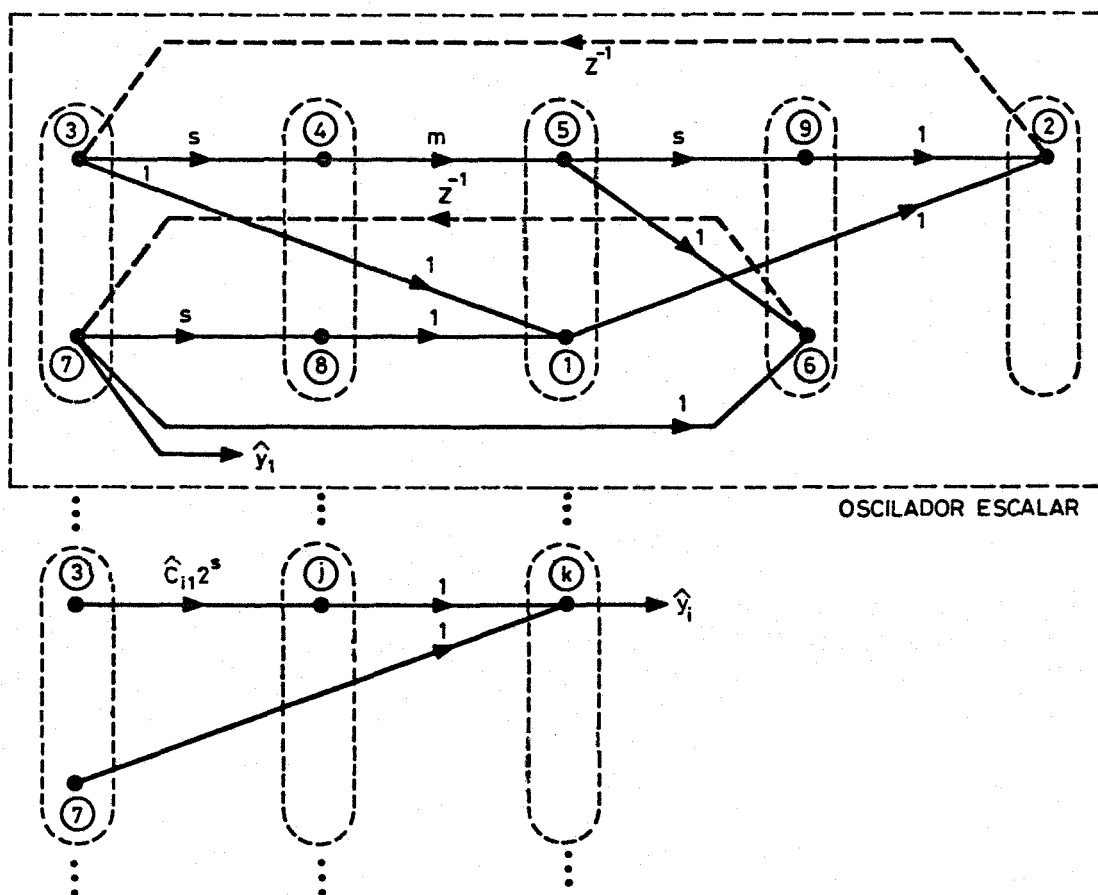
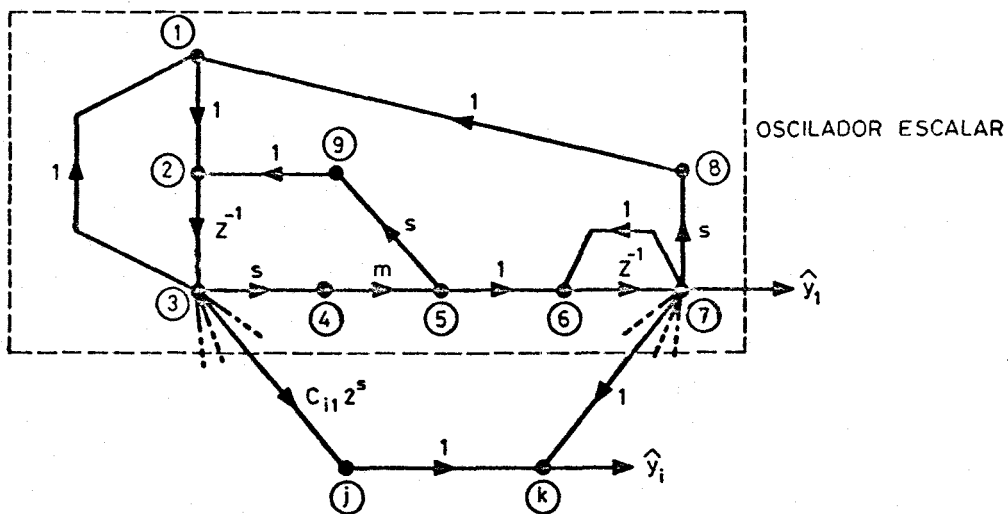
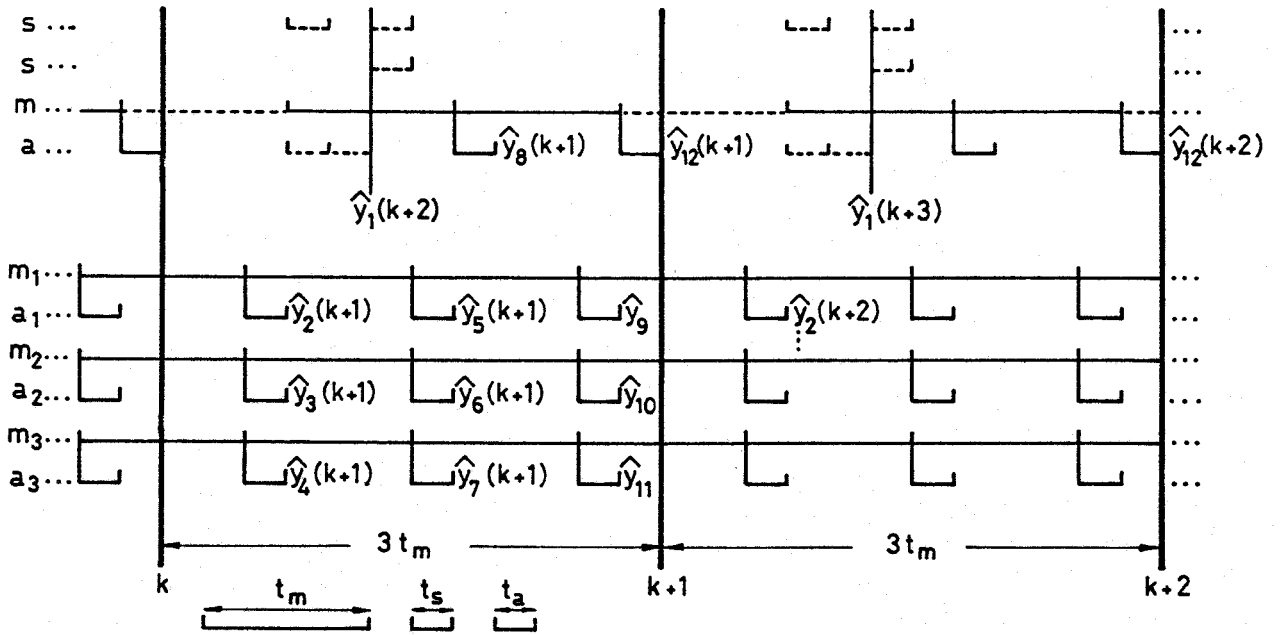
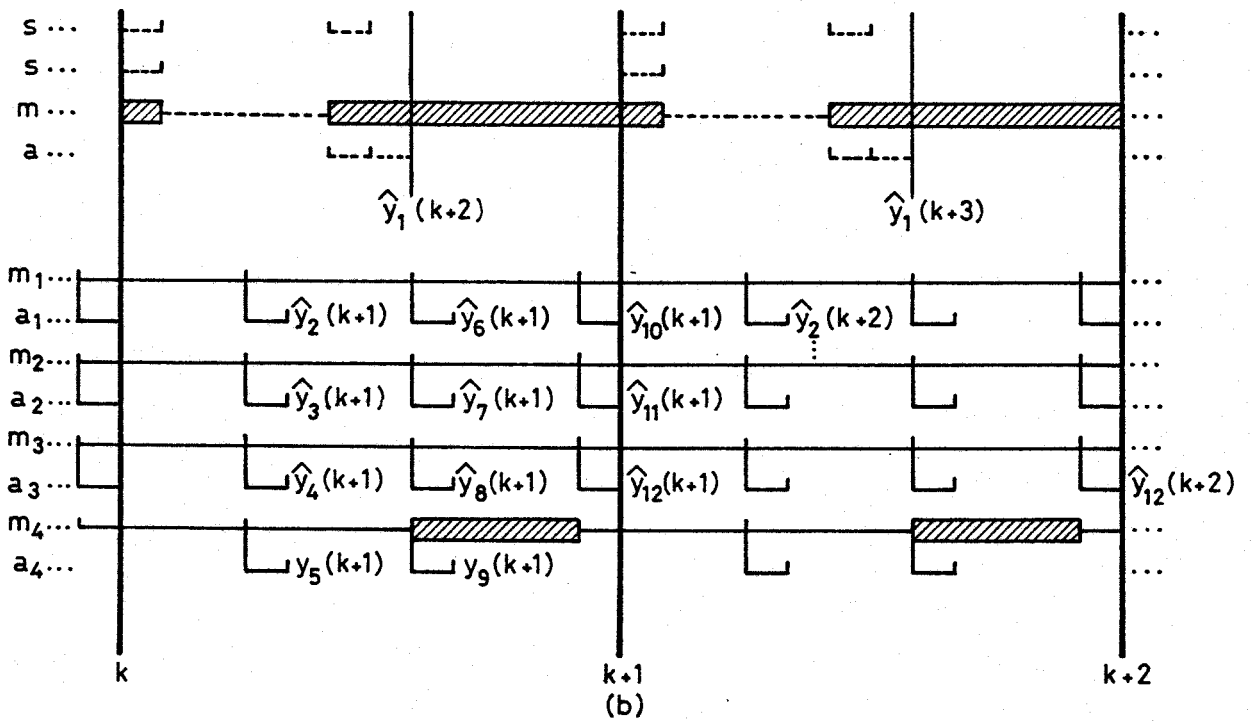


FIGURA III. 11

- a) Diagrama de flujo para los osciladores escalar y de bloques
 b) "Forma de precedencia". Para el oscilador de bloques (escalar)
 $m = \hat{\beta} 2^{2S}$ ($m = \beta 2^{2S}$), $S = 2^{-S_1}$ ($S = 2^{-S_1}$)
 nudo $j = 6 + 2i$ y nudo $k = j + 1$ ($i = 2, \dots, L$)



(a)



(b)

FIGURA III. 12 - DIAGRAMAS DE TIEMPO PARA $L=12$ Y
a) $M=3$, b) $M=4$

CAPITULO 4.

**REALIZACION SOBRE TMS32010 DE UN
OSCILADOR DE BLOQUES DE BAJA FRECUENCIA**

4.0. INTRODUCCION.

En los últimos tiempos el procesado digital de señal ha avanzado considerablemente en lo que se refiere a sus aspectos prácticos y teóricos [1-7]. Simultáneamente a la aparición de nuevos algoritmos para el procesamiento digital de señal, se desarrollan mejores herramientas para implementarlos.

Uno de los más importantes avances en la tecnología electrónica lo constituyen los procesadores digitales de señal de alta velocidad. Estos procesadores integrados en un único "chip" están disponibles comercialmente en circuitos VLSI.

Los procesadores digitales de señal son, esencialmente, microprocesadores ó microcomputadores de alta velocidad, diseñados específicamente para realizar algoritmos de procesamiento digital de señal de alto nivel de computación. Aprovechando las ventajas de una arquitectura avanzada, del procesamiento paralelo y de un conjun

to de instrucciones dedicado exclusivamente al proceso digital de la señal, estos dispositivos son capaces de ejecutar millones de operaciones de DSP(procesado digital de señal) por segundo. Con todo ello, es posible implementar algoritmos realmente complicados sobre un pequeño "chip" de silicio, cuando hasta hace poco se hacía necesario el uso de un minicomputador o de un "array processor".

Estos procesadores son especialmente atractivos, ya que al ser programables casi en alto nivel, permiten la realización de diferentes tareas con el mismo dispositivo, cambiando simplemente los programas.

Por causa de estas y otras muchas ventajas, los procesadores digitales de señal están adquiriendo una importancia relevante en áreas de propósito general de procesado digital de la señal, telecomunicaciones, tratamiento de voz, imagen, control, gráficos, instrumentación, etc.

En la Tabla 4.I. se hace un resumen de las aplicaciones más típicas de un procesador DSP.

Dentro de la gran familia de procesadores digitales de señal, centramos nuestro interés en el procesador de la familia Texas Instruments, TMS320, el TMS32010.

Después de hacer una breve descripción de la familia TMS320, describiremos, en el apartado 1, las características de manejo y funcionamiento del procesador TMS32010.

En el apartado 2, describiremos el "software" y "hardware" disponible, adicional, necesario para la comprobación experimental de los diseños realizados sobre dicho procesador.

En el apartado 3, se presenta el método general de implementación de un determinado algoritmo sobre el procesador, centrándonos en el caso particular de un oscilador de bloques de baja frecuencia, con la estructura descrita en el Capítulo anterior.

Por último, en los apartados 4 y 5, se describen los programas FORTRAN y en código TMS realizados, así como los resultados experimentales obtenidos con la realización del diseño anterior sobre los medios de que se ha dispuesto.

Hay que señalar que la realización de este Capítulo ha sido posible gracias a los medios disponibles en el Dpto. de I+D de Telefónica y que la elección del procesador TMS32010 se ha visto motivada fundamentalmente por el hecho de encontrarse éste y los accesorios adicionales disponibles en el citado Dpto.

4.1. MANEJO Y APLICACIONES DEL TMS32010 DSP.

4.1.1. MANEJO.

EL TMS32010 DSP pertenece a la primera generación de procesadores digitales de señal [8]. Su "hardware" realiza una gran cantidad de funciones que otros procesadores se ven obligados a realizar por "software". Por ejemplo, contiene un multiplicador por "hardware" que realiza una multiplicación de 16 · 16 bits, con resultado en 32 bits, en un solo ciclo de instrucción de 200ns.

Contiene también un registro de desplazamiento "hardware" que se usa para desplazar los datos en su camino a la ALU (Unidad Aritmético-Lógica). Además, incluye registros auxiliares que proporcionan direcciones indirectas de memoria de datos RAM y que permiten la manipulación de la tabla de datos.

En la Figura IV.1, se encuentra un diagrama de bloques del procesador, cuyas características generales describimos a continuación:

- Ciclo de instrucción de 200ns.
- Bus de datos e instrucciones de 16 bits.
- RAM interna de $144 \cdot 16$ bits = 288 bytes.
- Dos modos de trabajo ó versiones seleccionables:
 - Microprocesador, μ P TMS32010.
 - Microcomputador, μ C TMS320M10.
- En el caso de la versión μ C, dispone de una ROM interna de $1,5K \cdot 16$ bits (3Kbytes).
- En el caso de la versión μ P, se pueden direccionar un total de $4K \cdot 16$ bits.
- La ALU y el Acumulador son de 32 bits.
- Realiza multiplicación "hardware" en 200ns
 $16 \cdot 16 \longrightarrow 32$ bits
- Tiene 8 puertos de entrada y ocho de salida.
- Realiza las operaciones en punto fijo con aritmética de complemento a 2.

- Realiza desplazamientos aritméticos por "hardware" a la entrada y a la salida del acumulador.
- Está fabricado en tecnología NMOS de 2,4 y se produce en pastillas con 40 "pines" con una tensión de alimentación de 5 voltios.

Haremos ahora un análisis detallado de la arquitectura del procesador, así como de sus principales elementos.

4.1.1.1. Arquitectura.

El procesador TMS32010 emplea una arquitectura HARVARD modificada. Un esquema general de este tipo de arquitecturas se encuentra en la Figura IV.2.

Básicamente se caracteriza porque la memoria de programa y la de datos se encuentran separadas, con lo cual es posible un completo solapamiento entre la ejecución y la búsqueda de la instrucción, lo que trae como consecuencia un incremento de la velocidad. La memoria de programa reside en una ROM interna y la memoria de datos en una RAM interna.

En el caso del TMS32010, la memoria de programa reside en una ROM (1,5K palabras) en el interior ó fuera del "chip", y el resto, hasta un total de 4K palabras, fuera del mismo. En total se pueden direccio-

nar 4K palabras de memoria de programa directamente. Para la memoria de programa exterior se necesitan memorias rápidas, de tiempos de acceso por debajo de los 100ns.

La memoria de datos es una memoria RAM de 144 palabras, con la característica de que todos los operandos de las instrucciones se buscan en esta RAM, ningún operando se buscará directamente fuera. Ahora bien, y como veremos más adelante, existe la posibilidad de transferencia entre el espacio de datos y el espacio de programa. Esta modificación de la arquitectura Harvard incrementa la flexibilidad de la estructura y posibilita, entre otras cosas, el almacenamiento de coeficientes en memoria de programa y su transferencia a la memoria de datos, y el direccionamiento inmediato así como el salto a subrutinas basado en valores calculados.

4.1.1.2. Elementos Aritméticos.

Existen cuatro elementos aritméticos básicos en el procesador: la ALU o Unidad Aritmético-Lógica, el Acumulador, el Multiplicador y los Registros Desplazamiento. A continuación analizaremos cada uno de ellos.

4.1.1.2.-1. UNIDAD ARITMETICO-LOGICA (ALU).

La Unidad Aritmético-Lógica, de propósito general, opera con palabras de 32 bits, es decir con precisión de 32 bits.

Realiza las operaciones aritméticas de suma y resta en complemento a dos, además de las operaciones lógicas AND, OR, y XOR. Y frente a desbordamientos, presenta dos modos de operación : modo "saturación" y modo "desbordamiento" [8].

4.1.1.2.-2. ACUMULADOR.

El acumulador se encarga del almacenamiento de la salida de la ALU y también es, con frecuencia, la entrada a la misma. Trabaja con 32 bits, estando dividido en dos partes de 8 bits cada una, la parte alta y la parte baja. El conjunto de instrucciones proporciona el modo de cargar tanto la parte alta como la baja, así como la forma de proporcionar los datos con o sin desplazamiento de bits.

4.1.1.2.-3. MULTIPLICADOR.

Es un multiplicador "hardware" de 16*16 bits y que consta de dos registros T y P. El primero es de 16 bits y almacena el multiplicando, el segundo,

es un registro de 32 bits que almacena el producto.

Para realizar la multiplicación, el multiplicando se carga en el registro T desde la Data RAM ó Memoria de Datos. A continuación se lleva a cabo la multiplicación y el resultado queda cargado en el registro P. En el procesador TMS32010 se ha incorporado un "hardware" que obliga a almacenar, una vez que se ha realizado la multiplicación, el contenido del registro P en el Acumulador.

4.1.1.2.-4. REGISTROS DE DESPLAZAMIENTO.

Existen dos registros "hardware" que se utilizan. Uno para "manipular" los datos que, provenientes de la Data RAM, se introducen en el Acumulador para ser cargados, sustraídos o sumados. Este registro realiza un desplazamiento de 0 a 15 lugares hacia la izquierda de dichos datos, haciendo a su vez la conversión a 32 bits.

El segundo registro "manipula" los datos del Acumulador que se dirigen hacia la Data RAM, en el sentido de que desplaza 0,1, ó 4 lugares hacia la izquierda los bits que constituyen la palabra, proporcionando de esta forma la pérdida de los bits altos.

Puede comprobarse [8] que por medio de estas operaciones se consiguen multiplicaciones y divisiones por potencias de dos.

4.1.1.3. Memoria de Datos (Data RAM).

La memoria de datos consta, como ya hemos dicho, de 144 palabras de 16 bits de RAM interna. Estas se encuentran paginadas en dos páginas, la página 0 con 128 palabras y la página 1 con 16 palabras, según se indica en la Figura IV.3.

En la memoria se encontrarán almacenados todos los datos que se usan como operandos en las instrucciones. Es importante señalar que las instrucciones sólo direccionan los operandos contenidos en la memoria de datos. Ahora bien, se pueden tener almacenados datos fuera de esta memoria interna y leerlos y pasarlos a la memoria RAM interna conforme vayan siendo necesitados. Para ello existen las instrucciones TBLR, TBLW, IN, OUT.

En cuanto a los modos de direccionamiento de la memoria de datos, existen dos: directo e indirecto.

En el modo indirecto se usan los ocho bits menos significativos de los registros auxiliares AR0, AR1, para almacenar las direcciones de la memoria de datos. Estos registros pueden, a su vez, ser incrementados o decrementados durante una instrucción con direccionamiento indirecto, como puede verse en [8].

4.1.1.4. Memoria de Programa

La memoria de programa consta de hasta 4K palabras de 16 bits. Dependiendo de si el procesador trabaja como microprocesador o microcomputador, esas palabras serán todas externas ó 1,5K serán internas y el resto externas.

El mapa de memoria puede observarse en la Figura IV.4, en (a) para la versión microcomputador y en (b) para la versión microprocesador.

La ejecución del programa contenido en la memoria de programa externa se realiza a la misma velocidad que el contenido en la memoria de programa interna, y, además, hemos de señalar que en ambas memorias pueden existir datos que pasarán a la memoria de datos cuando se necesiten.

En lo que se refiere al uso de una memoria de programa externa, la configuración posible del sistema puede verse en la Figura IV.5. Esta configuración implica la necesidad de empleo de memorias rápidas, de tiempo de acceso inferior a 100ns.

En el caso de que la memoria sea RAM, habrá que tener en cuenta un problema que se plantea y que se tendrá que prevenir con circuitería especial. Consiste en que las instrucciones de escritura en puertos equivalen o producen escritura en las ocho primeras posiciones de la RAM. Esto quiere decir que si se produjera durante el programa alguna escritura en puerto, se al

terarían las instrucciones de programa de las primeras posiciones. Esto se previene prohibiendo escritura en las ocho primeras posiciones de la RAM.

4.1.1.5. Registros.

Haremos ahora una breve descripción de los registros del procesador en cuestión:

- AR0,AR1.- Son los registros auxiliares de 16 bits que realizarán las funciones de almacenamiento temporal y direccionamiento indirecto para el que se usan los 8 bits más bajos.

- PC.- Es el contador de programas, que emplea 12 bits para el direccionamiento de las 4K palabras del programa.

- PILA.- Contiene cuatro niveles de 12 bits y se caracteriza porque no se detectan desbordamientos en la misma.

- REGISTRO DE STATUS.- Es un registro de 5 bits:

OV.- Indica desbordamiento en el acumulador.

OVM.- Define el comportamiento del microprocesador frente a desbordamientos.

INTM.- Es la máscara de interrupciones.

ARP.- Es el puntero de los registros auxiliares. Selecciona el registro auxiliar en curso.

DP.- Puntero de página de la memoria de datos.

Cuando se efectúan transferencias entre el registro de STATUS y la memoria de datos se modifican los bits que se indican en la Figura IV.6.

Desde el registro de STATUS se puede acceder únicamente a la página 1 de la memoria de datos usando direccionamiento directo. Si es indirecto se accederá a cualquier posición. Por otro lado, desde la memoria de datos al registro de status es accesible cualquier posición de la RAM.

4.1.2. APLICACIONES.

Entre las aplicaciones del TMS32010 nos interesa precisamente el procesado de señales, debido a que es éste un procesador diseñado y creado con ese propósito especial.

En el procesado de señales habrá que distinguir dos tareas fundamentales. La primera consistirá en la toma de datos en sí a través del convertor A/D correspondiente y la segunda en el procesado propiamente dicho de la señal.

Para realizar este trabajo existen dos opciones una primera en la que el TMS32010 atiende directamente al conversor A/D (Figura IV.7(a)), y una segunda en la que el procesador atiende a un elemento intermedio (Figura IV.7(b)).

La primera opción tiene la ventaja de su sencillez cuando lo que se necesita es el procesamiento muestra a muestra en los casos en los que el tiempo de proceso es menor que el periodo de muestreo. Pero presenta inconvenientes si el procesado requiere un conjunto de muestras, ya que en ese caso se pierde parte de la señal. Presenta también una mala gestión del tiempo del procesador y no sirve si se necesita reconstruir la señal a partir del procesado de un bloque de muestras. Estos problemas se solventan con la segunda configuración, a costa de un aumento en la complejidad de la circuitería.

4.2. SOFTWARE Y HARDWARE ADICIONAL.

Una vez descritas las características más significativas del procesador TMS32010, nos vemos en la necesidad de describir el equipo adicional empleado para la completa realización del oscilador de baja frecuencia sobre el mismo.

Distinguiremos entre herramientas de tipo "Software" y herramientas tipo "Hardware".

4.2.1. HERRAMIENTAS "SOFTWARE".

Entre éstas se encuentran disponibles para PC y para VAX un Ensamblador/Linkador y un Simulador, cuyo objetivo es desarrollar y depurar los algoritmos realizados para el TMS32010.

El Macro-ensamblador traduce el código fuente en lenguaje ensamblador TMS320 en código objeto ejecutable. Nos permite, así, trabajar en lenguaje nemónico en vez de hexadecimal, pudiéndose referenciar las posiciones de memoria con direcciones simbólicas. El Macro-ensamblador soporta llamadas a "Macros" y definiciones con el ensamblador [9].

El Linkador permite que el programa sea diseñado e implementado en módulos separados, que más adelante se encadenan para formar el programa. El Linkador crea un fichero objeto que puede ser ejecutado por el Simulador o por el Emulador.

El Simulador es un programa Software que, como su propio nombre indica, simula las operaciones del TMS32010 permitiendo con ello la verificación del programa realizado. El modo "DEBUG" nos permite conocer el estado interno de los registros del TMS32010 simulado mientras el programa se ejecuta. El Simulador usa el código objeto producido por el Linkador.

Durante la ejecución del programa los registros internos y la memoria del TMS32010 simulado se modifican conforme cada instrucción es interpretada por el "HOST", es decir, por el PC o el VAX según sea el caso. Una vez que la ejecución del programa se suspende, los registros internos y los datos de memoria y programa se pueden inspeccionar e incluso modificar.

4.2.2. HERRAMIENTAS HARDWARE.

Entre las herramientas "Hardware" se ha dispuesto de:

- a) una placa de evaluación EVM [10]
- b) una placa de interfase analógico/digital AIB [11].

4.2.2.a. Placa EVM.

El EVM es un sistema de desarrollo en una sola tarjeta que nos permite comprobar si el TMS32010 cumple los requerimientos de tiempo en una aplicación concreta. Esta tarjeta contiene todas las herramientas necesarias para evaluar el dispositivo, así como para proporcionar una emulación del mismo "en el circuito".

Este potente sistema contiene un "Debugger", Editor, Ensamblador, Programador de EPROM, Comunicación por Software a dos puertos EIA y una interfase de audio-casette.

En dicha tarjeta se proporciona también la comunicación a un "Host Computer" y varios periféricos. Los dos puertos EIA permiten que la EVM pueda conectarse a un terminal y/o a una impresora de línea o a un "Host Computer".

Con todo ello, puede funcionar como un sistema de desarrollo empleando su propio editor de textos para creación de ficheros en Ensamblador TMS y las interfaces a cassette y cinta como sistema de almacenamiento. Pero, también puede, y así es como lo hemos empleado, aceptar ficheros de una CPU "Host" a través de uno de sus puertos.

En cualquiera de los dos casos el Ensamblador de la tarjeta convertirá el texto de entrada en código ejecutable en una sola pasada. Este código objeto se almacena en un espacio de memoria de 4 Kb con lo cual se permite la utilización completa del espacio de direcciones del TMS32010.

Esta placa permite también la lectura del programa a ejecutar desde una EPROM, así como la grabación del mismo en una EPROM.

4.2.2.b. Placa AIB.

La tarjeta interfase analógico/digital TMS32010AIB, no es más que una tarjeta de conversión A/D y D/A que se puede usar en conjunción con la EVM. Contiene dos convertidores A/D y D/A de 12 bits, con posibilidad de expansión para otros convertidores adicionales.

La frecuencia de reloj para la tarjeta se obtiene de la salida CLKOUT del TMS32010 y puede programarse para tener entradas y salidas analógicas

periódicas.

Dispone, además, de dos filtros paso bajo. Uno a la entrada del conversor A/D, que limita la señal y evita o minimiza, en lo posible, los efectos de "aliasing" [12]. Y otro que se encuentra a la salida del D/A y suaviza la misma. La respuesta en frecuencia de estos filtros se controla modificando las componentes externas en sus etapas y su frecuencia de corte es programable entre 4,7 kHz y 20 kHz.

En la Figura IV.8 se encuentra un esquema de bloques de la configuración general del sistema de desarrollo para una aplicación determinada sobre el TMS32010.

En nuestro caso, como se trata de realizar un oscilador, no nos preocupará la entrada, con lo cual nos fijaremos únicamente en la salida.

Ya hemos dicho que la frecuencia de muestreo se obtiene del propio TMS32010, programando un divisor por N.

En principio, esta constante N es de 16 bits y puede cargarse en el registro sacándola por el puerto 1 de la memoria de programa, a través de la instrucción OUT. Podría obtenerse así, dado que la frecuencia de reloj del TMS es de 5 MHz, una frecuencia comprendida entre 76.29 Hz y 5 MHz, pero existe una limitación impuesta por el conversor A/D, con lo cual el rango de variación de N está entre 144 y 65535. Esto origina que las frecuencias que pueden obtenerse oscilen entre

76.29 Hz y 34.48 KHz.

Más información sobre las dos placas anteriores puede encontrarse en [10,11].

4.3. PROCEDIMIENTO GENERAL PARA LA REALIZACION DE UN SISTEMA SOBRE EL PROCESADOR.

Nos interesa ahora abordar el problema de descripción de un método general de trabajo para la realización de un algoritmo determinado sobre un TMS32010.

En nuestro caso nos plantearemos, siguiendo el tema del Capítulo anterior, el caso particular de la realización de un oscilador de bloques de baja frecuencia, es decir, de frecuencia de oscilación pequeña comparada con la de muestreo.

En primer lugar, haremos una serie de consideraciones sobre la aritmética y formato empleado por el TMS32010 en la representación y realización de operaciones aritméticas.

Como ya hemos visto en un apartado anterior, los cálculos en el TMS32010 se basan en una representación de los números en punto fijo y con aritmética de complemento a 2 [13]. El rango de valores posibles o de números representables variará entre -128 (1000000000000000) y 127996(0111111111111111).

Un número determinado se representará con un bit de signo, i bits de parte entera y $(15-i)$ bits decimales, en cuyo caso se dice que el número se representa en formato $Q(15-i)$.

Este hecho impone la necesidad de conocer en cada instante el formato de cada variable o coeficiente empleado, para poder realizar con precisión y correctamente cualquier operación aritmética. Así, siempre que sea posible se intentará unificar formatos y emplear la misma notación para todas las variables. Normalmente se suele trabajar o con fracciones o con enteros, es decir, con los formatos $Q15$ o $Q0$ respectivamente.

En la Figura IV.9 se representan dos magnitudes A, B , una en formato $Q15$ y otra en $Q14$. En general, $N1$ y $N2$ serán, olvidando cualquier tipo de representación dos números enteros expresados en complemento a 2: 1 bit signo + 15 bits. Las magnitudes A y B se obtendrán de los mismos por:

$$\begin{aligned}
 A &= N_1 \cdot 2^{(-15)} && \in R \\
 B &= N_2 \cdot 2^{(-14)} && \in R
 \end{aligned}
 \tag{4.1}$$

Ahora bien, el supuesto anterior puede no cumplirse y, si los rangos de variación de los coeficientes y variables difieren mucho unos de otros, será necesario emplear formatos distintos para los mismos. Esto no implicará más que el hecho de que habrán de tenerse en cuenta los formatos de las variables a la hora de realizar las operaciones que conlleve el algoritmo y así, por ejemplo, a la hora de sumar habrá que alinear el punto decimal y a la hora de multiplicar habrá que tener en cuenta el número de bits enteros del resultado [8].

Todo esto nos lleva, pues, a definir un primer paso en nuestro proceso de realización del algoritmo, que consistirá en realizar una simulación del mismo en un lenguaje de alto nivel, que en nuestro caso ha sido FORTRAN, para obtener el rango dinámico de los coeficientes y variables del sistema que se quiere implementar, es decir, para conocer el rango dinámico de la señal en cualquier punto de la estructura que se desea realizar y poder determinar así su formato de representación en el TMS32010.

Hay que hacer notar que con esta simulación en FORTRAN, como es una simulación en máxima precisión, sólo se resuelve el problema de representación de las variables o señales pero no los problemas derivados de la cuantización y de los redondeos que son

consecuencia de la longitud finita de los registros del procesador.

Para tenerlos en cuenta habría que realizar un programa FORTRAN que simulara ante cada operación el redondeo y las cuantizaciones, como de hecho se realizó en el Capítulo 3.

Una vez encontrados los formatos para cada variable, se procederá a escribir el algoritmo en lenguaje ensamblador del TMS32010.

No parece interesante ahondar más en las instrucciones y forma de realizar operaciones sobre el TMS32010, ya que todas las posibilidades se encuentran ampliamente detalladas en [8].

Después de escrito el programa en ensamblador, se ensambla y se encadena con ayuda de las herramientas antes descritas, con lo que se dispone ya del programa en código ejecutable TMS.

Por último y antes de su emulación, se simula el diseño realizado con ayuda del programa Simulador, con lo que se verifica el buen comportamiento del sistema. Si los resultados son los deseados se procede a la emulación con la ayuda del sistema de la Figura IV.8.

4.4. DESCRIPCION DEL OSCILADOR DE BLOQUES REALIZADO.

Centrémonos ahora en el caso concreto del diseño del oscilador de bloques representado en la Figura IV.10, cuyas características generales son:

- Orden de bloques, $L=4$
- Frecuencia, $N=f_m/f_o=1024$, f_m es la frecuencia de muestreo

A partir de dichas características y con ayuda del programa OSC.FOR, en FORTRAN 77, que se incluye al final del Capítulo, y que no es más que la versión particularizada al caso concreto del programa empleado en el Capítulo anterior para la simulación de osciladores con esa estructura, se obtienen los siguientes valores para los coeficientes y condiciones iniciales del oscilador:

s=n bits parte entera del acumulador=6

B=-0.03855

COEF(2)=-9.638739E-3

COEF(3)=-1.927711E-3

COEF(4)=-2.891477E-2

Condiciones iniciales de las variables de estado:

V1=-0.6365858

(4.2)

V2=0

Además, las ecuaciones que proporcionan las salidas del oscilador serán:

YOUT(1)= V2

(4.3)

YOUT(I)=V1 COEF(I)+V2 I=2,3,4

y las condiciones iniciales para el ciclo siguiente:

$V1=[V1 \cdot 2^S + B \cdot V1 + V2] \cdot 2^{-S}$

(4.4)

$V2=[V1 \cdot B \cdot 2^S + V2 \cdot 2^S] \cdot 2^{-S}$

Con todo esto podemos elegir un formato para los coeficientes. Puesto que todos son menores que uno y la estructura se encuentra escalada a la unidad, elegimos el formato Q15, es decir, 15 bits de parte decimal y uno de signo con lo cual los valores a almacenar en el programa TMS serán los dados por (4.2) multiplicados por 2^{15} , es decir:

$$B=-1263$$

$$\text{COEF}(2)=-316$$

(4.5)

$$\text{COEF}(3)=-632$$

$$\text{COEF}(4)=-947$$

En cuanto a las variables de estado V1 y V2 se representarán en formato Q14, es decir, 1 bit signo+1 bit entera+14 bit parte decimal, y por tanto los valores de (4.2) se deberán multiplicar por 2^{14} con lo que

$$V1=-10430$$

$$V2=0$$

Con estos datos se puede obtener un oscilador con una frecuencia 1024 veces menor que la de muestreo y también cuatro osciladores de frecuencia $4(f_m/1024)$.

El listado del programa en código ensamblador TMS, OSC.TMS se incluye al final del Capítulo. Asimismo, se incluye éste ya adaptado a su aplicación concreta sobre la placa de emulación. En dicho programa se ha tenido que "programar" la frecuencia de muestreo, que por las razones expuestas anteriormente no puede ser superior a 34.48KHz.

4.5. RESULTADOS EXPERIMENTALES:

SIMULACION Y EMULACION.

A) SIMULACION

En primer lugar, se ha procedido a la simulación del oscilador mediante el programa simulador del TMS32010 que corre sobre VAX.

En la Figura IV.11 se muestran las cuatro señales obtenidas, correspondientes a las salidas YOUT(i) (i=1,4) de la Figura IV.10, así como la señal escalar, y(n), correspondiente. En todas ellas se han representado un total de 1024 puntos, comprobándose que, efectivamente, se cumplen las relaciones de frecuencia buscadas.

En la Figura IV.12 se representa el módulo de la transformada DFT de dichas señales, M_1, M_2, M_3, M_4 y M , para cuyo cálculo se han tomado 1024 muestras y se ha empleado el programa FAST del paquete de programas IEEE [14].

Se pone de manifiesto el buen comportamiento del oscilador, tal y como se había comentado en el Capítulo anterior, así como su baja distorsión (no hay componentes de frecuencia ajenas).

B) EMULACION.

Una vez comprobado el algoritmo TMS, procedemos a verificar si cumple los requerimientos de tiempo. Esto lo hacemos a través de la emulación del comportamiento del TMS32010 con ayuda del sistema de la Figura IV.8.

Como ya hemos dicho tomamos como frecuencia de muestreo la mayor posible, esto es, 34.48 KHz, con lo cual si la realización del oscilador es correcta obtenemos cuatro salidas de frecuencia 134.68 Hz.

La codificación del programa ensamblador para la aplicación en tiempo real viene condicionada por la memoria disponible en el procesador. El TMS32010, como ya hemos visto, puede direccionar una memoria de $4K \cdot 16$ bits, que es suficiente para ubicar el código ejecutable de nuestra estructura, dentro de los márgenes donde es buena la aproximación que se utiliza.

utiliza.

Por otro lado, existirá una limitación en cuanto al orden de la estructura de bloques a realizar y que podrá venir impuesta o por la memoria de datos o por el tiempo de ejecución del algoritmo. Analizaremos a continuación el gasto de memoria de datos y los tiempos de inicialización y de ejecución de nuestro programa.

Bl. Memoria.

El procesador dispone de 128 posiciones de memoria, de las cuales un oscilador de orden L ocuparía

$$7+2L \qquad (4.6)$$

con lo cual el orden máximo de bloques sería, aproximadamente, 60.

Lógicamente se induce de ello que la memoria de datos no será limitación a la hora de aumentar el orden de la estructura de bloques y, que ésta vendrá impuesta muy posiblemente por el tiempo de ejecución del programa.

En nuestro caso, para $L=4$, se han ocupado 15 posiciones de memoria, lo cual representa el 11.5% de la memoria disponible.

B2. Tiempo de ejecución.

Para un oscilador de orden L , teniendo en cuenta que el ciclo de instrucción del procesador es de $0.2 \mu\text{s}$, el tiempo de ejecución del programa vendrá dado por

$$(7L+14)0.2 \mu\text{s} \quad (4.7)$$

Puesto que el ciclo disponible es de $(1/f_m)\text{s}$, el máximo orden posible de la estructura vendrá dado por:

$$(7L+14)0.2 \cdot 10^{-6} = 1/f_m \quad (4.8)$$

que para el caso que nos ocupa, en el cual $f_m=34.48\text{KHz}$ es, aproximadamente, igual a 18.

Para $L=4$ el tiempo de ejecución correspondiente es de $8.4 \mu\text{s}$, lo cual supone aproximadamente el 29% del ciclo disponible.

B3. Tiempo de inicialización.

Es el tiempo empleado en la realización para la inicialización de las variables. Para una estructura de orden L , este tiempo viene dado por:

(43+7L)0.2 μ s

(4.9)

que en nuestro caso es de 14.2 μ s.

Puesto que sólo disponemos de una salida, esto es, de un conversor D/A, conectamos ésta a la entrada de un osciloscopio sobre cuya pantalla podemos visualizar la forma de onda de una de las salidas del oscilador. Una fotografía de la pantalla del osciloscopio se presenta en la Figura IV.13.

Para una mejor visualización sobre la pantalla del osciloscopio, sería interesante una frecuencia mayor. Esta podría obtenerse, o aumentando el orden de los bloques, o disminuyendo la relación entre la frecuencia de muestreo y dicha frecuencia. En este último caso nos encontraríamos con la limitación de que la estructura es tanto mejor cuanto mayor es dicha relación.

En cualquier caso, se comprueba el buen comportamiento de la estructura propuesta y se pone de manifiesto la posibilidad de obtener osciladores de baja frecuencia con baja distorsión.

4.6. REFERENCIAS.

- [1]. Oppenheim and Schaeffer, "Digital Signal processing". Prentice Hall. Englewood Cliffs. 1975
- [2]. L.R.Rabiner and B.Gold, "Theory and application of digital signal processing". Prentice Hall. Englewood Cliffs. 1975.
- [3]. Oppenheim, "Applications of digital signal processing". Prentice Hall. 1978.
- [4]. M.Schwartz, "Computer Communications network design and analysis". Prentice Hall. 1977.
- [5]. K.Feher, "Digital Communications". Prentice Hall. Englewood Cliffs. 1983.
- [6]. I.Korn, "Digital Communications". Van Nostrand, R.C. Inc. 1985.
- [7]. K.Feher, "Digital Communications: Microwave Applications". Prentice Hall. 1981.
- [8]. TMS32010 User's Guide. Texas Instruments.
- [9]. TMS32010 Assembly Language Programmer's Guide. Texas Instruments.

- [10]. TMS32010 Evaluation Module User's Guide. Texas Instruments.
- [11]. TMS32010 Analog Interface Board User's Guide. Texas Instruments.
- [12]. L.R.Rabiner, R.W. Schaeffer, "Digital Processing of speech signals". Prentice Hall. 1978.
- [13]. A. Antoniou, "Digital Filters: Analysis and design". Mc.Graw Hill. 1979.
- [14]. "Programs for Digital Signal Processing" Digital Signal Processing Committee IEEE Acoustics, Speech and Signal Processing Society. IEEE Press. 1979.

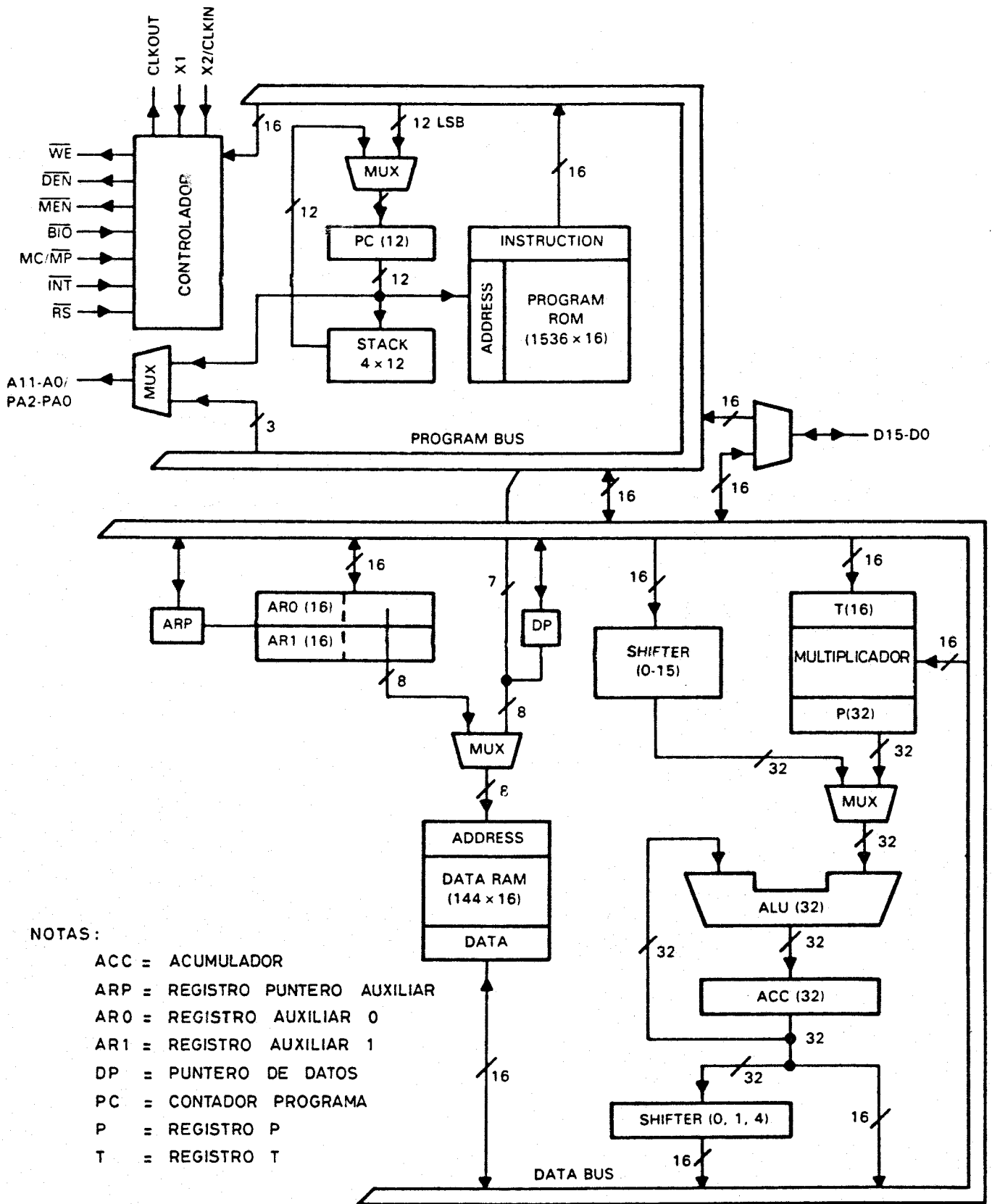
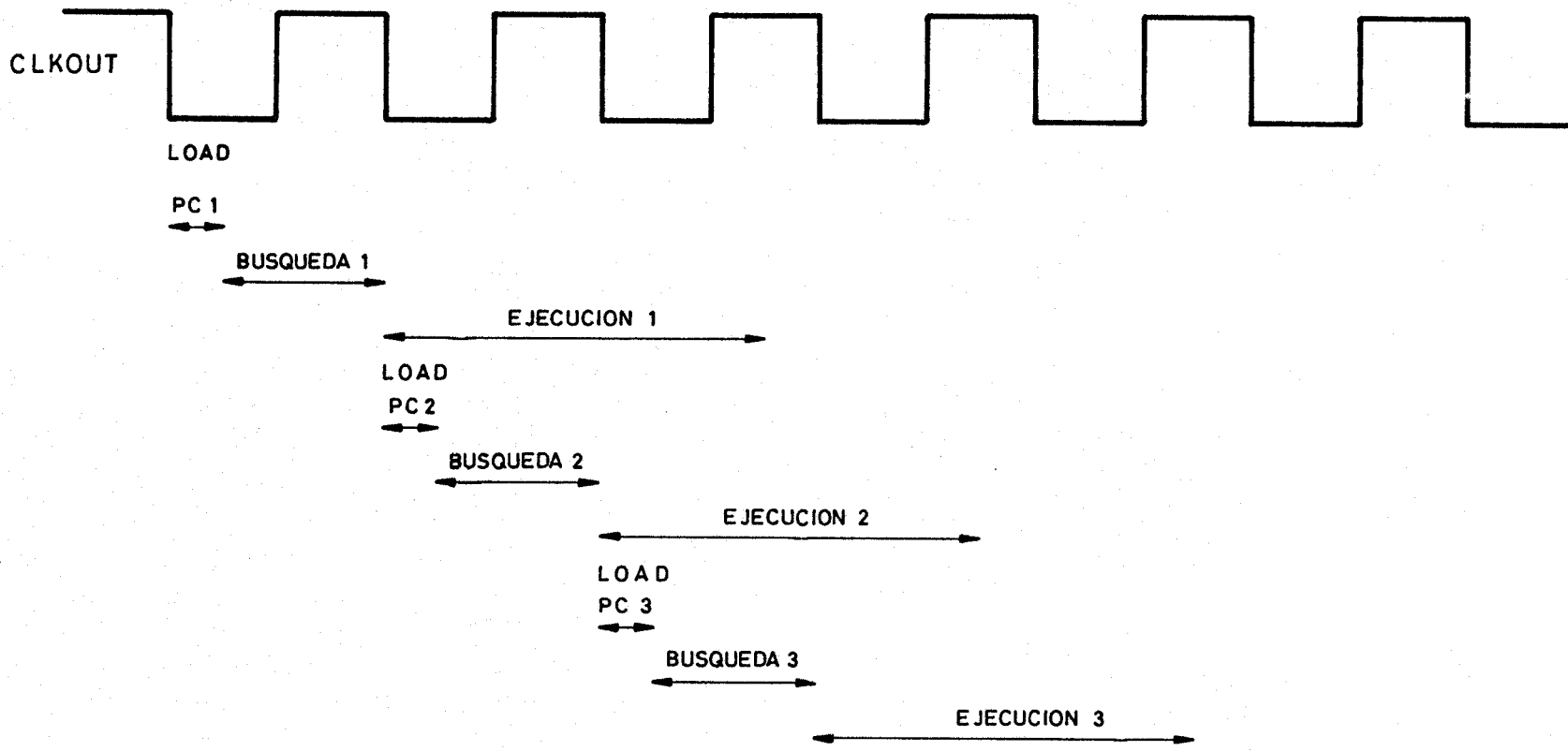


FIGURA IV 1 - ARQUITECTURA DEL TMS 32010



4.37

FIGURA IV. 2 :- ARQUITECTURA HARVARD

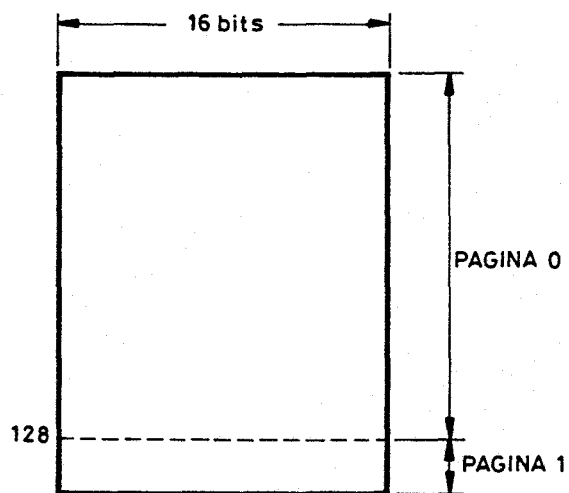


FIGURA IV. 3 : ESQUEMA DEL MAPA DE MEMORIA DE DATOS

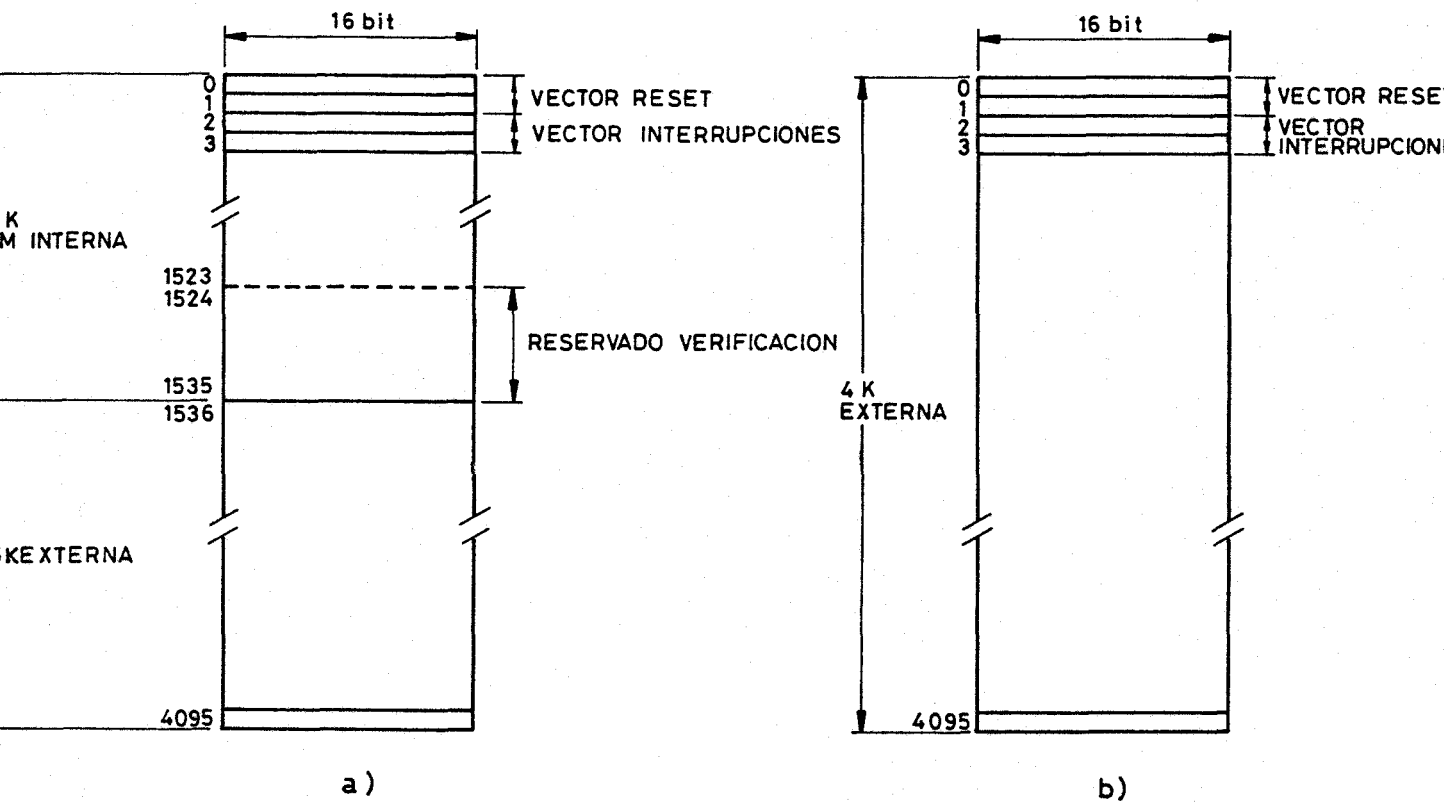


FIGURA IV. 4 - MAPA DE MEMORIA DE PROGRAMA

a) TMS 320M10 b) TMS 32010

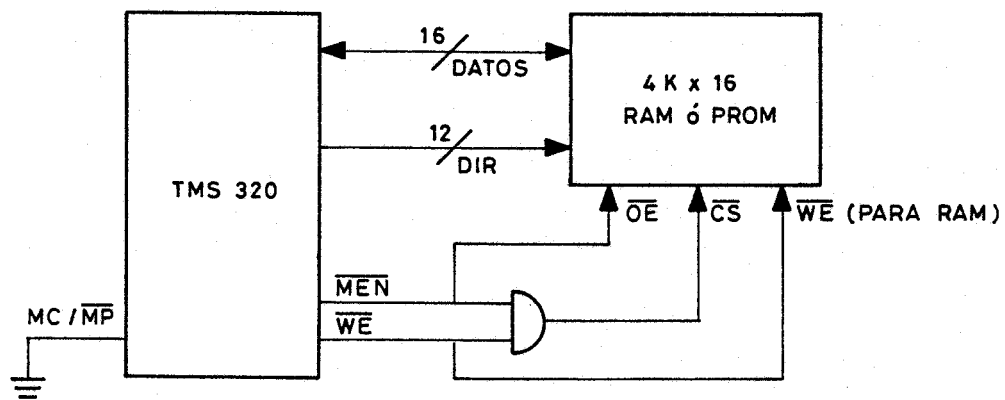


FIGURA IV 5.- CONFIGURACION CON MEMORIA DE PROGRAMA EXTERNA

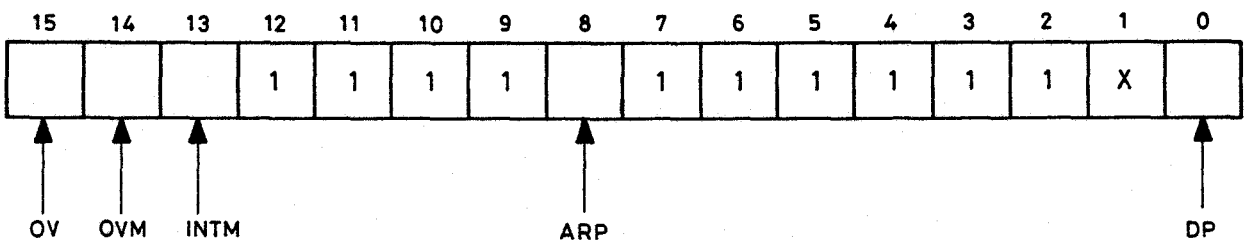
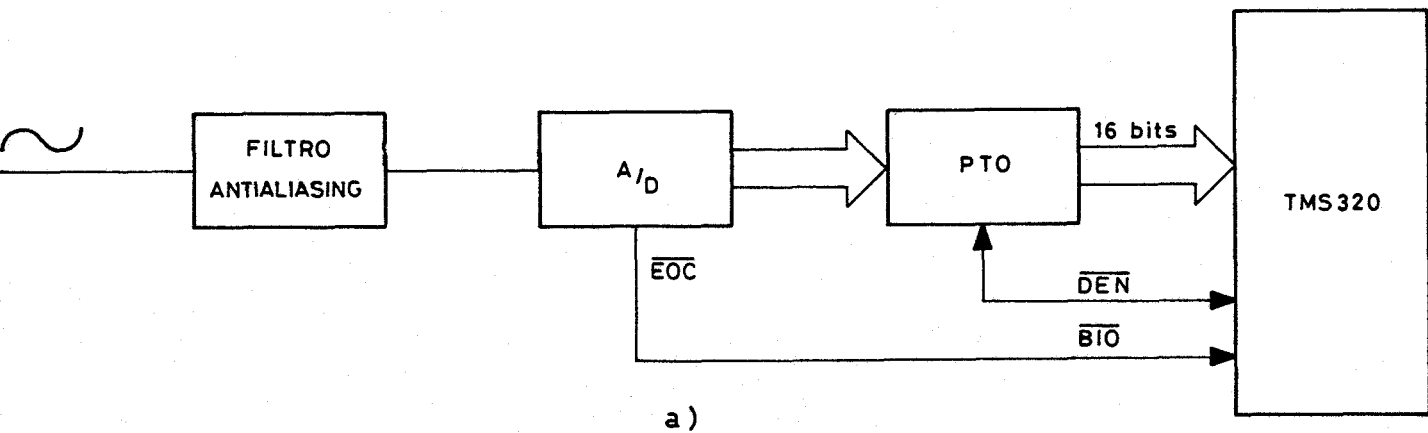
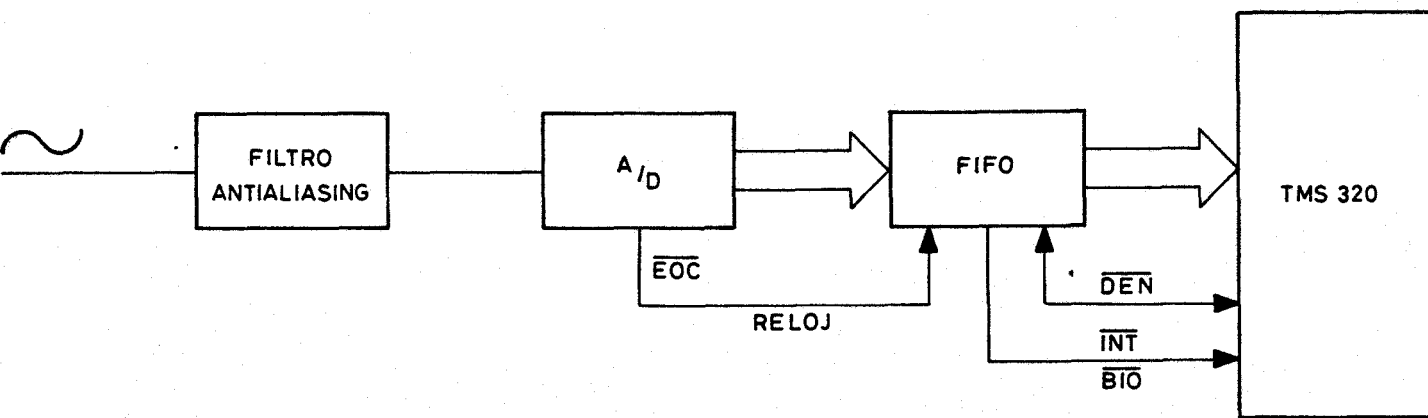


FIGURA IV. 6 :- BITS MODIFICADOS CON LAS TRANSFERENCIAS ENTRE REGISTRO STATUS Y MEMORIA DE DATOS



a)



b)

FIGURA IV. 7 - APLICACIONES DEL TMS32010 AL PROCESADO DE SEÑALES

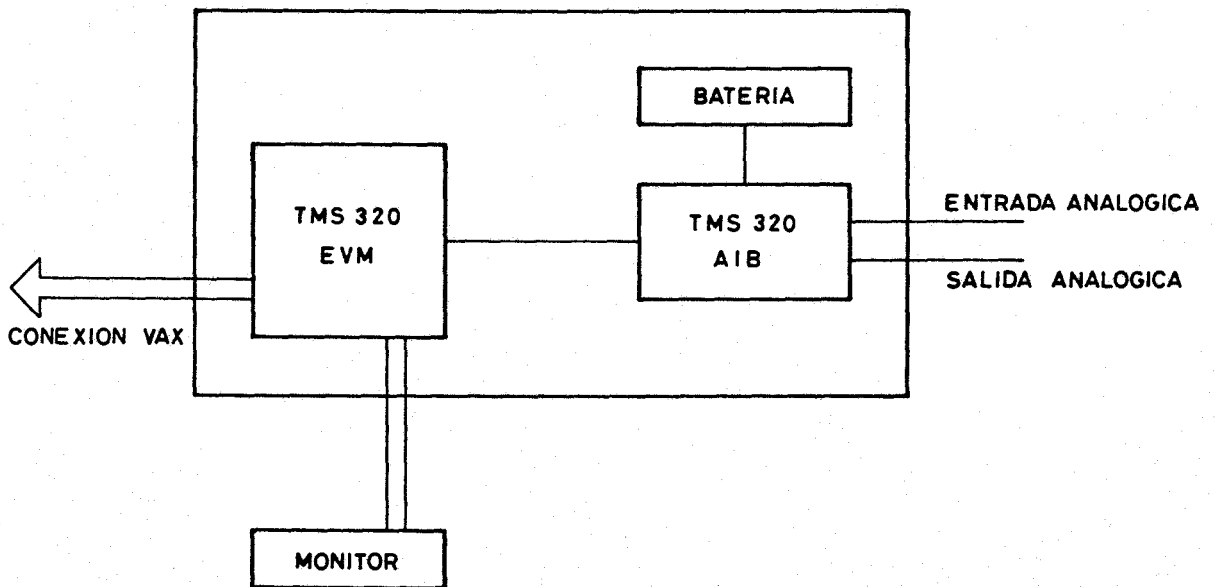
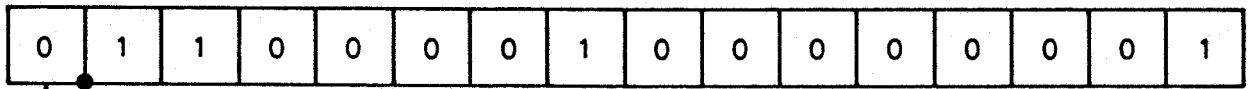
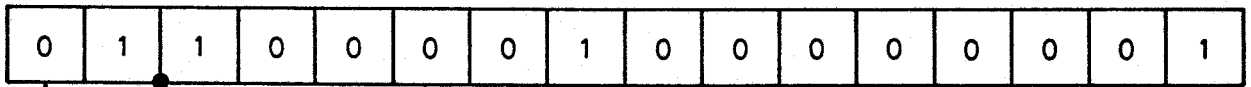


FIGURA IV 8 - CONFIGURACION DEL SISTEMA DE DESARROLLO



a) N1



b) N2

FIGURA IV. 9 - REPRESENTACION DE MAGNITUDES EN DIFERENTES FORMATOS :

a) Q15

b) Q14

4.45

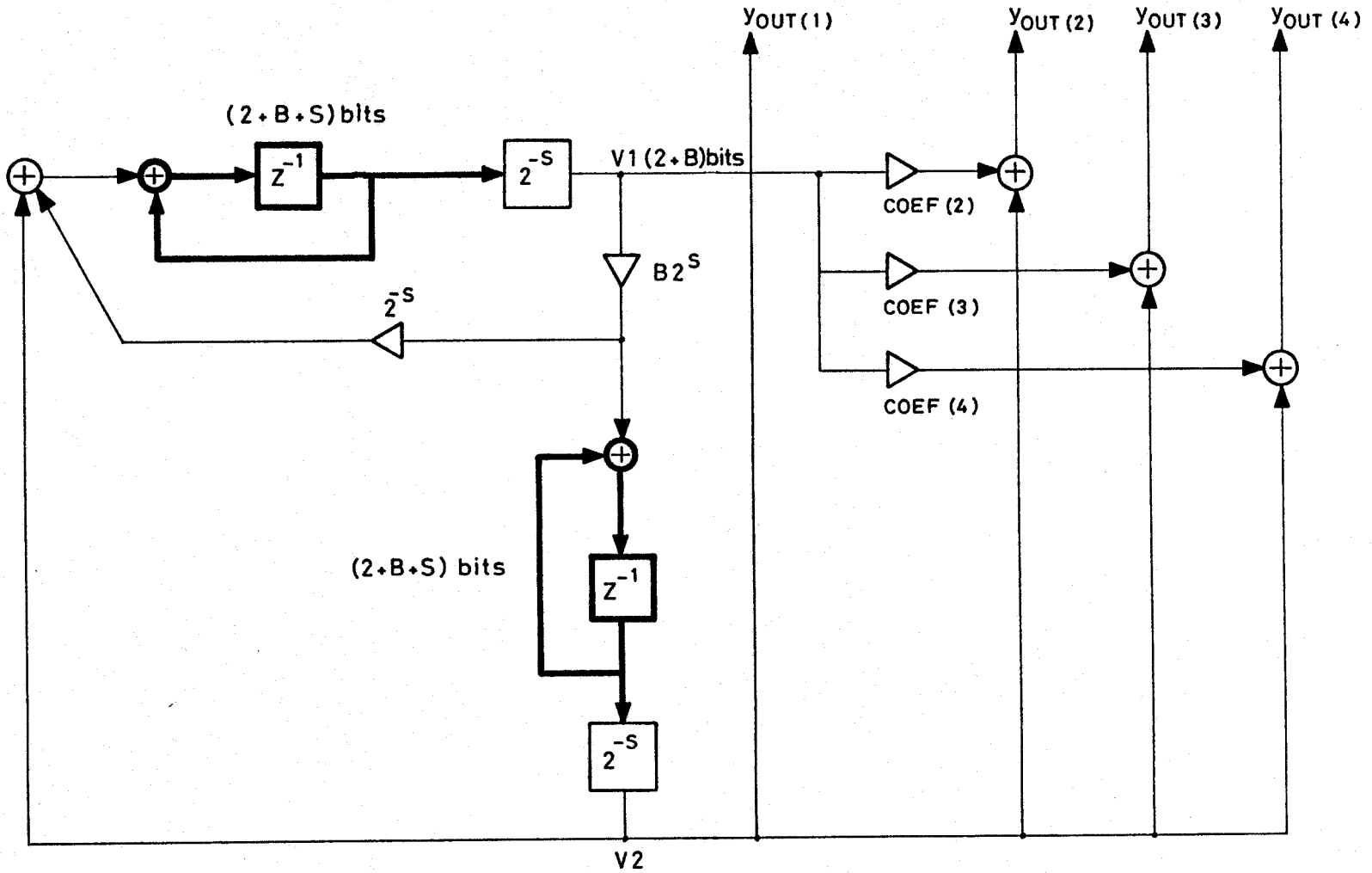


FIGURA IV 10 - OSCILADOR DE BLOQUES

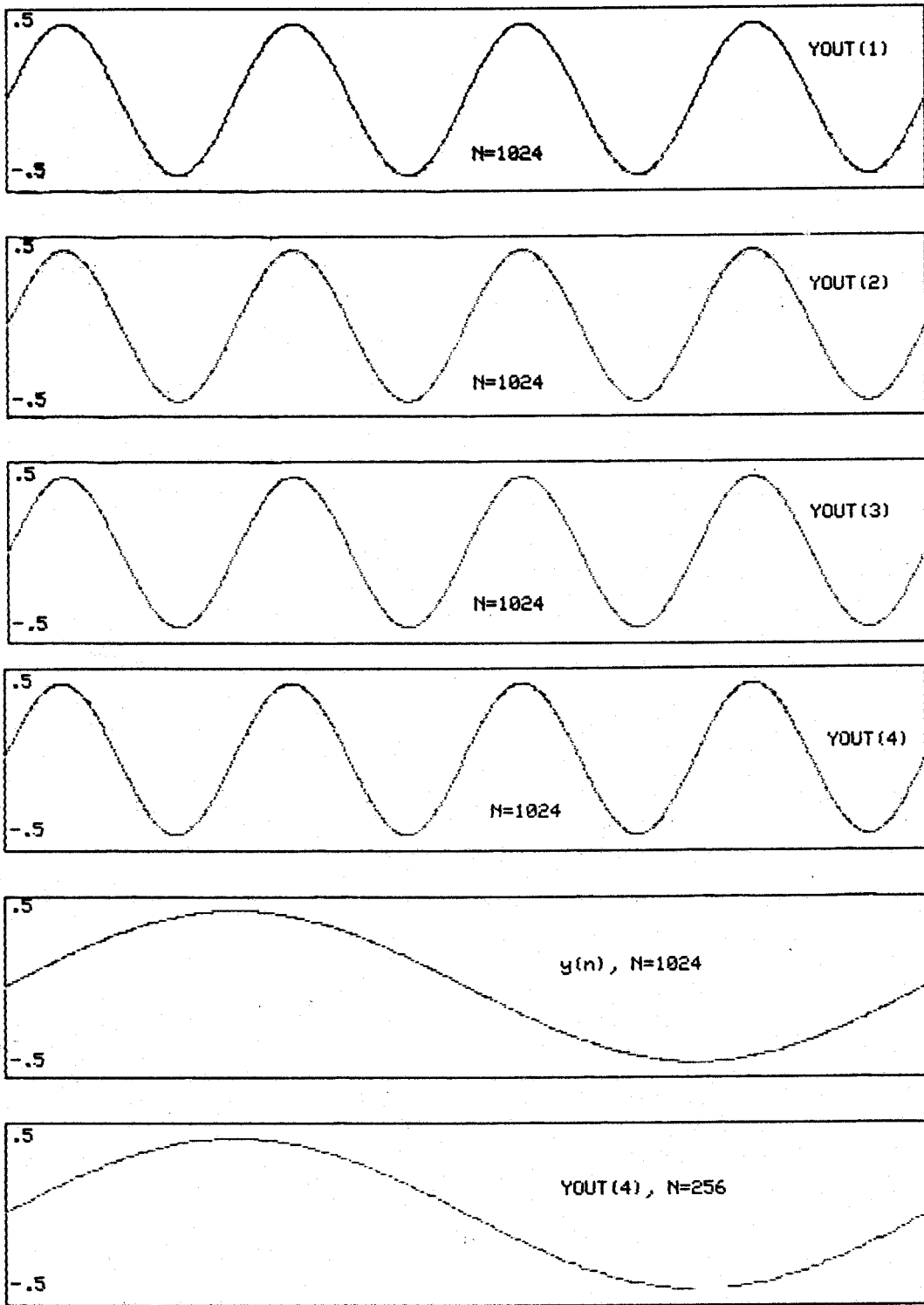


Figura IV.11

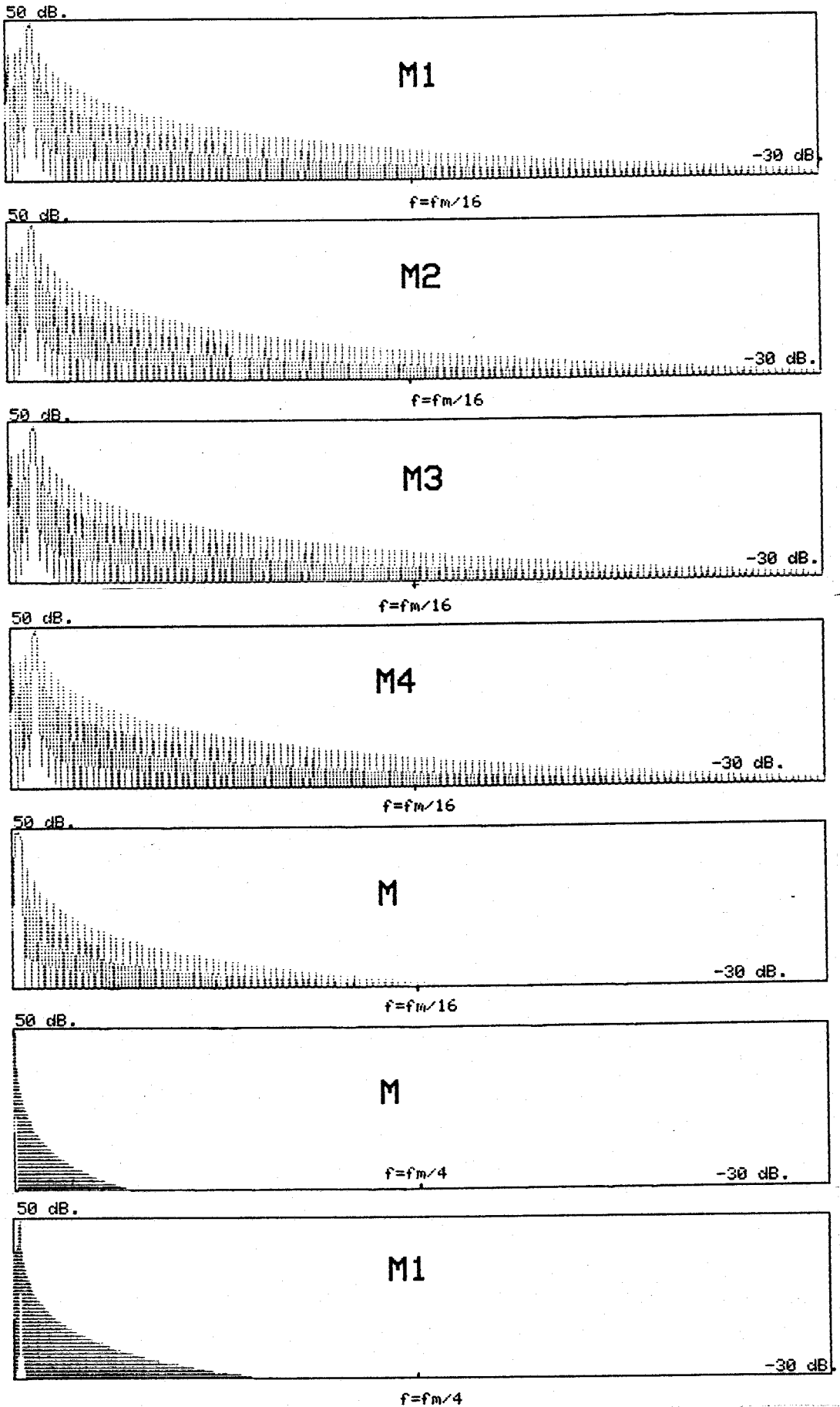


Figura IV.12

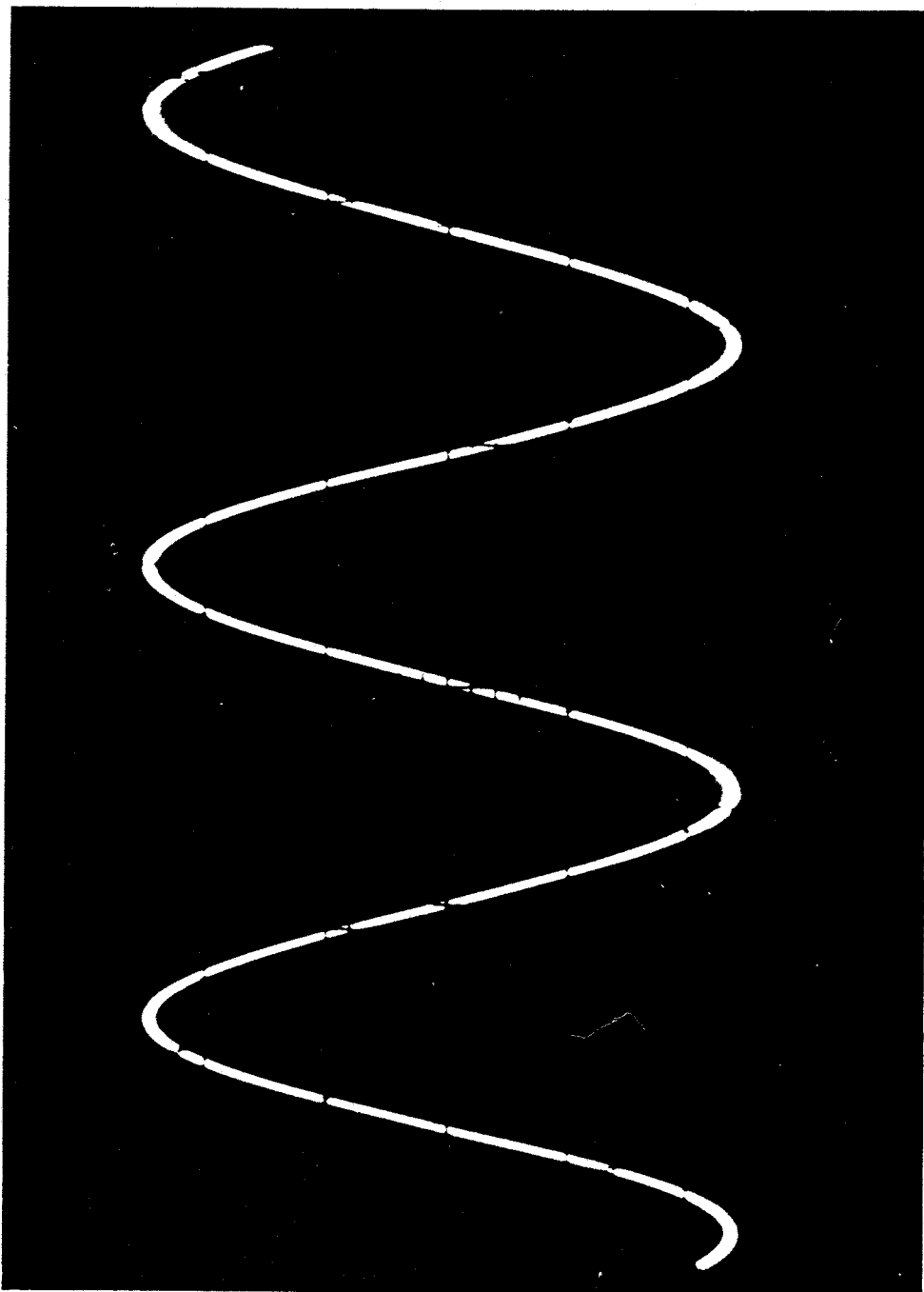


Figura IV.13

```

C *****
C *
C *   PROGRAMA :   OSC
C *
C *   Este programa simula el funcionamiento de un
C *   oscilador de bloques, L=4, N=1024
C *
C *   AUTOR :      A. Ayerbe Garcia
C *
C *               C.T.N.E.
C *               DPTO. DE INVESTIGACION Y DESARROLLO
C *
C *   SECCION :   Investigacion Aplicada
C *
C *   Fecha de ultima revision : 12-Nov-1985
C *
C *****

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C

```

```

PROGRAM OSC

```

```

C-----1-----2-----3-----4-----5-----6-----7-
C

```

```

DIMENSION Z4(20),COEF(20)
DIMENSION X1P(2),X2P(2),YOUTP(20),SA(30000)
CHARACTER*10, NOMBRE

```

```

C Lectura de datos de entrada: L, N, IB
C

```

```

TYPE 100

```

```

100  FORMAT(' ****OSCILADOR DE BLOQUES-1 ****'///)
      L=4
      N=1024
      PI=4.*ATAN(1.0)

```

```

C Calculo de los parametros, FREC
C

```

```

FREC=(2.*PI/N)*L
B=-2.*(1.-COS(FREC))
A1=1./SQRT(ABS(B))

```

```

C Calculo del numero de bits para la parte entera del acumulador:IAC+1
C

```

```

AC=ALOG10(A1)/ALOG10(2.)
IAC=ICEIL(AC)
TYPE*, ' IAC=', IAC
TYPE*, ' B=', B

```

```

C Calculo de los coeficientes multiplicadores
C

```

```

COEF1=2.**IAC
COEF2=2.**(-IAC)
COEF3=B*2.**(2*IAC)
IF (1.EQ.1) GO TO 25
DO I=2,L
      COEF(I)=B*(SIN((I-1)*FREC/L))*COEF1/SIN(FREC)
END DO
TYPE*, ' COEF3=', COEF3*COEF2
DO I=2,L
TYPE*, ' COEF( ', I, ' )=', COEF(I)
END DO

```

C
C Calculo de las condiciones iniciales

C
25 X1P(1)=SIN(FREC)/B
X2P(1)=0.
TYPE*, 'Condiciones iniciales'
TYPE*, 'X1(0)=', X1P(1)
TYPE*, 'X2(0)=', X2P(1)

C
30 TYPE 500
500 FORMAT(
* ' Desea almacenar en fichero las muestras de la senal de'
* ' salida? (Si=1, No=0): ', \$)
READ(5, *, ERR=40) IQ
IF(IQ.LT.0 .OR. IQ.GT.1) THEN
40 CALL LECT3(*30)
END IF
IF(IQ.NE.1) GO TO 60
TYPE*, ' Escriba el nombre del fichero: '
ACCEPT 50, NOMBRE
50 FORMAT(A10)
OPEN(UNIT=2, FILE=NOMBRE, FORM='UNFORMATTED', STATUS='NEW')
WRITE(2) 2*N

C
C Calculo de las salidas

C
60 DO J=1, 3*N/L

C
C a) En maxima precision

C
Z1P=COEF2*X1P(1)
Z3P=COEF3*Z1P
Z2P=COEF2*X2P(1)
YOUTP(1)=Z2P
SA((J-1)*L+1)=YOUTP(1)
IF (L.EQ.1) GO TO 65
DO I=2, L
Z4(I)=Z1P*COEF(I)
YOUTP(I)=Z4(I)+Z2P
SA((J-1)*L+I)=YOUTP(I)
END DO
65 TYPE*, 'Z1P=', Z1P
TYPE*, 'Z2P=', Z2P
TYPE*, 'Z3P=', Z3P
TYPE*, 'YOUT=', YOUTP(1)
IF (L.EQ.1) GO TO 70
DO I=2, L
TYPE*, 'Z4(', I, ')=', Z4(I)
TYPE*, 'YOUT(', I, ')=', YOUTP(I)
END DO

C
C Calculo de las nuevas condiciones iniciales

C
C
C En maxima precision
C
70 X1P(2)=X1P(1)+COEF2*Z3P+Z2P
X2P(2)=Z3P+X2P(1)
X1P(1)=X1P(2)
X2P(1)=X2P(2)
TYPE*, 'X1(1)=', X1P(1)
TYPE*, 'X2(1)=', X2P(1)
END DO

C
C Formacion del fichero de salida
C

```
IF(IQ.EQ.1) THEN
WRITE(2) (SA(I),I=1,2*N)
CLOSE(2)
END IF
STOP
END
```

```
C
C-----1-----2-----3-----4-----5-----6-----7-
      INTEGER FUNCTION ICEIL(X)
C-----1-----2-----3-----4-----5-----6-----7-
C
```

```
IF(X.EQ.0.) THEN
      ICEIL=X
      GO TO 10
END IF
IF(ABS(X-INT(X)).LT.1.E-12) THEN
      ICEIL=INT(X)
      GO TO 10
END IF
IF(ABS(X-INT(X)).GT.(1.-1.E-12)) THEN
      A=ABS(X)+1.
      ICEIL=INT(A)
      IF(ICEIL.LT.0) THEN
            ICEIL=-ICEIL
      END IF
      GO TO 10
END IF
IF(X.GT.0.) THEN
      ICEIL=INT(X+1)
      GO TO 10
END IF
IF(X.LT.0.) THEN
      ICEIL=-INT(ABS(X))
END IF
10  RETURN
END
```

```
C
C
C-----1-----2-----3-----4-----5-----6-----7-
      SUBROUTINE LECT3(*)
C-----1-----2-----3-----4-----5-----6-----7-
C
```

```
TYPE*, ' Error en la entrada, introduzca dato de nuevo'
RETURN 1
END
```

*-----1-----2-----3-----4-----5-----6

* PROGRAMA : OSCILADOR DE BLOQUES L=4

* AUTOR: A. AYERBE GARCIA

* FECHA: 12-11-85

* Sec. : INVESTIGACION APLICADA

* Dept.: INVESTIGACION Y DESARROLLO

* TELEFONICA

*-----1-----2-----3-----4-----5-----6

* Descripcion :

* =====

* Este programa permite la realizacion de un
* oscilador de bloques, de orden 4, de frecuencia
* (Fm/N), con N=1024.

*-----1-----2-----3-----4-----5-----6

* Codificacion

* IDT 'OSC'

*
* Declaracion de literales

L EQU 4

S EQU 6

*..... Definicion de la DATA RAM.....

DSEG

NM BSS 1

UNO BSS 1

TRES BSS 1

COFB BSS 1

COF2 BSS 1

COF3 BSS 1

COF4 BSS 1

V1 BSS 1 Variables de estado, formato Q14

V2 BSS 1

Y1 BSS 1 Salidas del oscilador, formato Q14

Y2 BSS 1

Y3 BSS 1

Y4 BSS 1

AUX BSS 1

AUX1 BSS 1

AUX2 BSS 1

DEND

*
* PSEG

B OSC

RET

*.....Lectura de los coeficientes de un archivo

COEF DATA -1263,-316,-632,-947,-10430,0

*.....Nucleo del programa

DEF OSC

OSC EQU \$

*.....Inicializacion

LACK 1

SACL UNO,0
LACK 3
SACL TRES

*
*....Carga de los coeficientes en DATA RAM
*

LACK L
ADD UNO
SACL AUX
LT UNO
MPYK COEF
CALL CARGA

*
* Calculo de las salidas
*

BEGIN LAC V2
SACL Y1
LT V1
MPY COF2
PAC
ADD UNO,14
ADD V2,15
SACH Y2,1
MPY COF3
PAC
ADD UNO,14
ADD V2,15
SACH Y3,1
MPY COF4
PAC
ADD UNO,14
ADD V2,15
SACH Y4,1

*
* Hay cuatro salidas
*

OUT Y1,PA2
OUT Y2,PA2
OUT Y3,PA2
OUT Y4,PA2
LAC NM
ADD UNO
SACL NM

*
* Calculo de las condiciones iniciales
*

LT V1
MPY COFB
PAC
ADD UNO,14
SACH AUX2,1
ZALH V1
ADD AUX2,10
ADD V2,10
SACH AUX1
LAC V2,15
ADD AUX2,15
SACH V2,1
LAC AUX1
SACL V1
B BEGIN

*
*.....Subrutina de lectura de coeficientes
*

CARGA ZAC
PAC

	LAR	ARO,AUX
	LARK	AR1,COFB
BUCLE	LARP	1
	TBLR	*,ARO
	ADD	UNO
	BANZ	BUCLE
	ZAC	
	RET	
	PEND	

*-----1-----2-----3-----4-----5-----6

* PROGRAMA : OSCILADOR DE BLOQUES L=4

* AUTOR: A. AYERBE GARCIA

* FECHA: 12-11-85

* Sec. : INVESTIGACION APLICADA

* Dept.: INVESTIGACION Y DESARROLLO

* TELEFONICA

*-----1-----2-----3-----4-----5-----6

* Descripcion :

* =====

* Este programa permite la realizacion de un
* oscilador de bloques, de orden 4, de frecuencia
* (Fm/N), con N=1024.

*-----1-----2-----3-----4-----5-----6

* Codificacion

* IDT 'OSCIL'

*
* Declaracion de literales

L EQU 4

S EQU 6

*..... Definicion de la DATA RAM.....

DSEG

DATO BSS 1 Variable para la AIB

UNO BSS 1

COFB BSS 1

COF2 BSS 1

COF3 BSS 1

COF4 BSS 1

V1 BSS 1 Variables de estado Q14

V2 BSS 1

Y1 BSS 1 Salidas del oscilador Q14

Y2 BSS 1

Y3 BSS 1

Y4 BSS 1

AUX BSS 1

AUX1 BSS 1

AUX2 BSS 1

DEND

*
* PSEG

B OSCIL

RET

* Definicion del modo AIB y de la frec. muestreo

RATE DATA 144

MODE DATA >FA

*.....Lectura de los coeficientes de un archivo

COEF DATA -1263,-316,-632,-947,-10430,0

*.....Nucleo del programa

DEF OSCIL

OSCIL EQU \$

*
*.....Inicializacion

*
LACK 1
SACL UNO,0
LACK RATE
TBLR DATO
OUT DATO,PA1
LACK MODE
TBLR DATO
OUT DATO,PA0

*
*....Carga de los coeficientes en DATA RAM

*
LACK L
ADD UNO
SACL AUX
LT UNO
MPY COEF
CALL CARGA

*
* Calculo de las salidas

*
ESPERA BIOZ BEGIN
B ESPERA
BEGIN LAC V2
SACL Y1
LT V1
MPY COF2
PAC
ADD UNO,14
ADD V2,15
SACH Y2,1
MPY COF3
PAC
ADD UNO,14
ADD V2,15
SACH Y3,1
MPY COF4
PAC
ADD UNO,14
ADD V2,15
SACH Y4,1

*
* Hay cuatro salidas

*
OUT Y1,PA2
*
OUT Y2,PA2
*
OUT Y3,PA2
*
OUT Y4,PA2
*

*
* Calculo de las condiciones iniciales

*
LT V1
MPY COFB
PAC
ADD UNO,14
SACH AUX2,1
ZALH V1
ADD AUX2,10
ADD V2,10
SACH AUX1
LAC V2,15
ADD AUX2,15
SACH V2,1
LAC AUX1

SACL V1
B ESPERA

*
*.....Subrutina de lectura de coeficientes
*

CARGA	ZAC	
	PAC	
	LAR	ARO,AUX
	LARK	ARI,COFB
BUCLE	LARP	1
	TBLR	++,ARO
	ADD	UNO
	BANZ	BUCLE
	ZAC	
	RET	
	PEND	

CONCLUSIONES.

1. Después de analizar la importancia relevante del procesado de bloques en la actualidad, se ha hecho una revisión general de las formas más importantes de descripción de estructuras de bloques.

2. Se ha abordado, a continuación, el problema de análisis de la sensibilidad de las estructuras realizadas por bloques a la cuantización de los coeficientes, independientemente de la forma de descripción elegida para las mismas, haciendo uso de la relación entre la función de transferencia escalar y la matriz de transferencia de bloques.

3. Puesto que para estructuras escalares las cascadas de secciones de segundo orden presentan baja sensibilidad respecto a la cuantización de los coeficientes, se particulariza el estudio anterior de bloques a estructuras de segundo orden y se deduce una expresión para la sensibilidad de primer orden del módulo de la respuesta en frecuencia.

4. Se define una cota para dicha función de sensibilidad y se utiliza la misma para estudiar la influencia del orden de bloques sobre la sensibilidad de la magnitud de la respuesta en frecuencia con la cuantización de los coeficientes. Se comparan, así, para distintos valores de L , las secciones de segundo orden obtenidas en diseños de filtros digitales TAN paso bajo de orden 6, según las cuatro aproximaciones clásicas.

5. Se pone de manifiesto con lo anterior que para bandas pasantes estrechas las estructuras presentan una fuerte degradación con la cuantización de los coeficientes, que disminuye al aumentar el orden de los bloques.

6. Conforme aumenta el ancho de banda la influencia del orden de los bloques sobre la sensibilidad disminuye y se estabiliza, pero en ningún caso aumenta.

7. Se comprueba la validez de las medidas anteriores, así como de los resultados, estudiando por simulación los efectos de la cuantización de los coeficientes sobre el módulo de la respuesta en frecuencia de la estructura realizada por conexión en cascada de bloques de segundo orden. Asimismo, se da una medida experimental de dicha sensibilidad.

8. Se propone a continuación la realización de un oscilador de baja frecuencia (frecuencia pequeña comparada con la de muestreo), empleando una estructura de bloques de sensibilidad mínima y se compara con los osciladores de bloques directo y acoplado.

9. Se realiza un estudio completo de las estructuras de los osciladores de bloques anteriores, en lo que se refiere a escalado y a ruido de redondeo de las operaciones de multiplicación, dando una medida teórica del mismo.

10. A continuación, se efectúa una medida experimental por simulación del redondeo después de la multiplicación y se comparan los resultados obtenidos con las medidas teóricas para los tres tipos de osciladores.

Esto pone de manifiesto, en primer lugar, la validez de los modelos presentados y en segundo lugar, que la estructura de bloques propuesta presenta mejor comportamiento en cuanto a ruido que las clásicas.

11. Se compara el oscilador de bloques propuesto con su correspondiente escalar, resaltándose las ventajas computacionales obtenidas con la realización por bloques.

12. Se presenta, por último, la realización del oscilador de bloques de baja frecuencia ($f_0 \ll f_m$) ya descrito sobre el procesador de señal TMS32010.

Después de hacer una descripción del mismo, así como del "software" y "hardware" adicional disponible, se describe un método general de realización de algoritmos sobre el mismo.



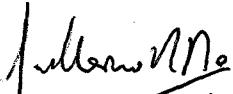
13. Se diseña un oscilador concreto de baja frecuencia comparada con la de muestreo sobre el procesador TMS32010, describiéndose las características del mismo y comprobando mediante simulación y emulación los resultados del estudio anterior.

UNIVERSIDAD DE SEVILLA

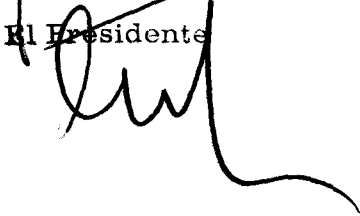
Reunido el Tribunal integrado por los abajo firmantes en el día de la fecha, para juzgar la Tesis Doctoral de D. ARANZAZU AYERBE GARCIA titulada Una aportación al estudio de filtros y osciladores digitales de bloques. acordó otorgarle la calificación de APTO CUM LAUDE

Sevilla, 29 de Mayo 1987

El Vocál,



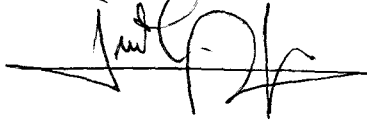
El Presidente



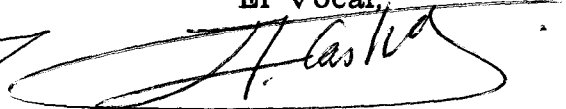
El Vocal,



El Secretario,



El Vocal,



El Doctorado,

