ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

TESIS DE GRADO

PREVIA LA OBTENCION DEL TITULO DE INGENIERO
EN ELECTRONICA Y TELECOMUNICACIONES

TEMA: DESARROLLO DEL SOFTWARE PARA LA SIMULACION DE FILTROS DIGITALES RECURSIVOS PARTIENDO DEL DISEÑO DE LOS FILTROS ANALOGICOS DE BUTTERWORTH, CHEVISHEV Y ELIPTICOS.

CARLOS HERNAN SUAREZ LUNA

OCTUBRE DE 1995

CERTIFICACION:

Certifico que, bajo mi dirección, la presente tesis fue realizada en su totalidad por el señor Carlos Hernán Suarez Luna.

Dr. Gualberto Hidalgo
DIRECTOR DE TESIS

DEDICATORIA

A los seres mas preciados de este mundo, que incansablemente me han dado todo.

Dios con su bondad infinita me envió a estos dos ángeles para que cuidaran de mí.

Amor, admiración y respeto para ellos. Mis Padres.

CARLOS

AGRADECIMIENTO

Al Dr. Gualberto Hidalgo por la dirección y apoyo prestados en la realización de este trabajo.

CARLOS

INDICE

INTRODUCCION

El tratamiento de las señales digitales, en los tiempos actuales, es un tópico de mucha importancia.

El campo de lo analógico, se convierte en digital para ser tratado con mayor cuidado y con muchas ventajas.

Este trabajo se dedica, a una parte relevante de lo que constituye el tratamiento digital de señales, los filtros digitales.

Los tres primeros capítulos de este trabajo, son dedicados a realizar el desarrollo matemático para la simulación del filtro :

En el primer capítulo se desarrolla el estudio de los filtros analógicos pasa-bajos de Butterworth, Chevishev tipo I, Chevishev tipo II, y Elípticos. De estos, se analiza en forma mas profunda el desarrollo de los filtros Elípticos debido a su complejidad. El diseño de estos filtros termina al determinar la función de transferencia.

El segundo capítulo se centra en el estudio de la digitalización de los filtros analógicos pasa-bajos. Los métodos de conversión que se analizan son: Invariancia de Impulso, Solución Numérica de Ecuaciones Diferenciales, Transformación Bilineal, Invariancia de Pulso. El último método es nuevo, razón por la cual no está muy bien difundido. Con este desarrollo pretendemos introducir una nueva herramienta de trabajo en el tratamiento de señales.

En el tercer capítulo, se analizan los métodos de transformación de banda de un filtro pasa-bajos. De esta manera podemos encontrar la función de transferencia de filtros digitales pasa-bajos, pasa-altos, pasa-banda y elimina banda.

En el cuarto capítulo se desarrolla el programa que simulará el filtro digital. En base a los conocimientos adquiridos en los tres primeros capítulos, aquí se efectúa el nexo de unión entre lo teórico y lo práctico.

En el quinto capítulo mostramos los resultados obtenidos al realizar estos filtros. Se incluyen comparaciones entre los varios tipos de filtros que se pueden diseñar.

El lenguaje utilizado para este trabajo es el QuickBASIC, sobretodo porque este lenguaje de programación proporciona un buen manejo de gráficos. Para que se pueda utilizar correctamente este programa, hemos escrito un MANUAL DEL USUARIO, donde se indica la forma de operación.

Espero haber dado un valioso aporte en el desarrollo del procesamiento digital de señales.

CAPITULO 1

ANALISIS DE LOS FILTROS ANALOGICOS

Antes de introducirnos en el análisis de los filtros analógicos de Butterworth, Chevishev y Elípticos; discutiremos la terminología que se empleará.

Un filtro analógico puede ser representado por la ecuación:

$$H(s) = \frac{Vo(s)}{Vi(s)}$$

donde Vo(s) y Vi(s) son la transformada de Laplace de las señales de entrada y salida del filtro respectivamente.

H(s) es la función de transferencia que tiene un polinomio en el denominador y un polinomio en el denominador. Esta función debe satisfacer las siguientes condiciones:

1) Debe ser una función racional de "s" con coeficientes reales.
2) Los polos de esta función deben estar en el semi-plano izquierdo del plano s.
3) El grado del polinomio del numerador debe ser menor o igual que el grado del polinomio del denominador.

de esta manera se garantiza la realización del filtro, la estabilidad y causalidad de la red.

- "s" es la frecuencia compleja en el dominio analógico.

$$s = \sigma + j\Omega$$

- "Ω" es la frecuencia angular y es igual a

$$\Omega = 2 \pi f$$

f es la frecuencia de una componente de la señal en el dominio del tiempo. Por tanto las unidades de Ω son [radianes / segundos].

Cuando se diseña un filtro se lo hace a partir de la respuesta de frecuencia que se desea obtener. La respuesta de frecuencia de un filtro caracteriza el grado de atenuación que tendrán las componentes de frecuencia de la señal a filtrar a lo largo de todo el espectro. Por ejemplo un filtro pasa-bajos atenuará con mayor rigor las frecuencias altas antes que las bajas.

Generalmente la respuesta de frecuencia especifica por zonas, la atenuación. Asi tenemos:

Banda de Paso (BP), es aquella zona del espectro donde la atenuación es mínima y se permite el paso de las componentes a esas frecuencias.

Banda de Supresión (BS), es aquella donde la atenuación es máxima e impide el paso de las componentes a esas frecuencias.

Banda de Transición (BT), es aquella donde la magnitud de la respuesta de frecuencia cae suavemente de la banda de paso a la banda de supresión o viceversa, y hace posible la realización física del filtro.

Las bandas están limitadas por las frecuencias angulares Ωp, Ωs y Ωc. Donde Ωp es la frecuencia analógica de corte de la banda de paso, Ωs es la frecuencia analógica de corte de la banda de supresión, y Ωc es la frecuencia de corte a -3dB, es decir, donde la respuesta de frecuencia es igual a $1/\sqrt{2}$.

La figura 1.1 representa la respuesta de frecuencia de un filtro pasa-bajos donde la banda de paso está entre 0 y Ωp, la banda de transición entre Ωp y Ωs, y la banda de supresión desde Ωs hasta el infinito.

En este capítulo nos dedicaremos al diseño de filtros pasa-bajos. A partir de estos filtros podremos obtener cualquier tipo de filtro utilizando las adecuadas transformaciones.

1.1 FILTROS DE BUTTERWORTH

Esta clase de filtros se caracteriza por que la función de transferencia del filtro está dada utilizando polinomios de Butterworth, de ahí su nombre.

[1.1]Los polinomios de Butterworth se especifican de la siguiente manera:

$$\mathbf{L}(\omega^2) = \mathbf{B}_0 + \mathbf{B}_1\omega^2 + \ldots + \mathbf{B}_N\omega^{2N}$$

ahora, si los coeficientes tienen los siguientes valores :

$$\mathbf{B}_1 = \mathbf{B}_N = 1 \qquad \text{y} \qquad \mathbf{B}_1 = \mathbf{B}_2 = \ldots = \mathbf{B}_{N-1} = 0$$

entonces uno de los polinomios de Butterworth sería :

$$\mathbf{L}(\omega^2) = 1 + \omega^{2N}$$

El inverso de esta función viene a ser el módulo al cuadrado de la función de transferencia del un filtro pasa-bajos de Butterworth, así :

$$\left|\mathbf{Ha}(j\Omega)\right|^2 = \frac{1}{1 + (\frac{j\Omega}{j\Omega c})^{2N}} \qquad (1.1)$$

se ha realizado un cambio de variable w = $\Omega/\Omega c$,donde Ωc es la frecuencia de corte a -3dB.

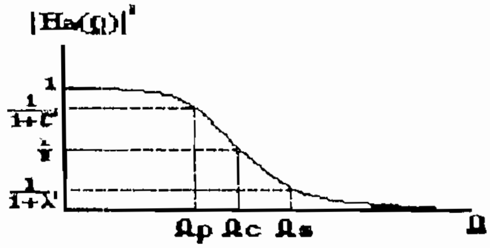


Figura 1.1: Respuesta de frecuencia
de un filtro de Butterworth

Obedeciendo a esta función de transferencia, la respuesta de frecuencia del filtro se muestra monótonamente decreciente. Además no tiene rizado en ninguna de las bandas y es máximamente plano en la banda de paso. Esto lo podemos visualizar en la figura 1.1

Donde:

Ωp es la frecuencia analógica de corte de la banda de paso.

Ωs es la frecuencia analógica de corte de la banda de supresión.

ε es la constante de atenuación de la banda de paso.

λ es la constante de atenuación de la banda de supresión.

De esta manera, con esta respuesta de frecuencia el filtro se hace poco selectivo, porque la zona de transición cae lentamente.

Para un filtro de orden N, el hecho de que sea máximamente plano en la banda de paso, significa que: "Las primeras 2N-1 derivadas de $|\mathbf{Ha}(j\Omega)|^2$ sean iguales a cero en $\Omega=0$ ".

La función de transferencia obtenida a partir de los polinomios de Butterworth, cumple esa condición. Para la demostración asumiremos que $\Omega c=1$, entonces tenemos:

$$|\mathbf{Ha}(\Omega)|^2 = \frac{1}{1+(\Omega)^{2N}} = 1-\Omega^{2N}+\Omega^{4N}-\Omega^{6N}+\ldots$$

En esta expresión hemos expandido en términos de una serie infinita. A partir de esta empezamos a derivar término a término, y obtenemos:

$$\frac{d}{d\Omega}\left(|Ha(\Omega)|^2\right) = -2N\Omega^{2N-1} + 4N\Omega^{4N-1} - \ldots$$

En consecuencia la derivada de orden 2N-1, sería:

$$\frac{d^{2N-1}}{d\Omega^{2N-1}}\left(|Ha(\Omega)|^2\right) = \frac{-2N(2N-1)(2N-2)\ldots.2\Omega}{2} + \ldots$$

hasta aquí se conserva el Ω, y como evaluamos en $\Omega=0$, la 2N-1 derivada de la expresión resulta ser cero. Si derivamos una vez mas se elimina la variable Ω quedando un valor constante distinto de cero.

[1.2]

1.1.1 OBTENCION DE LOS POLOS DE UN FILTRO DE BUTTERWORTH

Como se observa estos polinomios de Butterworth pueden ser descompuestos en 2N factores. Estos factores se encuentran en el denominador de la función de transferencia, y se convierten en polos del plano complejo S.

Para encontrar los polos de un filtro de Butterworth, simplemente debemos igualar a cero el denominador de la expresión del módulo al cuadrado de $Ha(j\Omega)$, asi:

$$|Ha(j\Omega)|^2 = \frac{1}{1+\left(\frac{S}{j\Omega c}\right)^{2N}} \quad ; \quad \text{donde } S=j\Omega.$$

entonces: $1+\left(\frac{S}{j\Omega c}\right)^{2N} = 0$

despejando "s" tenemos:

$$S = (-1)^{1/2N} j\Omega c$$

reemplazando en esta expresión las siguientes equivalencias:

$$(-1) = e^{j\pi(2p-1)}$$

$$j = e^{j\frac{\pi}{2}\cdot\frac{N}{N}}$$

tenemos:

$$Sp = \Omega c . e^{j\pi\left(\frac{2p+N-1}{2N}\right)}$$; donde: p=1,2,3....,2N

De esta expresión podemos concluir que:

- **Los polos se ubican simétricamente respecto del eje imaginario.**

  Consideremos a:

$$\theta(p) = \frac{(2p+N-1)}{2N}\pi$$;como el argumento de Sp.

Así tenemos que, el argumento $\theta(p)$toma los siguientes valores:

a) $\frac{\pi}{2} < \theta(p) < \frac{3\pi}{2}$; para $1 < p < N$

b) $\frac{3\pi}{2} < \theta(p) < \frac{5\pi}{2}$; para $N < p < 2N$

En el plano complejo S, el ángulo $\theta(p)$ se toma respecto al semi-eje positivo real. Por lo tanto para el caso a) los puntos que representan a los polos se encuentran al

lado izquierdo del eje imaginario, mientras que para el caso b) los polos se sitúan al lado derecho.
Entonces podemos estar seguros que, la mitad de polos se encuentran a la izquierda del eje imaginario y la otra mitad a la derecha.

Para lograr la estabilidad de un sistema se eliminan los polos que se encuentran a la derecha del eje imaginario.

Significa que se tomará solo los polos donde $1 \leq p \leq N$, para formar la función de transferencia del filtro.

PLANO COMPLEJO "S"

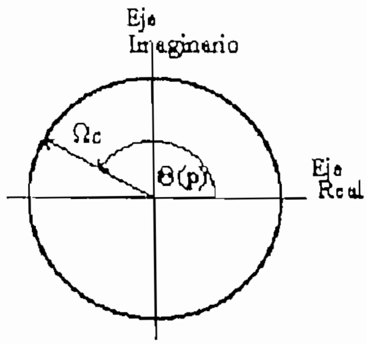


Figura 1.2: Indicación del Plano S

Con la variación arriba indicada para "p" se tiene que:

- **Ningún polo cae sobre el eje imaginario, ni para N par ni para N impar.**
  Necesariamente para que algún polo caiga sobre el eje imaginario, se necesita que el argumento $\theta(p)$ sea igual a $p/2 \pm mp$, donde $m=0,1,2,3,...$, los valores mas cercanos de "p" para esta posibilidad serían: $p = 1, N, N+1$ ó $2N$.

Entonces reemplazando en $\theta(p)$, tenemos:

$$p = 1 \quad , \quad \theta(p) = \frac{N+1}{2N}\pi$$

$$p = N \quad , \quad \theta(p) = \frac{3N-1}{2N}\pi$$

$$p = N+1 \quad , \quad \theta(p) = \frac{3N+1}{2N}\pi$$

$$p = 2N \quad , \quad \theta(p) = \frac{5N-1}{2N}\pi$$

Como se puede apreciar, siempre existe en el numerador un número ± 1, que evita que el argumento sea un múltiplo de $\pi/2$.

- **Para N impar tenemos dos polos sobre el eje real.** Consideremos $N=2k+1$, donde k es un número entero, entonces si un polo cayera en el eje real, necesariamente el argumento $\theta(p)$ debería ser un múltiplo de π. Ahora reemplazando, tenemos:

$$\theta(p) = \frac{2p+(2k+1)-1}{2(2k+1)}\pi = \pi$$

simplificando esta expresión, vemos que se cumple :

$$p = k + 1$$

dado que k y p son enteros, no existe contradicción.

- **Los polos estan equi-espaciados alrededor de un círculo de radio Ωc.** Esto se justifica porque el módulo de Sp efectivamente es Ωc, y además el ángulo experimenta incrementos iguales.

Ejemplo:

Sea N=3 (impar)

p=1,2,3..,6

$$Sp = \Omega c.e^{j\pi\left(\frac{2p+3-1}{6}\right)}$$

$$S_1 = \Omega c.e^{j\frac{2\pi}{3}}$$
$$S_2 = \Omega c.e^{j\pi}$$
$$S_3 = \Omega c.e^{j\frac{4\pi}{3}}$$
$$S_4 = \Omega c.e^{j\frac{5\pi}{3}}$$
$$S_5 = \Omega c.e^{j2\pi}$$
$$S_6 = \Omega c.e^{j\frac{7\pi}{3}}$$

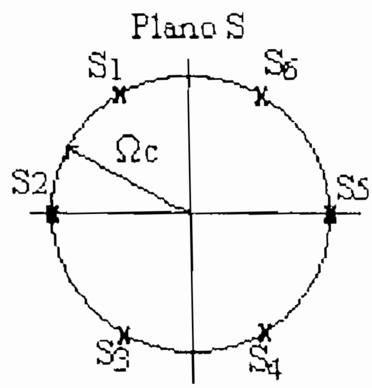


Figura 1.3: Ubicación de los polos

1.1.2 DISEÑO DEL FILTRO DE BUTTERWORTH.

A partir de las especificaciones dadas para la respuesta de frecuencia, se realiza el diseño del filtro, que consiste en determinar la función de transferencia Ha(S).

La respuesta de frecuencia de un filtro viene especificada por el valor de atenuación máxima en la banda de paso A[dB], el valor de atenuación mínima en la banda de supresión B[dB]. De hecho estos valores corresponden a las frecuencias angulares Ωp y Ωs respectivamente.

Para Ωp atenuación < A [dB];

Para Ωs atenuación > B [dB];

A y B son números reales positivos.

Usualmente el valor de las atenuaciones se da en decibelios y el de la frecuencia angular en radianes/segundo. Debemos indicar además que los decibelios se dan en valor positivo, entendiéndose que por ser atenuación en la deducción de las fórmulas correspondientes se ha considerado ya el valor negativo.

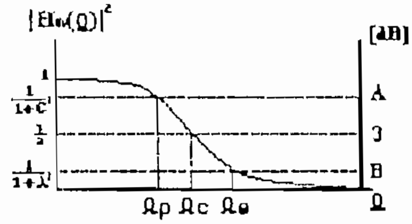


Figura 1.4: Especificaciones en decibelios

En esta figura, se observa la respuesta de frecuencia del filtro de Butterworth, donde el eje de la izquierda especifica la atenuación adimensionalmente, en cambio en el eje de la derecha se especifica en decibelios.

De hecho, para cumplir con esas especificaciones la función de transferencia cumple:

$$|Ha(\Omega)|^2 = \frac{1}{1+\varepsilon^2(\frac{\Omega}{\Omega p})^{2N}}$$

es correcto, porque si $\Omega = \Omega p$ se verifica que :

$$|Ha(\Omega)|^2 = \frac{1}{1+\varepsilon^2}$$

Ahora si $\Omega = \Omega s$, obtenemos la siguiente igualdad:

$$\frac{1}{1+\varepsilon^2\left(\frac{\Omega s}{\Omega p}\right)^{2N}}=\frac{1}{1+\lambda^2}$$

a continuación realizaremos simplificaciones de esta expresión hasta despejar el valor de N:

$$\frac{\lambda^2}{\varepsilon^2}=\left(\frac{\Omega s}{\Omega p}\right)^{2N}$$

$$\log\left(\frac{\lambda}{\varepsilon}\right)=N.\log\left(\frac{\Omega s}{\Omega p}\right)$$

$$N=\frac{\log\left(\lambda/\varepsilon\right)}{\log\left(\Omega s/\Omega p\right)} \qquad (1.2)$$

Si el valor de N resulta ser fraccionario se debe aproximar al entero superior.

Para obtener los valores de λ y ε a partir de los valores de atenuación dados en decibelios (A y B), realizamos las correspondientes equivalencias:

$$10\log\left|Ha(\Omega p)\right|^2=-A \qquad\qquad 10\log\left|Ha(\Omega s)\right|^2=-B$$

reemplazando :

$$\left|Ha(\Omega p)\right|^2=\frac{1}{1+\varepsilon^2} \qquad\qquad \left|Ha(\Omega s)\right|^2=\frac{1}{1+\lambda^2}$$

tenemos:

$$10 \log\left(\frac{1}{1+\varepsilon^2}\right) = -A \qquad\qquad 10 \log\left(\frac{1}{1+\lambda^2}\right) = -B$$

$$\log(1+\varepsilon^2) = \frac{A}{10} \qquad\qquad \log(1+\lambda^2) = \frac{B}{10}$$

despejando ε : despejando λ :

$$\varepsilon = \sqrt{10^{A/10}-1} \qquad (1.3) \qquad\qquad \lambda = \sqrt{10^{B/10}-1} \qquad (1.4)$$

Reemplacemos los valores de frecuencia Ωp y Ωs en la función de transferencia (1.1), y encontremos la relación que estos tienen con la frecuencia Ωc :

Para Ωp: Para Ωs:

$$\frac{1}{1+\varepsilon^2} = \frac{1}{1+\left(\frac{\Omega p}{\Omega c}\right)^{2N}} \qquad\qquad \frac{1}{1+\lambda^2} = \frac{1}{1+\left(\frac{\Omega s}{\Omega c}\right)^{2N}}$$

$$\varepsilon^2 = \left(\frac{\Omega p}{\Omega c}\right)^{2N} \qquad\qquad \lambda^2 = \left(\frac{\Omega s}{\Omega c}\right)^{2N}$$

extrayendo la raíz cuadrada en ambos miembros de las expresiones, y despejando Ωc tenemos:

$$\Omega_c = \frac{\Omega_p}{\varepsilon^{\frac{1}{N}}} \qquad (1.5) \qquad\qquad \Omega_c = \frac{\Omega_s}{\lambda^{\frac{1}{N}}} \qquad (1.6)$$

Las expresiones obtenidas son iguales siempre que se reemplace el valor exacto de N (es decir sin la aproximación al entero próximo). Desde luego como no tiene sentido tener un valor de

N fraccionario, la relación entre ellos es:

$$\Omega c_{(\epsilon)} \leq \Omega c_{(\lambda)}$$

donde:

$\Omega c_{(\epsilon)}$ corresponde a la frecuencia obtenida a partir de ϵ
$\Omega c_{(\lambda)}$ corresponde a la frecuencia obtenida a partir de λ

esto se justifica porque el valor de ϵ es menor que la unidad y en cambio el valor de λ es mayor, entonces:

$$N_e \leq N$$

$$\epsilon^{1/N_e} \leq \epsilon^{1/N}$$

$$\frac{1}{\epsilon^{1/N_e}} \geq \frac{1}{\epsilon^{1/N}}$$

donde Ne es el N exacto.

Vemos claramente que el valor de $\Omega c_{(\epsilon)}$ resulta ser menor que el valor Ωc exacto, y:

$$N_e \leq N$$

$$\lambda^{1/N_e} \geq \lambda^{1/N}$$

$$\frac{1}{\lambda^{1/N_e}} \leq \frac{1}{\lambda^{1/N}}$$

el valor de $\Omega c_{(\lambda)}$ es mayor que el valor exacto. Por lo tanto:

$$\Omega c_{(\epsilon)} \leq \Omega c \leq \Omega c_{(\lambda)}$$

convenientemente se utilizarán estos valores.

La respuesta de frecuencia para cada valor de Ωc se muestra en la figura 1.5.

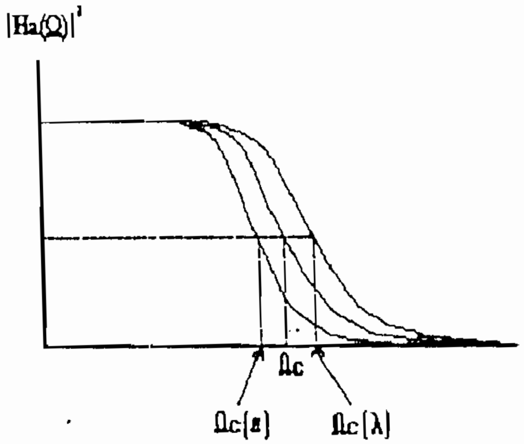


Figura 1.5: Respuesta de Frecuencia para los distintos valores de Ωc

La función de transferencia se la determinará, formando binomios con los polos encontrados para p = 1 ...N, estos los colocamos en el denominador, en el numerador se colocará una constante a determinar haciendo s=0.

$$Ha(s) = \frac{K}{(s-s_1)(s-s_2)\ldots(s-s_N)}$$

donde:

$$K = Ha(0).(-s_1)(-s_2)\ldots(-s_N)$$

y Ha(0) = 1

1.2 FILTROS DE CHEVISHEV TIPO I

En la respuesta de frecuencia de un filtro de Butterworth veíamos que decrece monótonamente la función en la banda de paso. Con el diseño de los filtros de Chevishev se pretende balancear la característica de frecuencia en la banda de paso, logrando tener respuesta con oscilaciones uniformes que van desde el valor 1 hasta el valor atenuado de $1/(1+\varepsilon^2)$.

La función de transferencia del filtro se obtiene a partir de los polinomios de Chevishev.

1.2.1 POLINOMIOS DE CHEVISHEV

Un Polinomio de orden N de Chevishev, está definido por:

$$V_N = \cos[N.\arcos(x)] \tag{1.7}$$

Con la particularidad de que el coseno que interviene es esta expresión, es trigonométrico para valores de $x \leq 1$. Para valores de $x > 1$, el coseno de estos polinomios es hiperbólico. Por esta condición, para $x \leq 1$ la función está dentro de los límites de -1 y 1.

A continuación determinaremos los polinomios de Chevishev de los ordenes inferiores a partir de la expresión (1.7):

a) Para N = 0:

$$V_o(x) = \cos[0.\ \arccos(x)]$$
$$V_o(x) = \cos\ (0°)$$
$$V_o(x) = 1$$

El polinomio de orden cero es una constante igual a 1.

b) Para N = 1:

$$V_1(x) = \cos\ [1.\ \arcos(x)]$$
$$V_1(x) = x$$

El polinomio de primer orden es una recta que parte del origen con pendiente positiva igual a 1.

c) Para N = 2:

$V_2(x) = \cos\,[2.\ \arcos(x)]$

Sea:

$$\cos(\theta) = x \quad \rightarrow \quad \theta = \arcos(x)$$
$$2\theta = 2.\arcos(x)$$
$$\cos(2\theta) = 2.\cos^2(\theta) - 1 \quad ;(\text{Identidad trigonométrica})$$

entonces:

$$\cos[2.\arcos(x)] = 2x^2 - 1$$

por lo tanto:

$$V_2(x) = 2x^2 - 1$$

El polinomio de Chevishev de segundo orden es una parábola, que parte desde el punto (0,-1).

d) Para cualquier orden N:

Utilizando identidades trigonométricas se puede demostrar la siguiente fórmula de recurrencia, con la cual se podrían calcular los polinomios de Chevishev de cualquier orden.

$$V_{N+1}(x) = 2x.V_N(x) - V_{N-1}(x) \qquad (1.8)$$

Demostración:

Sea:

$$V_{N-1}(x) = \cos[(N-1).\arcos(x)]$$
$$V_{N-1}(x) = \cos[N.\arcos(x) - \arcos(x)]$$
$$V_{N-1}(x) = \cos[N.\arcos(x)].\cos[\arcos(x)] + $$
$$+ \operatorname{sen}[N.\arcos(x)].\operatorname{sen}[\arcos(x)]$$
$$V_{N-1}(x) = V_N(x).x + \operatorname{sen}[N.\arcos(x)].\operatorname{sen}[\arcos(x)]$$

Despejando el producto de los senos tenemos:

$$\operatorname{sen}[N.\arcos(x)].\operatorname{sen}[\arcos(x)] = V_{N-1}(x) - V_N(x).x \qquad (1.9)$$

Ahora desarrollemos el término $V_{N+1}(x)$

$$V_{N+1}(x) = \cos[(N+1).\text{arcos}(x)]$$
$$V_{N+1}(x) = \cos[N.\text{arcos}(x) + \text{arcos}(x)]$$
$$V_{N+1}(x) = \cos[N.\text{arcos}(x)].\cos[\text{arcos}(x)] - $$
$$+ \text{sen}[N.\text{arcos}(x)].\text{sen}[\text{arcos}(x)]$$
$$V_{N+1}(x) = V_N(x).x - \text{sen}[N.\text{arcos}(x)].\text{sen}[\text{arcos}(x)]$$

Reemplazando aquí el valor de (1.9), tenemos:

$$V_{N+1}(x) = V_N(x).x - [\ V_{N-1}(x) - V_N(x).x\ ]$$

Por lo tanto:

$$V_{N+1}(x) = 2x.V_N(x) - V_{N-1}(x) \qquad (1.10)$$

Utilizando esta fórmula por ejemplo, determinemos el polinomio de tercer orden:

$N + 1 = 3$
$N = 2$
$N - 1 = 1$

$$V_3(x) = 2x.V_2(x) - V_1(x)$$
$$V_3(x) = 2x.(2x^2-1) - x$$
$$V_3(x) = 4x^3 - 2x - x$$
$$V_3(x) = 4x^3 - 3x$$

Este polinomio de orden 3 tampoco se sale de los límites de -1 y 1. Esto lo veremos en la figura 1.6.

Ahora si a los polinomios de Chevishev los elevamos al cuadrado tendremos funciones limitadas entre 0 y 1. Esto lo podemos visualizar en la figura 1.7. Además vemos que cuando el orden N es par empieza en 1 y si N es impar empezamos en 0. Esta observación es muy importante, y servirá cuando diseñemos el filtro.

Justamente por la forma que estos polinomios tienen para valores de $x \leq 1$, la respuesta de frecuencia del filtro tendrá un equirrizado en la banda de paso.

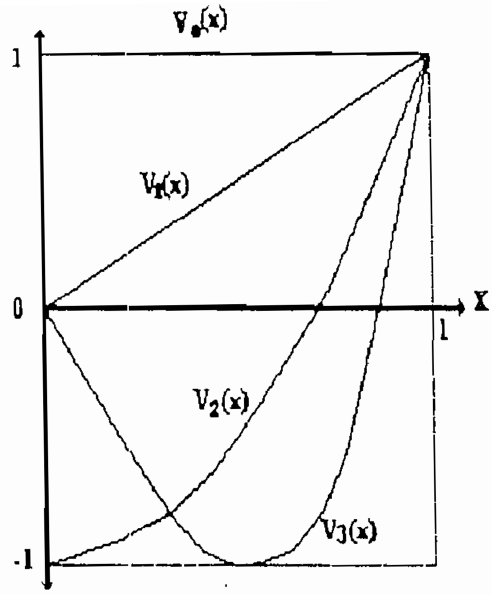


Figura 1.6: Polinomios de Chevishev
de orden 0,1,2 y 3

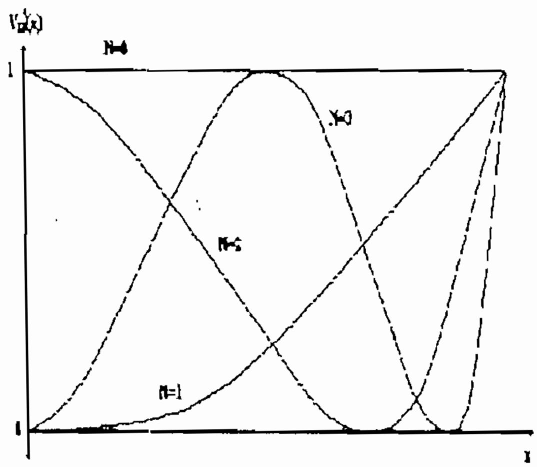


Figura 1.7: Polinomios de Chevishev
elevados al cuadrado

1.2.2 LA FUNCION DE TRANSFERENCIA

El módulo al cuadrado de la función de transferencia del filtro pasa-bajos de Chevishev está definido de la siguiente manera:

$$|Ha(\Omega)|^2 = \frac{1}{1 + \varepsilon^2 V_N^2\left(\frac{\Omega}{\Omega_p}\right)} \qquad (1.11)$$

La variable de los polinomios de Chevishev, ahora es $\Omega/\Omega p$, esto quiere decir que el comportamiento trigonométrico de los polinomios de Chevishev en la función, se mantendrá mientras:

$$\Omega/\Omega p \leq 1 \quad \rightarrow \quad \Omega \leq \Omega p$$

Entonces la respuesta de frecuencia tiene un rizado en la banda de paso hasta Ωp, a partir de esta frecuencia la función decrece y lo hace monótonamente ya que para valores > 1 el argumento de los polinomios de Chevishev es hiperbólico.
Los límites del rizado son: El superior en 1 y el inferior en:

$$|Ha(\Omega p)|^2 = \frac{1}{1 + \varepsilon^2}$$

Para mayor claridad presentamos la respuesta de frecuencia de un filtro de Chevishev, para N par y N impar, en la figura 1.8

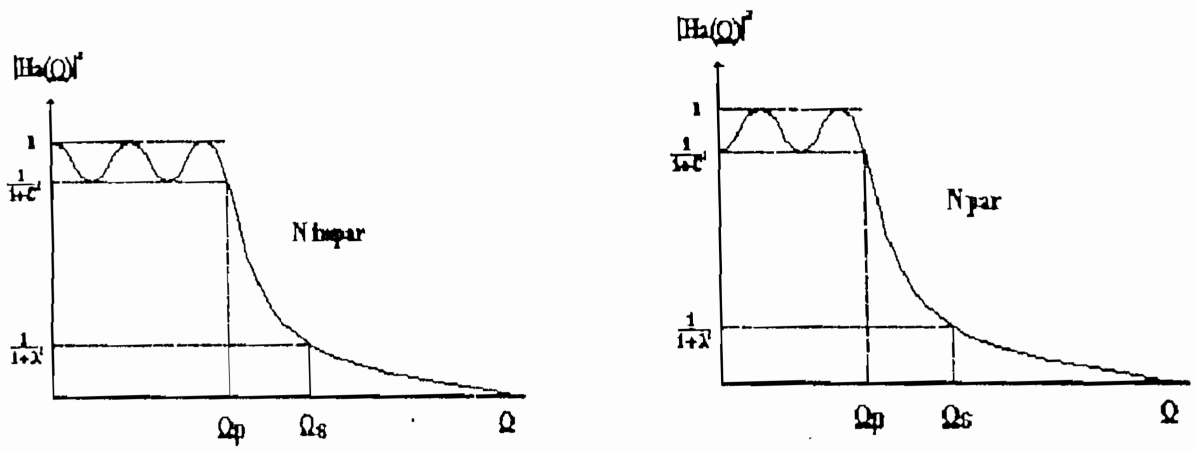


Figura 1.8: Respuestas de Frecuencia de un Filtro Chevishev tipo I

De esto podemos concluir que en $\Omega = 0$ tenemos:

Para N par

$$|Ha(0)|^2 = \frac{1}{1+\varepsilon^2}$$

Para N impar

$$|Ha(0)|^2 = 1$$

También podemos decir que Ωp es la frecuencia a la que

$|Ha(\Omega)|^2$, pasa por última vez por: $\frac{1}{1+\varepsilon^2}$

Con esta respuesta de frecuencia decimos también que este filtro es medianamente selectivo.

1.2.3 OBTENCION DE LOS POLOS DE UN FILTRO DE CHEVISHEV

Al igual que los filtros de Butterworth, los filtros de Chevishev tiene solo polos. La expresión 1.11, indica que solo el denominador puede ser igual a cero, así:

$$1+\varepsilon^2 V_N^2\left(\frac{\Omega}{\Omega_p}\right) = 0 \qquad (1.12)$$

Necesariamente para determinar un polo, los polinomios de Chevishev deben estar en el rango para el cual el coseno es hiperbólico, porque cuando el coseno es trigonométrico, al ser elevado al cuadrado nos proporciona un número positivo que nunca hará cero a la expresión 1.12.

$$1+\varepsilon^2 \left(\cos\left[N.\,arcos\left(\frac{s}{j\Omega_p}\right)\right]\right)^2 \geq 1$$

Para la deducción de la fórmula para el cálculo de los polos, comenzaremos haciendo algunas consideraciones previas:

$$s = \sigma + j\Omega$$

dividimos esta igualdad para $j\Omega p$:

$$\frac{s}{j\Omega p} = \frac{\sigma+j\Omega}{j\Omega p}$$

Sea : $\operatorname{arcosh}(s/j\Omega p) = u + jv$, en consecuencia tenemos:

$$\cosh(u+jv) = \frac{s}{j\Omega p} = \frac{\sigma+j\Omega}{j\Omega p}$$

$$\sigma+j\Omega = j\Omega p \cosh(u+jv) \qquad (1.13)$$

Por otra parte, desarrollando el coseno de la suma:

$$\cosh(u+jv) = \cosh(u).\cosh(jv) - \operatorname{senh}(u).\operatorname{senh}(jv) \qquad (1.14)$$

reemplacemos las siguientes identidades en 1.14

$$\cosh(jv) = \cos(v)$$
$$\operatorname{senh}(jv) = j.\operatorname{sen}(v)$$

entonces:

$$\cosh(u+jv) = \cosh(u).\cos(v) - \operatorname{senh}(u).j\operatorname{sen}(v)$$

reemplacemos esta expresión en 1.13:

$$\sigma + j\Omega = j\Omega p.[\cosh(u).\cos(v) - \operatorname{senh}(u).j\operatorname{sen}(v)] \qquad (1.15)$$

entonces:

$$\sigma = -\Omega p.\operatorname{senh}(u).\operatorname{sen}(v) \qquad (1.16)$$
$$\Omega = \Omega p.\cosh(u).\cos(v) \qquad (1.17)$$

Ahora trabajemos con la expresión 1.12 :

$$1 + \epsilon^2\{\cosh[N.\mathrm{arcosh}(\Omega/\Omega p)]\}^2 = 0$$
$$\epsilon^2\{\cosh[N.\mathrm{arcosh}(\Omega/\Omega p)]\}^2 = -1$$
$$\cosh[N.\mathrm{arcosh}(\Omega/\Omega p)] = \pm j/\epsilon$$

como: $\mathrm{arcosh}(s/j\Omega p) = u + jv$, entonces:

$$\cosh[N.(u+jv)] = \pm j/\epsilon \tag{1.18}$$

desarrollando este coseno tenemos:

$$\cosh(Nu).\cos(Nv) + j\,\mathrm{senh}(Nu)\mathrm{sen}(Nv) = \pm j/\epsilon$$

entonces:

$$\cosh(Nu).\cos(Nv) = 0 \tag{1.19}$$
$$\mathrm{senh}(Nu).\mathrm{sen}(Nv) = \pm 1/\epsilon \tag{1.20}$$

De 1.19:

$\cosh(Nu) \geq 1$; (conocido como la catenaria)

$\cos(Nv) = 0$

$$Nv = (2k-1)\pi/2$$
$$v = (2k-1)\pi/2N$$

De 1.20:

$\mathrm{sen}(Nv) = \pm 1$; (para los valores permitidos de v)

$\mathrm{senh}(Nu) = 1/\epsilon$

$$u = \frac{1}{N}\,\mathrm{arcsen}(\frac{1}{\epsilon}) \tag{1.21}$$

Reemplazando en 1.16 y 1.17 los valores determinados para u y v, tenemos:

$$\sigma_k = -\Omega p.\,\mathrm{senh}[\frac{1}{N}\mathrm{arcsenh}(\frac{1}{\epsilon})]\,.\,\mathrm{sen}(\frac{(2k-1)\pi}{2N})$$

$$\Omega_k = \Omega p.\cosh[\frac{1}{N}\mathrm{arcsenh}(\frac{1}{\epsilon})]\,.\cos(\frac{(2k-1)\pi}{2N})$$

A continuación reemplazaremos en estas expresiones, las siguientes igualdades:

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

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

$$\alpha = \frac{1}{N}.arcsenh(\frac{1}{\varepsilon})$$

$$\sigma_k = \Omega p.senh(\alpha).\cos(\frac{(2k+N-1)\pi}{2N}) \quad (1.22)$$

$$\Omega_k = \Omega p.\cosh(\alpha).sen(\frac{(2k+N-1)\pi}{2N}) \quad (1.23)$$

Estas son las expresiones con las cuales se determinarán los polos. Estos polos están sobre una elipse, porque:

Sea:

$\beta_k = (2k+N-1)\pi/2N$

$\sigma_k = X$

$\Omega_k = Y$

por tanto:

$$\frac{X^2}{\Omega p^2.senh^2(\alpha)} = \cos^2(\beta_k)$$

$$\frac{Y^2}{\Omega p^2.\cosh^2(\alpha)} = sen^2(\beta_k)$$

sumando estas expresiones tenemos:

$$\frac{X^2}{a^2} + \frac{Y^2}{b^2} = \cos^2(\beta_k) + sen^2(\beta_k) = 1$$

que efectivamente es la ecuación de una elipse.

Para localizar los polos en el plano S, se debe:

1) Trazar dos círculos centrados en el origen, de radios Ωp.cosh(α) y Ωp.senh(α), respectivamente.
2) Se ubican sobre estos dos círculos puntos equiespaciados con un ángulo de π/N de tal manera que los puntos estén ubicados simétricamente con respecto al eje imaginario, y además un punto caiga sobre el eje real para N impar pero no para N par. (La división de los círculos se la hace igual a la de Butterworth).
3) Los polos del filtro de Chevishev caen sobre la elipse con la ordenada especificada por los puntos que caen sobre el círculo mayor y la abscisa por los puntos sobre el círculo menor.

En la figura 1.9 se muestran los polos para N=3, en el plano S

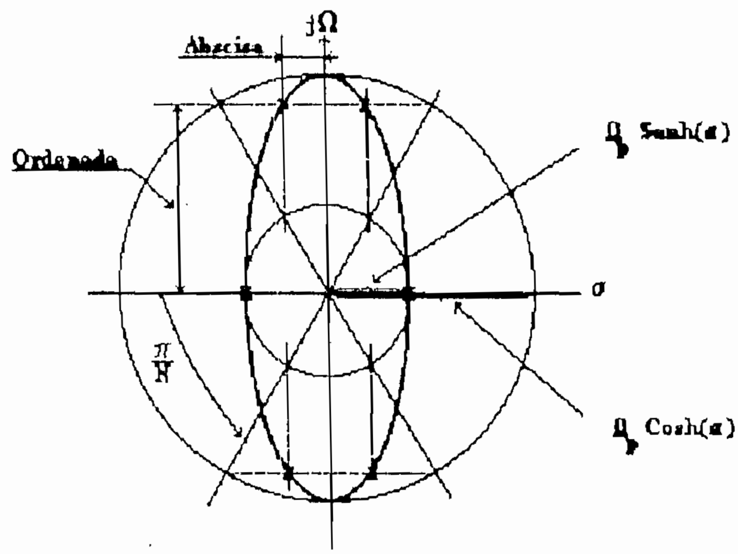


Figura 1.9: Ubicación de los Polos en el Plano "S" para un Filtro de Chevishev tipo I

1.2.4 DISEÑO DEL FILTRO DE CHEVISHEV TIPO I

El diseño de estos filtros consiste en determinar la función de transferencia a partir de las especificaciones dadas para la respuesta de frecuencia. Al igual que los filtros de Butterworth las especificaciones vienen dadas en decibelios, a las que hay que transformarlas a valores de ϵ y λ.

Primero, determinaremos el orden de un filtro de Chevishev:

Si $\Omega = \Omega p \rightarrow |Ha(\Omega p)|^2 = 1/(1+\epsilon^2)$

Si $\Omega = \Omega s \rightarrow |Ha(\Omega s)|^2 = 1/(1+\lambda^2)$

además:

$$|Ha(\Omega s)|^2 = \frac{1}{1+\epsilon^2 V_N^2(\frac{\Omega s}{\Omega p})}$$

y esto a su vez es igual a $1/(1+\lambda^2)$, por lo tanto:

$$\frac{1}{1+\epsilon^2 V_N^2(\frac{\Omega s}{\Omega p})} = \frac{1}{1+\lambda^2}$$

de aquí, despejaremos N:

$$\lambda^2 = \epsilon^2 V_N^2(\Omega s/\Omega p)$$

$$\lambda/\epsilon = V_N(\Omega s/\Omega p) = \cosh[N.\mathrm{arcosh}(\Omega s/\Omega p)]$$

$$N \geq \frac{\mathrm{arcosh}(\frac{\lambda}{\epsilon})}{\mathrm{arcosh}(\frac{\Omega s}{\Omega p})}$$

reemplazando arcosh(x) por su expresión logarítmica, se tiene:

$$N \geq \frac{\ln(\frac{\lambda}{\epsilon}+\sqrt{(\frac{\lambda}{\epsilon})^2-1})}{\ln(\frac{\Omega s}{\Omega p}+\sqrt{(\frac{\Omega s}{\Omega p})^2-1})} \qquad (1.24)$$

El valor de N debe ser entero, por lo tanto si es fraccionario se lo debe aproximar al entero superior mas cercano.

Luego el diseño continúa con la determinación de los polos usando las fórmulas encontradas para ello, donde K = 1...N, para tener una respuesta estable. Con estos polos se obtienen los factores del polinomio denominador de la función de transferencia, dejando en el numerador una constante a determinar con la evaluación de la función en $\Omega=0$. (Se tomará muy en cuenta la paridad de N).

Este procedimiento es igual al realizado en Butterworth.

$$Ha(s) = \frac{K}{(s-s_1)(s-s_2)\ldots(s-s_N)}$$

1.3 FILTROS DE CHEVISHEV TIPO II

Este tipo de filtros al contrario de los filtros de Chevishev tipo I, decrecen monótonamente en la banda de paso y tiene el equirrizado en la banda de supresión. Por esta característica al igual que los del tipo I, el filtro puede considerarse medianamente selectivo. Vea la figura 1.10

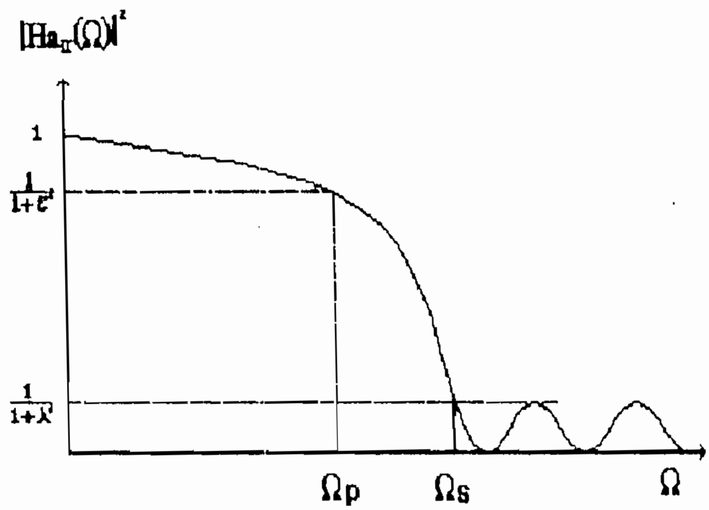


Figura 1.10: Respuesta de Frecuencia de un Filtro de Chevishev tipo II

1.3.1 FUNCION DE TRANSFERENCIA

Para encontrar una función que cumpla con la forma de respuesta de frecuencia, partiremos de la conocida función de transferencia del filtro tipo I. El primer paso a seguir será restar de la unidad el módulo al cuadrado de la función de transferencia del filtro de Chevishev tipo I ($|Ha_I(\Omega)|^2$).

$$1-|Ha_I(\Omega)|^2 = 1-\frac{1}{1+\epsilon^2 V_N^2(\frac{\Omega}{\Omega_P})}$$

$$= \frac{\epsilon^2 V_N^2(\frac{\Omega}{\Omega_P})}{1+\epsilon^2 V_N^2(\frac{\Omega}{\Omega_P})}$$

Este cambio ha hecho que la función tenga la forma de un filtro pasa-altos. (Figura 1.11)

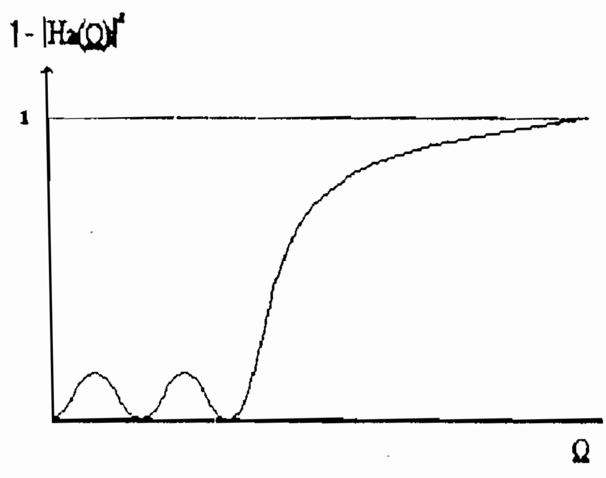


Figura 1.11

Ahora deseamos que el punto en $\Omega=0$ sea el ∞, y el $\Omega\to\infty$ sea el cero. Para este efecto hagamos un cambio de variable de Ω por $1/\Omega$, entonces:

$$|Ha_{II}(\Omega)|^2 = \frac{\epsilon^2 V_N^2(\frac{1}{\Omega.\Omega p})}{1+\epsilon^2 V_N^2(\frac{1}{\Omega.\Omega p})}$$

$$= \frac{1}{1+\frac{1}{\epsilon^2 V_N^2(\frac{1}{\Omega\Omega p})}}$$

En esta expresión se encuentran las constantes que especifican al filtro tipo I, para determinar las constantes que especifiquen al filtro tipo II las colocaremos como incógnitas, así:

$$|Ha_{II}(\Omega)|^2 = \frac{1}{1+\frac{K_1}{V_N^2(\frac{K_2}{\Omega})}}$$

El punto donde K_2/Ω empieza a ser menor que 1, y por tanto los polinomios son funciones trigonométricas es en $\Omega=\Omega s$, por lo tanto:

$$K_2/\Omega s = 1 \rightarrow K_2 = \Omega s$$

Con este resultado, evaluemos $|Ha_{II}(\Omega)|^2$ en $\Omega=\Omega p$ e igualemos a la especificación dada en la figura 1.10

$$|Ha_{II}(\Omega p)|^2 = \frac{1}{1+\frac{K_1}{V_N^2(\frac{\Omega s}{\Omega p})}} = \frac{1}{1+\epsilon^2}$$

entonces $K_1 = \epsilon^2 V_N^2(\Omega s/\Omega p)$, y por tanto el $|Ha_{II}(\Omega)|^2$ definitivo es:

$$|Ha_{II}(\Omega)|^2 = \frac{1}{1+\frac{\epsilon^2 V_N^2(\frac{\Omega s}{\Omega p})}{V_N^2(\frac{\Omega s}{\Omega})}} \tag{1.25}$$

Esta función de transferencia a diferencia de las funciones de transferencia de los filtros anteriores posee polos y ceros. Es decir que si factoramos, tendremos binomios en el numerador y en el denominador.

1.3.2 OBTENCION DE POLOS Y CEROS DEL FILTRO DE CHEVISHEV TIPO II

Trabajando un poco en la expresión 1.25, tenemos:

$$|Ha_{II}(\Omega)|^2 = \frac{V_N^2\left(\frac{\Omega s}{\Omega}\right)}{V_N^2\left(\frac{\Omega s}{\Omega}\right) + \epsilon^2 V_N^2\left(\frac{\Omega s}{\Omega p}\right)}$$

De esta expresión para determinar los ceros igualaremos a cero el numerador y para determinar los polos igualaremos a cero el denominador.

a) <u>OBTENCION DE LOS CEROS</u>

Entonces:

$$V_N^2(\Omega s/\Omega) = 0$$

$$\cos[N \arccos(\Omega s/\Omega)] = 0$$

Los ceros de $|Ha_{II}(\Omega)|^2$ "C_k" coinciden con los ceros de $V_N(x)$, y se ubican de la siguiente manera:

$$N.\arccos(\Omega s/\Omega) = \pi/2 + K\pi$$

$$\arccos(\Omega s/\Omega) = \pi/2N + 2K\pi/2N$$

$$\Omega s/\Omega = \cos[(2K+1)\pi/2N]$$

colocando el coseno en función del seno, tenemos:

$$\Omega s/\Omega = \mathrm{sen}[(2K+N-1)\pi/2N]$$

además como $C_k = j\Omega$, entonces:

$$C_k = \frac{j\Omega s}{sen\left(\frac{2K+N-1}{2N}\pi\right)} \tag{1.26}$$

donde K = 1,2,...2N

para $|Ha_{II}(\Omega)|$ se toman los primeros N ceros.

Esta fórmula obtenida nos hace ver que los ceros se ubican en el eje imaginario.

Un caso que puede suscitarse es el que si N es impar, el cero para k=(N+1)/2, se ubica en el infinito. Esto sucede porque el valor del seno que se encuentra en el denominador se hace cero.

b) OBTENCION DE LOS POLOS

Para la obtención de los polos, utilizaremos el valor de los polos encontrados para el filtro de Chevishev tipo I. La equivalencia que los relaciona es:

$$\frac{Sk_I}{\Omega p} = \frac{\Omega s}{Sk_{II}}$$

donde Sk_I es el k-ésimo polo del filtro tipo I y Sk_{II} es el k-ésimo polo del filtro tipo II. Esta equivalencia se la ha encontrado igualando las variables respectivas de los polinomios de Chevishev utilizadas en cada una de las funciones de transferencia.

Por lo tanto:

$$Sk_{II} = \Omega s \Omega p / Sk_I$$

donde: $Sk_I = \Omega p(\sigma k_I + j\Omega k_I)$

y:

$$\sigma k_I = \operatorname{senh}(\alpha).\cos(\beta k)$$
$$\Omega k_I = \cosh(\alpha).\operatorname{sen}(\beta k)$$

entonces:

$$Sk_{II} = \Omega s\Omega p/\Omega p(\ \sigma k_I + j\Omega k_I\ )$$

simplificando el valor de Ωp y racionalizando la expresión tenemos:

$$Sk_{II} = \frac{\Omega s}{\sigma k_I^2 + \Omega k_I^2}(\sigma k_I \pm j\Omega k_I)$$

Con esta expresión ya podemos determinar los polos del filtro de Chevishev tipo II.

Cabe indicar que en este caso:

$$\alpha = \frac{1}{N} arcsenh(\lambda) \tag{1.27}$$

$$\beta k = \frac{2k+N-1}{2N}\pi$$

donde λ se obtiene a partir del siguiente desarrollo:

$$|Ha_{II}(\Omega s)|^2 = \frac{1}{1+\frac{\epsilon^2 V_N^2(\frac{\Omega s}{\Omega p})}{V_N^2(\frac{\Omega s}{\Omega s})}} = \frac{1}{1+\lambda^2}$$

como $V_N^2(1)=1$ entonces:

$$\lambda = \epsilon.V_N(\Omega s/\Omega p)$$

Ahora demostraremos que efectivamente el valor de α es el correcto, con lo cual queda claramente establecido que para los filtros del tipo I α está en función de $1/\epsilon$, mientras que para los filtros del tipo II α está en función de λ.

Reemplacemos el valor de λ en 1.25, entonces:

$$|Ha_{II}(\Omega)|^2 = \frac{V_N^2(\frac{\Omega s}{\Omega})}{V_N^2(\frac{\Omega s}{\Omega})+\lambda^2}$$

igualemos el denominador a cero:

$$V_N^2(\frac{\Omega s}{\Omega})+\lambda^2 = 0$$

$$\cosh[N.(u+jv)] = \pm j\lambda$$

Este último resultado comparémosle con la expresión 1.18 y veamos claramente que se verifica la correspondencia.

1.3.3 DISEÑO DEL FILTRO DE CHEVISHEV TIPO II

Para determinar la función de transferencia a partir de las especificaciones dadas para la respuesta de frecuencia, determinemos la fórmula para el cálculo del orden N del filtro, y verifiquemos además que es idéntica a la determinada para los filtros tipo I de Chevishev.

Partamos de la expresión calculada para λ:

$$\lambda = \varepsilon.V_N(\Omega s/\Omega p)$$

entonces:

$$\lambda/\varepsilon = V_N(\Omega s/\Omega p) = \cosh[N.\mathrm{arcosh}(\Omega s/\Omega p)]$$

depejando N:

$$N \geq \frac{\mathrm{arcosh}(\frac{\lambda}{\varepsilon})}{\mathrm{arcosh}(\frac{\Omega s}{\Omega p})}$$

ahora reemplazando la equivalencia del coseno hiperbólico:

$$N \geq \frac{\ln\left(\frac{\lambda}{\epsilon}+\sqrt{\left(\frac{\lambda}{\epsilon}\right)^2-1}\right)}{\ln\left(\frac{\Omega_s}{\Omega_p}+\sqrt{\left(\frac{\Omega_s}{\Omega_p}\right)^2-1}\right)}$$

Esta respuesta es idéntica a 1.24

El diseño consiste en que determinados los polos y los ceros usando las fórmulas respectivas, y para K = 1...N; formemos los binomios en el numerador con los ceros, y en el denominador con los polos, así:

$$Ha(s)=\frac{K.(s-c_1)(s-c_2)\ldots(s-c_N)}{(s-s_1)(s-s_2)\ldots(s-s_N)}$$

La constante será determinada evaluando la función en s=0. Y como $|Ha(0)|^2=1$, entonces:

$$K=\frac{(-s_1)(-s_2)\ldots(-s_N)}{(-c_1)(-c_2)\ldots(-c_N)}$$

1.4 FILTROS ELIPTICOS [1.3]

En el estudio de los filtros analógicos, los filtros elípticos son considerados como los más complejos. El diseño de estos filtros demanda una solución numérica bastante extensa, además la mayoría de textos no tratan de este tópico.

La complejidad que conlleva el diseño de este filtro es compensada con las ventajas que de él se obtiene, ya que resulta ser un filtro altamente selectivo.

Esta propiedad de máxima selectividad se logra, al distribuir el error en la banda de paso y en la banda de supresión, utilizando rizado uniforme en ambas bandas. La forma que tiene la respuesta de frecuencia se muestra en la figura 1.12.

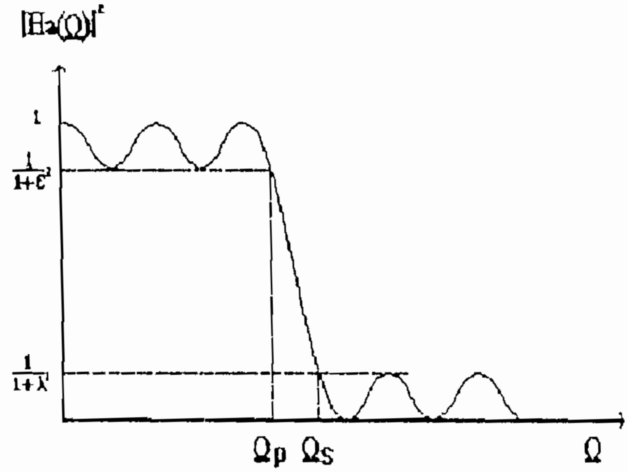


Figura 1.12: Respuesta de Frecuencia de un Filtro Elíptico

El análisis de este tipo de filtros fue realizado por Cauer, utilizando las funciones elípticas de Jacobi.

1.4.1 FUNCIONES ELIPTICAS

Las funciones elípticas de Jacobi son derivadas a partir de la integral elíptica de primera clase de Legendre. A continuación proporcionaremos un breve tratamiento de esta teoría, pero que será adecuada para nuestros propósitos.

1.4.1.1 Integral Elíptica de primera clase.

La integral elíptica de primera clase puede ser expresada como:

$$\mu(\phi, k) = \int_0^{\phi} \frac{d\theta}{\sqrt{1-k^2 sen^2(\theta)}} \qquad (1.28)$$

donde $0 \leq k < 1$. Para un valor real de ϕ, esta integral representa el área bajo la curva de la función:

$$I = \frac{1}{\sqrt{1-k^2 sen^2(\theta)}}$$

y las líneas verticales $\theta = 0$ y $\theta = \phi$

La función I es una función periódica con período π y tiene la forma: (ver la figura 1.13).

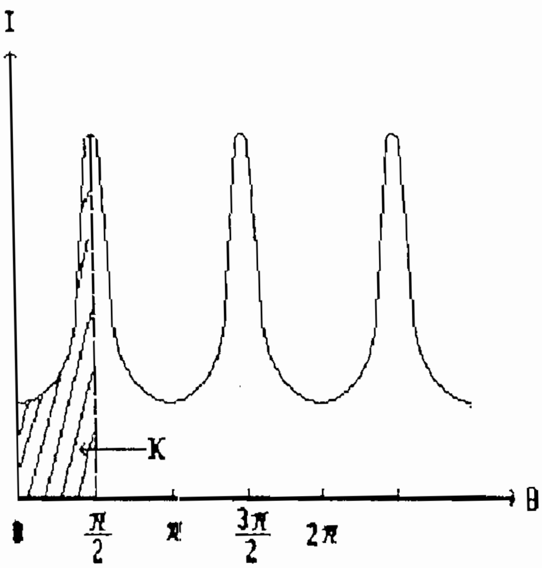


Figura 1.13: Función I

Por consecuencia de esta periodicidad el área limitada por las líneas $\theta = n.\pi/2$ y $\theta = (n+1).\pi/2$, es constante para cualquier número entero n. Cuando n=0 el área referida se conoce como la integral elíptica completa de primera clase, y está dada por:

$$\mu\left(\frac{\pi}{2}, k\right) = K = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1-k^2 sen^2(\theta)}}$$

De esta manera para un valor dado de k y ϕ, $\mu(\phi, k)$ puede ser cuantificado en unidades de K. Por ejemplo:

$$\mu(n\pi+\phi_1,k) = 2nk_1 + \mu(\phi_1,k)$$
$$\mu(\pi/2+\phi_1,k) = 2k_1 - \mu(\pi/2-\phi_1,k)$$

donde $0 \leq \phi_1 < \pi/2$. (Tomar en cuenta que $k \neq K$)

Evaluando 1.28 tenemos:

$$k=0 \quad \rightarrow \quad \mu(\phi,0) = \int_0^{\phi} d\theta = \phi$$

$$k=1 \quad \rightarrow \quad \mu(\phi,1) = \int_0^{\phi} \frac{d\theta}{\cos(\theta)} = \ln[\tan(\tfrac{\pi}{4}+\tfrac{\phi}{2})]$$

De estos resultados cuando $k=0$, $\mu(\phi,k)$ crece linealmente con ϕ, y cuando $k=1$, $\mu(\phi,k)$ tiene una discontinuidad en $\phi=\pi/2$. Por lo tanto para: $0 \leq \phi < \pi/2 \quad \rightarrow \quad \mu(\phi,0) \leq \mu(\phi,k) \leq \mu(\phi,1)$.

En la figura 1.14 se muestra la forma de esta integral μ, para algunos valores de k.

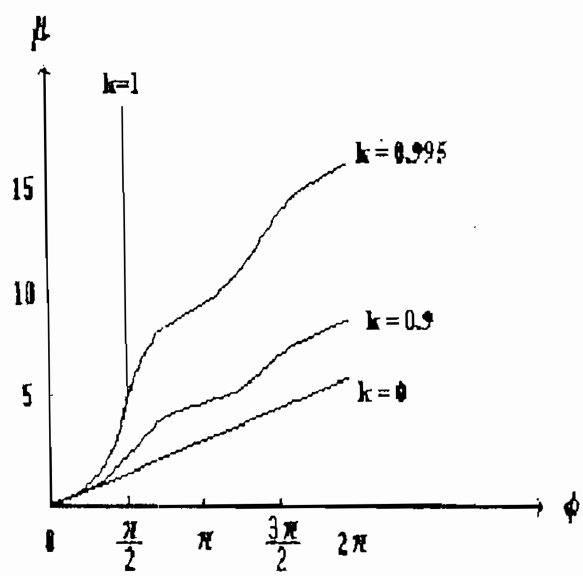


Figura 1.14: Formas de la Integral μ

1.4.1.2 Funciones Elípticas.

Hasta aquí hemos visto que existe una correspondencia entre μ y ϕ, donde a cada par de valores de (ϕ,k) le corresponde una amplitud única μ. Ahora dado un par de valores (μ,k), le corresponde un único valor de ϕ, tal que:

$$\phi = f(\mu,k)$$

Las funciones elípticas se definen utilizando el valor de ϕ, encontrado a partir de los valor de (μ,k), así:

$$\begin{aligned} sn(\mu,k) &= sen(\phi) \\ cn(\mu,k) &= cos(\phi) \\ dn(\mu,k) &= \sqrt{(1-k^2 sen^2 \phi)} \end{aligned} \tag{1.29}$$

estas funciones cumplen con algunas propiedades, derivadas de las propiedades trigonométricas:

$$sn^2(\mu,k) + cn^2(\mu,k) = 1$$
$$k^2 sn^2(\mu,k) + dn^2(\mu,k) = 1$$

La forma que tienen estas funciones se la muestra en la figura 1.15, donde son dibujadas en función de μ en unidades de K.

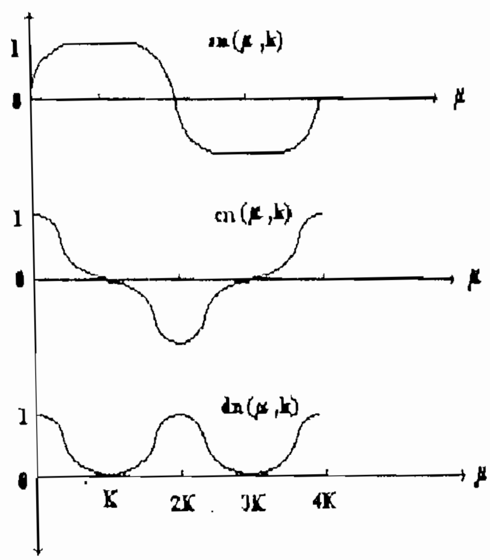


Figura 1.15:Funciones Elípticas

Como puede verse sn, cn y dn son funciones periódicas de μ con período 4K, 4K y 2K respectivamente:

$$sn(\mu + 4mK\ ,\ k) = sn(\mu\ ,\ k)$$
$$cn(\mu + 4mK\ ,\ k) = cn(\mu\ ,\ k)$$
$$dn(\mu + 2mK\ ,\ k) = dn(\mu\ ,\ k)$$

El efecto que produce el coeficiente k en las funciones elípticas, es el de alargamiento horizontal de la forma de las funciones seno y coseno trigonométricas.(Vea la figura 1.15)

Cuando k=0, la función sn es idéntica al seno y la función cn es idéntica al coseno. Cuando k=1 las funciones elípticas están en su máximo alargamiento.

1.4.1.3 Argumento Imaginario.

Hasta aquí el argumento de la función elíptica, llamado μ, ha sido asumido como una cantidad real. Realizando la integral 1.28 sobre una apropiada vía sobre el plano complejo, la integral elíptica puede asumir valores complejos. Vamos a considerar el caso de valor imaginario, donde:

$$jv = \int_0^{\Phi} \frac{d\theta}{\sqrt{1-k^2 sen^2(\theta)}} \tag{1.30}$$

Como en 1.29 podemos definir:

$$sn(jv,k) = sen(\Phi) \tag{1.31}$$
$$cn(jv,k) = cos(\Phi)$$
$$dn(jv,k) = \sqrt{(1-k^2 sen^2 \Phi)}$$

Estas funciones pueden ser expresadas en función de las funciones elípticas de argumento real, aplicando las siguientes transformaciones:

$$\text{sen}(\theta) = j\ \tan(\theta')$$
$$\text{sen}(\Phi) = j\ \tan(\Phi')$$

entonces:

$$\theta = \arcsin[j\ \tan(\theta')]$$

derivando respecto de θ':

$$\frac{d\theta}{d\theta'} = \frac{j}{\cos^2(\theta')\sqrt{1+\tan^2(\theta')}}$$

ahora dividiremos el término $\sqrt{[1-k^2\,\text{sen}^2(\theta)]}$, en ambos miembros:

$$\frac{d\theta}{\sqrt{1-k^2 sen^2\theta}} = \frac{j\,d\theta'}{\cos^2(\theta')\sqrt{1+\tan^2(\theta')}\sqrt{1-k^2 sen^2\theta}}$$

$$= \frac{j\,d\theta'}{\cos^2(\theta')\sqrt{1+\tan^2(\theta')}\sqrt{1-k^2(j\tan\theta')^2}}$$

aplicando identidades trigonométricas y simplicando, se tiene:

$$\frac{d\theta}{\sqrt{1-k^2 sen^2\theta}} = \frac{j\,d\theta'}{\sqrt{1-sen^2\theta'+k^2 sen^2\theta'}}$$

este resultado lo reemplazamos en 1.30, entonces tenemos:

$$jv = \int_0^{\psi'} \frac{j\,d\theta'}{\sqrt{1-sen^2(\theta')+k^2 sen^2(\theta')}}$$

la integral se evalúa en $\theta=\Phi$, como la variable ya no es θ, sino θ', tenemos:

$$\theta' = \arctan[\text{sen}(\theta)/j]$$
$$= \arctan[\text{sen}(\Phi)/j]$$
$$= \arctan[j\ \tan(\Phi')/j]$$
$$\theta' = \Phi'$$

reemplazando la constante $k' = \sqrt{(1-k^2)}$ llamada "módulo complementario", obtenemos:

$$v = \int_0^{\phi'} \frac{d\theta'}{\sqrt{1-(k')^2 sen^2(\theta')}}$$

Ahora de 1.29 tenemos:

$$\begin{aligned} sn(v,k') &= sen(\Phi') && (1.32)\\ cn(v,k') &= cos(\Phi') \\ dn(v,k') &= \sqrt{(1-(k')^2 sen^2 \Phi')} \end{aligned}$$

reemplazando 1.32 en 1.31, y haciendo algunas operaciones:

$$\begin{aligned} sn(jv,k) &= j\ tan(\Phi') \\ &= j\ sen(\Phi')/cos(\Phi') \\ &= j\ sn(v,k')/cn(v,k') \end{aligned}$$

$$cn(jv,k) = 1 / cos(v,k')$$

$$dn(jv,k) = dn(v,k')/cn(v,k') \qquad (1.33)$$

Por analogía la integral elíptica complementaria de primera clase, está dada por:

$$K' = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1-(k')^2 sen^2(\theta)}}$$

Y tiene la misma interpretación que K, esto es, que la cuarta parte del período de sn(v,k´) y cn(v,k´) o el medio período de dn(v,k´). Las funciones sn(jv,k), cn(jv,k) y dn(jv,k) son funciones periódicas de jv, como puede verse en la figura 1.16, con períodos de $j2k_1'$, $j4k_1'$ y $j4k_1'$, respectivamente.

$$sn(jv + 2nK', k) = sn(jv, k)$$
$$cn(jv + 4nK', k) = cn(jv, k)$$

$$dn(jv + 4nK', k) = dn(jv, k)$$

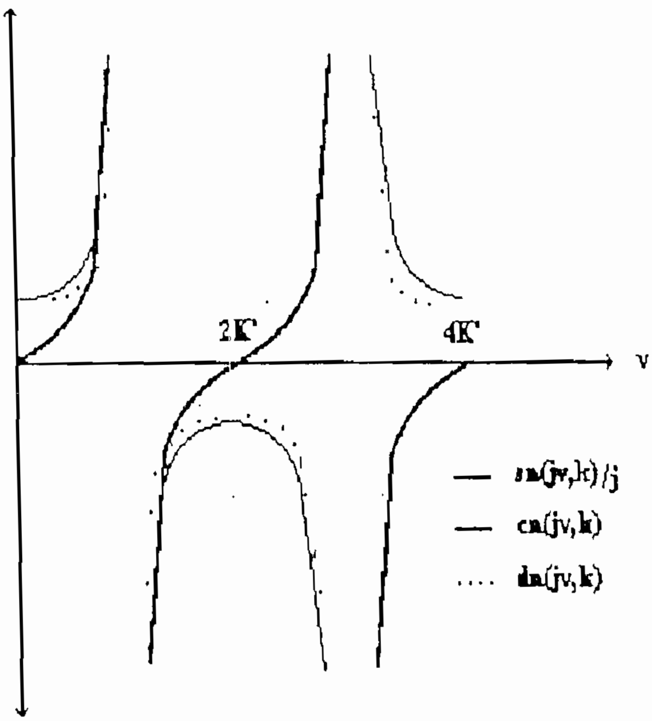


Figura 1.16: Funciones Elípticas con Argumento Imaginario

1.4.1.4 Algunas Fórmulas.

Las funciones elípticas al igual que las funciones trigonométricas son interrelacionadas con fórmulas. Una de las fórmulas más básicas es la "fórmula de adición", que tiene la siguiente forma:

$$sn(z1+z2,k) = \frac{sn(z_1,k)\,cn(z_2,k)\,dn(z_2,k) + cn(z_1,k)\,sn(z_2,k)\,dn(z_1,k)}{D}$$

$$cn(z1+z2,k) = \frac{cn(z_1,k)\,cn(z_2,k) - sn(z_1,k)\,sn(z_2,k)\,dn(z_1,k)\,dn(z_2,k)}{D}$$

$$dn(z1+z2,k) = \frac{dn(z_1,k)\,dn(z_2,k) - k^2 sn(z_1,k)\,sn(z_2,k)\,cn(z_1,k)\,cn(z_2,k)}{D}$$

donde $D = 1- k^2 sn^2(z_1,k)sn^2(z_2,k)$, y z_1, z_2 son variables reales o complejas.

Otra fórmula de interés es:

$$dn^2(\tfrac{z}{2},k) = \frac{dn(z,k)+cn(z,k)}{1+cn(z,k)} \tag{1.34}$$

1.4.1.5 <u>Variable de Transformación</u>.

Por otra parte la ecuación: $\Omega = \sqrt{k}\ sn(z,k)$, se conoce como una variable de transformación del plano z (complejo) en un plano Ω (complejo). Examinaremos a continuación las propiedades de esta transformación, que posteriormente necesitaremos en la derivación de la función de transferencia de los filtros elípticos.

Sea z_p un punto cualquiera del plano z, entonces:

$$z = z_p + 4mK + j2nK$$

Existen tres dominios de Ω que son de interés:

Dominio 1:	$z=\mu$	con	$0 \leq \mu \leq K$
Dominio 2:	$z=K+jv$	con	$0 \leq v \leq K'$
Dominio 3:	$z=\mu + jK'$	con	$0 \leq \mu \leq K$

En el DOMINIO 1:

$$\Omega = \sqrt{k}\ sn(\mu,k)$$

si $\mu=0 \quad \rightarrow \quad \Omega = \sqrt{k}\ sn(0,k) = 0$

si $\mu=K \quad \rightarrow \quad \Omega = \sqrt{k}\ sn(K,k) = \sqrt{k}$

esto quiere decir que el eje real del plano z entre 0 y K, se transforma en el eje real del plano Ω entre 0 y $\sqrt{k}$.

En el DOMINIO 2:

$$\Omega = \sqrt{k}\ sn(\ K+jv\ ,\ k)$$

de la fórmula de adición, tenemos:

$$\Omega = \frac{\sqrt{k}\, cn(jv,k)\, dn(jv,k)}{1-k^2 sn^2(jv,k)}$$

de las expresiones 1.33 y sabiendo que cn(K,k)=0, tenemos:

$$\Omega = \frac{\sqrt{k}\, dn(v,k')}{cn^2(v,k')+k^2 sn^2(v,k')}$$

reduciendo el denominador podemos llegar a:

$$\begin{aligned} cn^2(v,k')+k^2 sn^2(v,k') &= 1 - sn^2(v,k')+k^2 sn^2(v,k') \\ &= 1 - (k')^2 sn^2(v,k') \\ &= dn^2(v,k') \end{aligned}$$

por lo tanto:

$$\Omega = \frac{\sqrt{k}}{dn(v,k')}$$

en esta expresión:

si v=0 $\rightarrow$ $\Omega = \sqrt{k} / dn(0,k') = \sqrt{k}$

si v=K´ $\rightarrow$ $\Omega = \sqrt{k} / dn(K',k') = 1/\sqrt{k}$

si v=K´/2 $\rightarrow$ usando 1.34, tenemos:

$$\Omega = \frac{\sqrt{k}}{dn(K'/2,k')} = \sqrt{k}\sqrt{\frac{1+cn(K',k')}{dn(K',k')+cn(K',k')}} = 1$$

esto demuestra que la línea de puntos z=K+jv para v entre 0 y K´, se transforma en le eje real del plano Ω entre $\sqrt{k}$ y $1/\sqrt{k}$.

En el DOMINIO 3:

$$\Omega = \sqrt{k}\, sn(\mu+jK', k)$$

de la fórmula de adición, tenemos:

$$\Omega = \frac{1}{\sqrt{k}\ sn(\mu, k)}$$

si $\mu = 0$ →

$$\Omega = \frac{1}{\sqrt{k}\ sn(0, k)} = \infty$$

si $\mu = K$ →

$$\Omega = \frac{1}{\sqrt{k}\ sn(K, k)} = \frac{1}{\sqrt{k}}$$

esto nos dice que la línea de puntos $z=\mu+jK'$ con μ entre 0 y K se transforma en el eje real del plano Ω entre ∞ y $1/\sqrt{k}$.

En la figura 1.17 se muestra gráficamente la transformación de la trayectoria ABCD del plano z en la trayectoria A´B´C´D´ en el plano Ω.

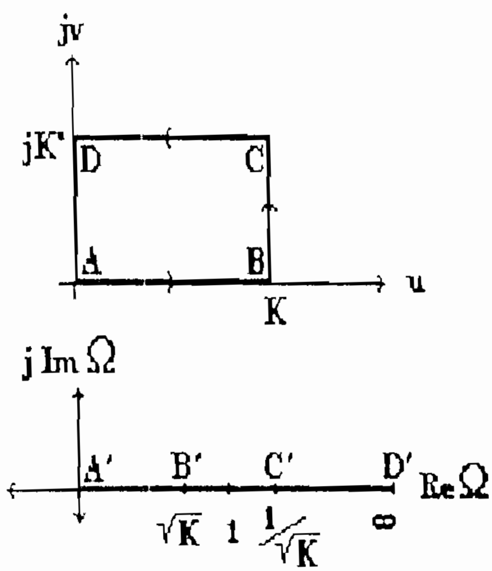


Figura 1.17: Transformación de la Trayectoria ABCD

1.4.2 FUNCION DE TRANSFERENCIA DEL FILTRO ELIPTICO

En base al gráfico de la respuesta de atenuación (Figura 1.18) , se determina la función de transferencia. Por facilidad de

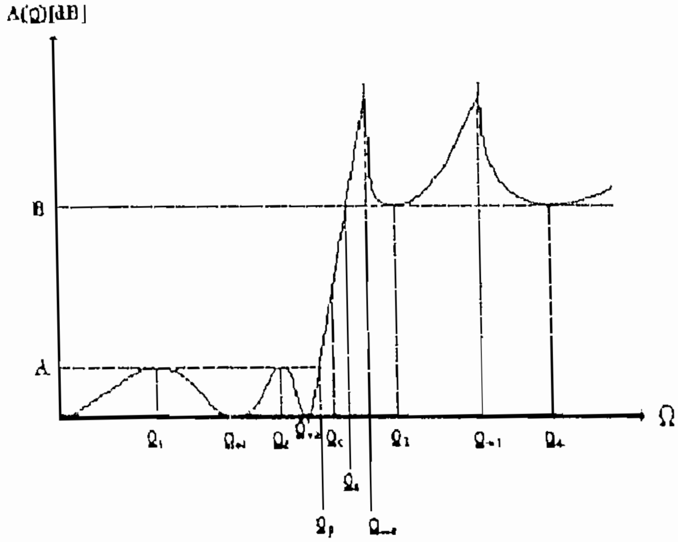


Figura 1.18: Respuesta de Atenuación de un Filtro Elíptico

la deducción se determina la función de transferencia de un filtro pasa-bajos NORMALIZADO, a partir del cual con las transformaciones adecuadas podemos obtener cualquier tipo de filtro y con las frecuencias de corte deseadas.

Denominamos filtro elíptico normalizado, aquel filtro cuya Ωc cumple con:

$$\Omega c = \sqrt{(\Omega p . \Omega s)} = 1$$

y por lo tanto:

$$\Omega p = \sqrt{k}$$
$$\Omega s = 1/\sqrt{k}$$
$$k = \Omega p/\Omega s$$

se define a la constante k_1 como:

$$k_1 = \varepsilon/\lambda$$

que en el desarrollo nos ayudará de mucho. Los valores de ε y

λ guardan la misma relación con los valores A y B de atenuación en decibelios.

1.4.2.1 <u>Aproximación para un filtro de orden quinto</u>.

Por ejemplo desarrollemos la aproximación para un filtro elíptico de orden N=5.

Sea función de la atenuación:

$$L(\Omega^2) = 1 + \varepsilon^2 F^2(\Omega)$$

$$A(\Omega) = 10 \log L(\Omega^2) \qquad \text{[En decibelios]}$$

Determinaremos algunas propiedades que la función F y L debe cumplir:

Propiedad 1:	$F(\Omega) = 0$	si: $\Omega = 0, \pm\Omega_{o1}, \pm\Omega_{o2}$
Propiedad 2:	$F(\Omega) = \infty$	si: $\Omega = \infty, \pm\Omega_{\infty 1}, \pm\Omega_{\infty 2}$
Propiedad 3:	$F^2(\Omega) = 1$	si: $\Omega = \pm\Omega_1, \pm\Omega_2, \pm\sqrt{k}$
Propiedad 4:	$F^2(\Omega) = 1/k_1^2$	si: $\Omega = \pm\Omega_4, \pm\Omega_3, \pm 1/\sqrt{k}$
Propiedad 5:	$dL^2(\Omega)/d\Omega = \infty$	si: $\Omega = \pm\Omega_1, \pm\Omega_2, \pm\Omega_3, \pm\Omega_4,$ $0, \pm\Omega_{o1}, \pm\Omega_{o2}$

De las propiedades 1 y 2 tenemos que:

$$F(\Omega) = \frac{M_1 \Omega (\Omega^2-\Omega_{01}^2)(\Omega^2-\Omega_{01}^2)}{(\Omega^2-\Omega_{\infty 1}^2)(\Omega^2-\Omega_{\infty 2}^2)} \qquad (1.35)$$

De las propiedades 2 y 3 tenemos que:

$$1-F^2(\Omega) = \frac{M_2 \cdot (\Omega^2-\Omega_1^2)(\Omega^2-\Omega_2^2)(\Omega^2-k)}{(\Omega^2-\Omega_{\infty 1}^2)^2 (\Omega^2-\Omega_{\infty 2}^2)^2}$$

De las propiedades 2, 4 y 5 tenemos que:

$$1-k_1^2F^2(\Omega) = \frac{M_3(\Omega^2-\Omega_3^2)(\Omega^2-\Omega_4^2)(\Omega^2-1/k)}{(\Omega^2-\Omega_{-1}^2)^2(\Omega^2-\Omega_{-2}^2)^2}$$

De la propiedad 5 tenemos:

$$\frac{dF(\Omega)}{d\Omega} = \frac{M_4(\Omega^2-\Omega_1^2)(\Omega^2-\Omega_2^2)(\Omega^2-\Omega_3^2)(\Omega^2-\Omega_4^2)}{(\Omega^2-\Omega_{-1}^2)^2(\Omega^2-\Omega_{-2}^2)^2}$$

Ahora combinando todas estas expresiones tenemos:

$$\left(\frac{dF(\Omega)}{d\Omega}\right)^2 = \frac{M_5[1-F^2(\Omega)][1-k_1^2F^2(\Omega)]}{(1-\Omega^2/k)(1-k\Omega^2)} \tag{1.36}$$

Alternativamente podemos escribir:

$$\int_0^F \frac{dx}{\sqrt{(1-x^2)(1-k_1^2x^2)}} = \sqrt{M_5}\int_0^\Omega \frac{dy}{\sqrt{(1-y^2/k)(1-ky^2)}} + M_7$$

y si y=√ky´, y´=y, entonces:

$$\int_0^F \frac{dx}{\sqrt{(1-x^2)(1-k_1^2x^2)}} = M_6\int_0^{\frac{\Omega}{\sqrt{k}}} \frac{dy}{\sqrt{(1-y^2)(1-k^2y^2)}} + M_7$$

usando las siguientes transformaciones $x=\mathrm{sen}(\theta_1)$, $F=\mathrm{sen}(\phi_1)$, $y=\mathrm{sen}(\theta)$, $y=\mathrm{sen}(\phi)$; tenemos:

$$\int_0^{\phi_1} \frac{\cos\theta_1 d\theta_1}{\sqrt{(1-sen^2\theta_1)(1-k_1^2 sen^2\theta_1)}} = M_6\int_0^{\phi} \frac{\cos\theta d\theta}{\sqrt{(1-sen^2\theta)(1-k^2 sen^2\theta)}} + M_7$$

Considerando que $dx=\cos(\theta)d\theta$, $\cos(\theta)=\sqrt{(1-\mathrm{sen}^2\theta)}$:

$$\int_0^{\phi_1} \frac{d\theta_1}{\sqrt{(1-k_1^2 sen^2\theta_1)}} = M_6 \int_0^{\phi} \frac{d\theta}{\sqrt{(1-k^2 sen^2\theta)}} + M_7$$

Estas integrales pueden asumir valores complejos, si se establece que ϕ y ϕ_1 son a su vez complejos. De esta manera, sea:

$$\int_0^{\phi} \frac{d\theta}{\sqrt{(1-k^2 sen^2\theta)}} = z$$

donde $z = \mu + jv$

La solución de 1.36 puede ser expresada en función de un par de ecuaciones simultáneas, como:

$$\Omega/\sqrt{k} = sen(\phi) = sn(z,k) \qquad (1.37)$$

$$F = sen(\phi_1) = sn(M_6\, z + M_7\ ,\ k_1) \qquad (1.38)$$

Deseamos determinar $F(\Omega)$ a lo largo de todo el espectro $0 \leq \Omega \to \infty$. En la sección 1.4.1.5 se desarrolla la transformación de Ω en z, por tanto tomando esta transformación y observando las propiedades que la función F debe cumplir, determinaremos la forma definitiva de F.

a) Sea del DOMINIO 1, $0 \leq \Omega \leq \sqrt{k}$:

entonces $z = \mu$; $0 \leq \mu \leq K$

por lo tanto F será:

$$F = sn\ (\ M_6\mu + M_7\ ,\ k_1\ )$$

La forma que tienen las funciones elípticas 1.37 y 1.38 en este rango se indican en la figura 1.19.

Para $\Omega=0$; de la propiedad 1 tenemos $F = 0$

y como $\mu=0$, entonces:

$$F = sn\ (M_7\ ,\ k_1\ ) = 0 \quad \to \quad M_7 = 0.$$

De las propiedades de las funciones elípticas, F tiene un período de $4K_1/M_6$ y Ω tiene un período de 4K. El dominio en el que se está trabajando resulta ser un cuarto del período de Ω. Si observamos la figura 1.19 la función F muestra los 5/4 de su período, en este dominio.

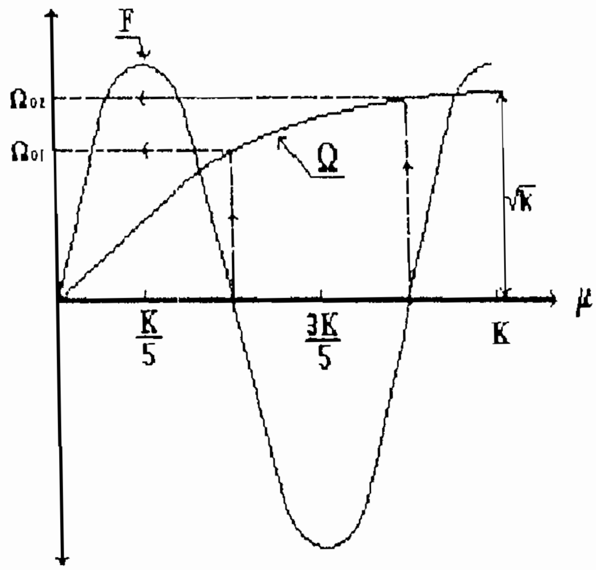


Figura 1.19: Funciones F y Ω

Por lo tanto:

$$K = \frac{5}{4}\frac{4K_1}{M_6}$$

$$M_6 = \frac{5K_1}{K}$$

De esta manera F será:

$$F = sn\left(\frac{5K_1}{K}\mu, k_1\right)$$

Además F=0 en los puntos de $u = 2K.i/5$ donde i=0,1,2 es decir para los valores de Ω igual a:

$$\Omega_{0i} = \sqrt{k}\; sn\left(\frac{2Ki}{5}, k\right) \qquad (1.39)$$

Hasta aquí para este dominio hemos encontrado que la forma de F es igual a la forma de la función elíptica de argumento real y con contante k. Además los valores de Ω para los cuales la función F es cero.

b) Sea del DOMINIO 3, $1/\sqrt{k} \leq \Omega \rightarrow \infty$:

entonces $z = \mu + jK'$; $0 \leq \mu \leq K$

por lo tanto F será:

$$F = sn\left(\frac{5K_1(\mu + jK')}{K}, k_1\right) \tag{1.40}$$

y Ω será:

$$\Omega = \frac{1}{\sqrt{k}\, sn(\mu, k)}$$

Para $\Omega \rightarrow \infty$, u=0 y $F \rightarrow \infty$, usando la propiedad 2.

$$F = sn\left(\frac{j\,5K_1\,K'}{K}, k_1\right) = \infty$$

De la ecuaciones 1.33, tenemos:

$$F = j\frac{sn\left(\frac{5K_1K'}{K}, k_1'\right)}{cn\left(\frac{5K_1K'}{K}, k_1'\right)} = \infty$$

donde $k_1' = \sqrt{[1-k_1^2]}$. Para que $F \rightarrow \infty$, entonces:

$$cn\left(\frac{5K_1K'}{K}, k_1'\right) = 0$$

Las cantidades K, K´ son funciones de k, y similarmente K_1, K_1' son funciones de k_1. Por eso para que cn sea cero, debe cumplirse que:

$$\frac{5K_1K'}{K} = K_1' \qquad (1.41)$$

Con este resultado, volviendo a 1.40 tenemos:

$$F = sn\left(\frac{5K_1}{K}\mu + jK_1', k_1\right)$$

Trabajando sobre esta expresión, obtenemos:

$$F = \frac{1}{k_1\, sn\left(\frac{5K_1}{K}\mu, k_1\right)}$$

evidentemente para que $F \to \infty$, μ será:

$$5K_1\mu/K = 2K_1 i$$

$$\to \quad \mu = 2Ki/5 \qquad \text{para} \quad i = 0,1,2$$

esto es que F tiene polos en:

$$z = 2Ki/5 + jK' \qquad \text{para } i = 0,1,2$$

los correspondientes valores de Ω para los cuales $F \to \infty$, son:

$$\Omega_{\infty i} = \frac{1}{\sqrt{k}\, sn\left(\frac{2Ki}{5}, k\right)}$$

Si nos fijamos en 1.39, este resultado es el inverso de esa expresión, por lo tanto:

$$\Omega_{\infty i} = \frac{1}{\Omega_{0i}}$$

entonces la expresión 1.35 se convierte en:

$$F(\Omega) = \frac{M_1'\Omega(\Omega^2-\Omega_{01}^2)(\Omega^2-\Omega_{02}^2)}{(1-\Omega^2\Omega_{01}^2)(1-\Omega^2\Omega_{02}^2)} \qquad (1.42)$$

Hasta aquí lo único que desconocemos es la constante M_1', para determinar su valor hacemos el análisis en el dominio 2.

c) Sea DOMINIO 2, $\sqrt{k} \leq \Omega \leq 1/\sqrt{k}$:

entonces $z = K + jv$; $0 \leq v \leq K'$

por lo tanto F será:

$$F = sn\left(\frac{5K1(K+jv)}{K}, k_1\right) \qquad (1.43)$$

Para este dominio Ω es:

$$\Omega = \frac{\sqrt{k}}{dn(v, k')}$$

Ahora evaluemos 1.42 en $\Omega=1$, entonces, $F(1)=M_1'$. También evaluemos 1.43 en $\Omega=1$, sabiendo que el correspondiente valor de μ es: $\mu = K'/2$.

$$F(1) = sn\left(5K_1+j\frac{5K'K_1}{2K}, k_1\right) = M_1'$$

Trabajando en esta expresión podemos llegar a:

$$M_1' = \frac{1}{dn\left(\frac{K_1'}{2}, k_1'\right)} = \frac{1}{\sqrt{k_1}}$$

Resumiendo todo este análisis nos llevó a determinar que $F(\Omega)$

para un filtro de quinto orden es:

$$F(\Omega) = \frac{\Omega(\Omega^2-\Omega_{01}^2)(\Omega^2-\Omega_{02}^2)}{\sqrt{K_1}(1-\Omega^2\Omega_{01}^2)(1-\Omega^2\Omega_{02}^2)}$$

de esta expresión se deprende que $F(\Omega)$ depende de k_1. Los valores Ω_{01} se determinan con una fórmula que depende de k. A su vez k y k_1 son funciones de las especificaciones A y B de la respuesta de frecuencia.

1.4.2.2 <u>Aproximación para un filtro de orden impar</u>.

Por analogía con el proceso anterior, de la ecuación 1.38 M_7 es cero y los N cuartos de período de F corresponde a un cuarto del período de Ω, esta condición determina el valor de la constante M_6.

$$M_6 = \frac{NK_1}{K}$$

reemplazando este valor en 1.38, tenemos:

$$F = sn(\ N\ K_1\ z/K\ ,\ k_1\ ) \tag{1.44}$$

similarmente con 1.41, podemos generalizar que:

$$NK'/K = K_1'/K_1$$

a partir de esta expresión por analogía con lo que se hizo para N=5 tenemos:

$$F(\Omega) = \frac{(-1)^r\Omega}{\sqrt{K_1}} \prod_{i=1}^{r} \frac{(\Omega^2-\Omega_i^2)}{(1-\Omega^2\Omega_i^2)}$$

donde $r = (N-1)/2$

y con este resultado los ceros y polos se los determina con la expresión:

$$\Omega_i = \sqrt{k}\, sn\,(\frac{2Ki}{N}, k)$$

para i = 1,2,...r

1.4.2.3 Aproximación para un filtro de orden par.

Es fácilmente extendible a partir de los resultados de la aproximación del filtro de orden impar. La función F es la misma que se obtuvo para orden impar, con la aclaración que para N par , r debe ser:

$$r = N/2$$

los ceros y polos son iguales a:

$$\Omega_i = \sqrt{k}\, sn\,(\frac{(2i-1)K}{N}, k)$$

para i = 1,2,...,r

1.4.2.4 Generalización.

Hasta aquí hemos determinado la función F(Ω), pero la función de transferencia no es ésta. Para determinar la función de transferencia a partir de F(Ω), debemos tener presente que entre F(Ω) y $|Ha(\Omega)|^2$ existe la relacion:

$$|Ha(\Omega)|^2 = \frac{1}{1+\alpha^2 F^2(\Omega)}$$

Un procedimiento a seguir para que el trabajo de diseñar un filtro elíptico sea formar "r" secciones de la forma:

$$\frac{s^2 + A_{0i}}{s^2 + B_{1i}s + B_{0i}} \qquad \text{con } i = 1\ ,\ 2\ ,\ldots,\ r$$

consiste en determinar los ceros y polos de la función $L(-s^2)$, que es el denominador de la función $|H(\Omega)|^2$, donde $s=\Omega/j$.

A continuación determinaremos los ceros y polos de L:

Sea

$$L = 1 + \varepsilon^2 F^2(\Omega)$$

reemplazamos 1.44, entonces:

$$L(z) = 1 + \varepsilon^2 \ sn^2(NK_1\ z/K, k_1)$$

$$L(z) = [1 + j\varepsilon\ sn(NK_1\ z/K, k_1)][1 - j\varepsilon\ sn(NK_1\ z/K, k_1)]$$

los ceros de L(z) se determinan resolviendo:

$$sn(\ NK_1\ z/K, k_1\ ) = j/\varepsilon$$

debido a que en la práctica $k_1 = \varepsilon/\lambda$ es muy pequeño, y es más cuando el filtro es más selectivo porque ε decrece y λ crece; entonces podemos asumir que $k_1 = 0$. Por esta aproximación el seno elíptico es igual al seno hiperbólico, entonces:

$$sen(\ NK_1\ z/K\ ) = j/\varepsilon$$

donde $K_1 = \pi/2$, de acuerdo con $\mu(\pi/2,0) = K$. Entonces tendremos:

$$-jN\pi z/2K = \text{arcsenh}(1/\varepsilon)$$

usando la identidad del arcsenh, reemplazando el valor de ε por 1.3, y sabiendo que $z_o = jv_o$, obtenemos que:

$$v_o = \frac{K}{N\pi}\ln\left(\frac{10^{0.05A}+1}{10^{0.05A}-1}\right)$$

Por la periodicidad de la función sn, tenemos que ceros de L se dan para:

$$z_i = z_o + 4Ki/N, \quad \text{donde } i=0,1,2....$$

El correspondiente valor en el plano "s" se encontrará con la fórmula de transformación; esto es:

$$s=\sigma_i + j\Omega_i' = j\sqrt{k}\ sn(\ jv_o + 4K_i/N,k\ ) \tag{1.45}$$

donde Ω_i' es distinto de Ω_i, ya que el Ω_i' es la parte imaginaria de los ceros de "L" y el Ω_i representa los ceros de la función "F". De 1.45, utilizando la fórmula de adición y a la periodicidad de las funciones elípticas, podemos llegar a:

$$\sigma_i + j\Omega_i' = \frac{(-1)^i\sigma_0 V_i \pm j\Omega_i W}{1+\sigma_0^2\Omega_i^2}$$

donde:

$$W = \sqrt{(1+k\sigma_0^2)\ (1+\sigma_0^2/k)}$$

$$V_i = \sqrt{(1-k\Omega_i^2)\ (1-\Omega_i^2/k)}$$

$$\Omega_i = \sqrt{k}\ sn(\frac{2Ki}{N},k)$$

si en 1.45 reemplazamos i=0, encontraremos σ_o:

$$\sigma_o = j\sqrt{k}\ sn(\ jv_o\ ,k\ )$$

Las funciones elípticas tienen una representación en forma de series, así:

$$\sigma_0 = \frac{-2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)}\, senh[(2m+1)\Lambda]}{1 + 2\sum_{m=1}^{\infty}(-1)^m q^{m^2}\cosh(2m\Lambda)}$$

donde $\Lambda = vo\pi/2K$

el parámetro "q" se llama "constante modular" y esta dado por:

$$q = e^{-\pi K'/K} \tag{1.46}$$

este valor se lo puede calcular evaluando K y K´. Existe otro método que se basa en:

Dado dn(0,k)=1, usando la representación en series tenemos:

$$\sqrt{k'} = \frac{1 - 2q + 2q^4 - 2q^9 + \ldots}{1 + 2q + 2q^4 + 2q^9 + \ldots}$$

haciendo una primera aproximación, despreciamos los términos que son de grado mayor a 1, y despejando q tenemos: (A esta aproximación le hemos llamado qo)

$$q_0 = \frac{1}{2}\left(\frac{1-\sqrt{k'}}{1+\sqrt{k'}}\right)$$

usando esta aproximación, y haciendo algunos artificios matemáticos, llegamos a la siguiente expresión para q:

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13}$$

Esta generalización es válida para N impar. En base a estas mismas expresiones se hace la aproximación para N par. Logramos esto haciendo que i = i - 1/2 para las expresiones que son funciones de i, para el resto de ecuaciones no cambiamos nada.

Cabe recalcar que todo el desarrollo se basa en asumir la siguiente relación:

$$N\frac{K}{K'} = \frac{K_1'}{K_1} \tag{1.47}$$

Haremos un análisis para relacionar k_1 en el diseño del filtro por este método. La constante k_1 es de mucha importancia porque lleva la información del grado de las atenuaciones en la banda de paso y en la banda de supresión.

De la definición de la función elíptica $sn(K_1,k_1)=1$, y expandiendo en series tenemos:

$$k_1 = 4\sqrt{q_1}\left(\frac{1 + q_1^2 + q_1^6 + \ldots}{1 + 2q_1 + 2q_1^4 + \ldots}\right)^2$$

donde:

$$q_1 = e^{-\pi K_1'/K_1}$$

Podemos asumir que:

$$k_1 \approx 4\sqrt{q_1} \qquad ó \qquad k_1^2 = 16q_1$$

entonces :

$$k_1^2 = 16e^{-\pi K_1'/K_1}$$

reemplazando 1.46 y 1.47 tenemos:

$$k_1^2 = 16q^N$$

de este resultado podemos calcular el orden del filtro que se diseñará. Despejemos N y definamos a $D = 1/k_1^2$

$$N \geq \frac{\log(16D)}{\log(1/q)}$$

de hecho:

$$D = \frac{10^{0.1B}-1}{10^{0.1A}-1}$$

Una vez conocida k, las cantidades k', q_0, q, σ_0, Ω_i, σ_i y Ω_i' pueden ser evaluadas. A partir de estos valores se puede formar la función de transferencia normalizada Ha(s).

La función de transferencia normalizada para un filtro pasa-bajos tiene la siguiente forma:

$$Ha(s) = \frac{Ho}{Do(s)} \prod_{i=1}^{r} \frac{s^2+A_{0i}}{s^2+B_{1i}s+B_{0i}}$$

donde:

r =	(N-1)/2	para N impar
	N/2	para N par

y:

Do(s) =	$s+\sigma_o$	para N impar
	1	para N par

los coeficientes y la constante Ho pueden ser calculados usando las siguientes fórmulas, en la secuencia que aparecen: (Por tener un formulario ordenado, algunas expresiones las repetiremos).

$$k' = \sqrt{1-k^2}$$

$$q_o = \frac{1}{2}\left(\frac{1-\sqrt{k'}}{1+\sqrt{k'}}\right)$$

$$q = q_o + 2q_o^5 + 15q_o^9 + 150q_o^{13}$$

$$D = \frac{10^{0.1B}-1}{10^{0.1A}-1}$$

$$N \geq \frac{\log(16D)}{\log(1/q)}$$

$$\Lambda = \frac{1}{2N}\ln\frac{10^{0.05A}+1}{10^{0.05A}-1}$$

$$\sigma_o = \frac{2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)}\, senh[(2m+1)\Lambda]}{1+2\sum_{m=1}^{\infty}(-1)^m q^{m^2}\cosh(2m\Lambda)}$$

$$W = \sqrt{(1+k\sigma_0^2)(1+\sigma_0^2/k)}$$

$$\Omega_i = \frac{2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)}\, sen\left[\frac{(2m+1)\pi\mu}{N}\right]}{1+2\sum_{m=1}^{\infty}(-1)^m q^{m^2}\cos\left(\frac{2m\pi\mu}{N}\right)}$$

este resultado viene de representar en series la función elíptica.

donde: $\mu = i$ para N impar

$\mu = i-1/2$ para N par.

para $i = 1,2,\ldots r$

$$V_i = \sqrt{(1-k\Omega_i^2)(1-\Omega_i^2/k)}$$

$$A_{0i} = \frac{1}{\Omega_i^2}$$

$$B_{0i} = \frac{(\sigma_0 V_i)^2 + (\Omega_i W)^2}{(1 + \sigma_0^2 \Omega_i^2)^2}$$

$$B_{1i} = \frac{2\sigma_0 V_i}{1 + \sigma_0^2 \Omega_i^2}$$

Ho :

$$= \sigma_0 \prod_{i=1}^{r} \frac{B_{0i}}{A_{0i}} \qquad \text{para N impar}$$

$$= 10^{-0.05A} \prod_{i=1}^{r} \frac{B_{0i}}{A_{0i}} \qquad \text{para N par.}$$

Las series mostradas en algunas fórmulas convergen rápidamente, y si evaluamos tres o cuatro términos será suficiente para nuestros propósitos. Este desarrollo se optimizará con la simulación de un programa.

REFERENCIAS:

[1.1] L.WEINBERG, Network Analysis and Synthesis , McGraw-Hill New York, 1962.

[1.2] A.V.OPPENHEIM AND R.W.SCHAFER, Digital Signal Processing Prentice-Hall, 1975.

[1.3] A.ANTONION, Digital Filters: Analysis and Design , McGraw-Hill Book Company, New York, 1979.

CAPITULO 2

CONVERSION DE FILTROS RECURSIVOS ANALOGICOS EN DIGITALES

[2.1] Cuando deseamos conseguir un filtro digital, uno de los caminos mejores a seguir, es obtener el filtro digital a partir de un filtro analógico. Esto es cierto para filtros recursivos.

Esta operación se la realiza por las siguientes razones:

a) El arte del diseño de los filtros analógicos esta altamente avanzado y es por lo tanto ventajoso utilizar los procedimientos de diseño ya desarrollados para filtros analógicos.

b) Muchos métodos de diseño analógicos tienen fórmulas de diseño relativamente simples que facilitan el correspondiente diseño de filtros digitales.

c) En muchas aplicaciones es interesante usar filtros digitales para estudiar el desempeño de un filtro analógico.

Existen algunos métodos de conversión de filtros analógicos en digitales, los cuales los estudiaremos detalladamente. Antes de internarnos al estudio de la conversión de los filtros, indicaremos como se caracteriza cada uno de los filtros:

Caracterización de un filtro en el dominio analógico:

a) En la frecuencia:

La función de transferencia es:

$$H_a(s) = \frac{\sum_{k=0}^{M} f_k s^k}{\sum_{k=0}^{N} C_k s^k} = \frac{Y_a(s)}{X_a(s)}$$

donde:

- "s" es la frecuencia compleja en el dominio analógico, y es igual a: $s = \sigma + j\Omega$
- $H_a(s) = £\{ h(t) \}$
- $X_a(s) = £\{ x(t) \}$
- $Y_a(s) = £\{ y(t) \}$

£{} es la transformada de la Laplace y:

- h(t) es la respuesta impulsiva del filtro.
- x(t) es la señal de entrada
- y(t) es la señal de salida

b) En el tiempo:

Integral de convolución

$$y_a(t) = \int_{-\infty}^{\infty} x_a(\tau)\, h(t-\tau)\, d\tau$$

Ecuación diferencial

$$\sum_{k=0}^{N} C_k \frac{d^k y_a(t)}{dt^k} = \sum_{k=0}^{M} f_k \frac{d^k x_a(t)}{dt^k}$$

Caracterización de un filtro en el dominio digital:

a) En la frecuencia:

La función de transferencia es:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{Y(z)}{X(z)}$$

donde:

- $H(z) = Z\{h(n)\}$

- $Y(z) = Z\{y(n)\}$
- $X(z) = Z\{x(n)\}$

$Z\{\}$ es la transformada zeta.

b) En el tiempo:

Suma de convolución

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$$

Ecuación de diferencias:

$$\sum_{k=0}^{N} a_k\, y(n-k) = \sum_{k=0}^{M} b_k\, x(n-k)$$

Condiciones universalmente válidas para la transformación de un filtro analógico en digital.

1)

"s" es la variable analógica

"z" es la variable discreta

donde:

$z = e^{sT}$

En virtud de esta transformación de variables, el eje imaginario del plano complejo "s", se debe transformar en el círculo unitario en el plano z.

$$\text{Sea:}\quad s = j\Omega \quad\rightarrow\quad z = e^{j\Omega T} = \cos(\Omega T) + j\,\mathrm{sen}(\Omega T)$$

$$|z| = [\cos^2(\Omega T) + \mathrm{sen}^2(\Omega T)]$$

$$\theta = \mathrm{arctg}[\mathrm{sen}(\Omega T)/\cos(\Omega T)] = \Omega T$$

$$z = |z|\angle\theta = 1\angle\Omega T$$

efectivamente, si $-\infty < \Omega < \infty$, en el plano complejo z se tienen puntos alrededor del círculo unitario. Específicamente tendremos un círculo para cada intervalo de Ω, de 2π.

2)

"Un filtro analógico estable debe transformarse en un filtro digital estable".

Esto implica que si un filtro analógico tiene sus polos en el semiplano izquierdo, debe transformarse en un filtro digital que tenga los polos dentro del círculo unitario.

A continuación estudiaremos cuatro métodos de conversión de filtros analógicos en digitales:

2.1 CONVERSION POR INVARIANCIA DE IMPULSO

Este método se basa en que, dado un filtro analógico cuya respuesta impulsiva es $h_a(t)$, la respuesta impulsiva del filtro digital se obtiene tomando muestras de $h_a(t)$ a un intervalo de T. Sea h(n) la respuesta impulsiva del filtro digital, entonces:

$$h(n) = h_a(nT), \text{ donde } n = 0, 1, 2, \ldots$$

De esta manera podemos decir que la respuesta impulsiva del filtro digital mantiene la misma forma que la respuesta impulsiva del filtro analógico. Por lo tanto el filtro digital adquirirá las características del filtro analógico, debido a que mantiene la respuesta impulsiva invariante.

En la figura 2.1 se ilustra la toma de muestras de la respuesta impulsiva.

Conocida la respuesta impulsiva del filtro digital podemos determinar la función de transferencia.

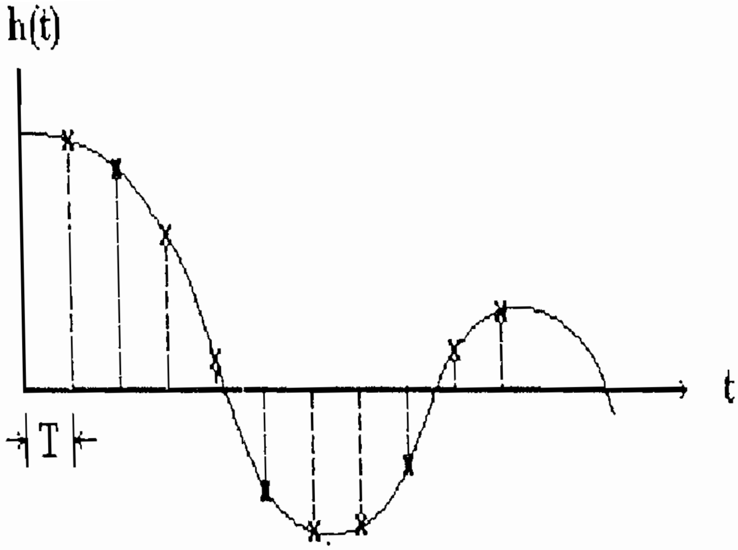


Figura 2.1: Toma de muestras de la
Respuesta Impulsiva

2.1.1 FORMULA DE TRANSFORMACION

A continuación determinaremos la función de transferencia del filtro digital en base a la función de transferencia de un filtro analógico.

Sea el filtro analógico cuya función de transferencia esta dada por:

$$H_a(s) = \sum_{k=1}^{N} \frac{A_k}{s-s_k} \tag{2.1}$$

este tiene polos en s_k del plano complejo, la respuesta impulsiva tendrá la forma :

$$h_a(t) = \sum_{k=1}^{N} A_k e^{s_k t} \mu(t)$$

donde $\mu(t)$ es la función paso. Esta respuesta se la encuentra utilizando la transformada inversa de la Laplace.

Aplicando la INVARIANCIA DE IMPULSO, tenemos que la respuesta de impulso del filtro digital sería:

$$h(n) = h_a(t)$$

$$h(n) = \sum_{k=1}^{N} A_k e^{s_k nT} \mu(n)$$

$$h(n) = \sum_{k=1}^{N} A_k (e^{s_k T})^n \mu(n)$$

En la figura 2.2 podemos ver la forma que tiene esta respuesta impulsiva del filtro digital.

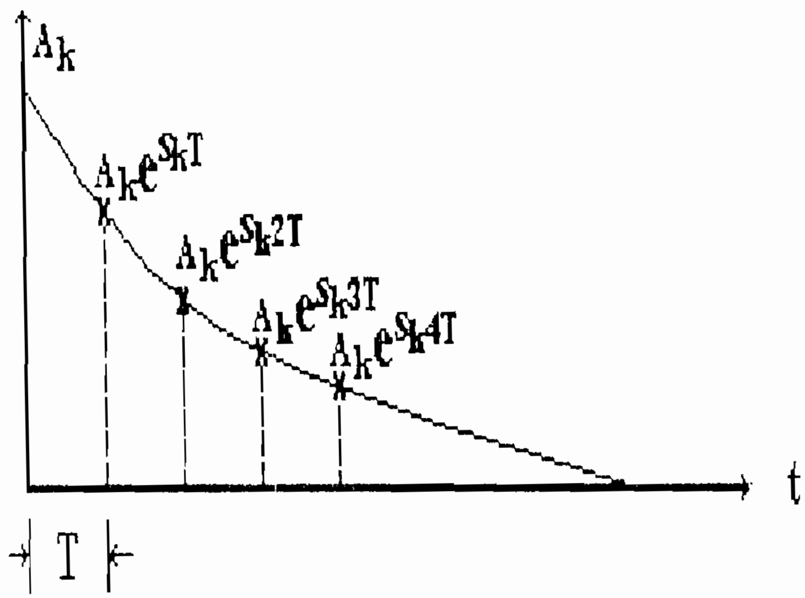


Figura 2.2: Respuesta Impulsiva del Filtro Digital

Para determinar la función de transferencia del filtro digital debemos obtener la transformada Z de la respuesta impulsiva h(n).

$$Z\{h(n)\} = \sum_{i=0}^{\infty} h(i) z^{-i}$$

$$Z\{\, h(n)\,\} = Z\{\sum_{k=1}^{N} A_k (e^{s_k T})^n\}$$

Aplicando la linealidad de la transformada Z tenemos:

$$Z\{\, h(n)\,\} = \sum_{k=1}^{N} Z\{\, A_k (e^{s_k T})^n\} \tag{2.2}$$

De la definición de la transformada Z y aplicando la fórmula para la suma de progresiones geométricas, tenemos:

$$\begin{aligned} Z\{\, A_k (e^{s_k T})^n\} &= \sum_{i=0}^{\infty} A_k (e^{s_k T})^i z^{-i} \\ &= \frac{A_k}{1-e^{s_k T} z^{-1}} \end{aligned}$$

reemplazando este resultado en 2.2 tenemos:

$$H(z) = Z\{\, h(n)\,\} = \sum_{k=1}^{N} \frac{A_k}{1-e^{s_k T} z^{-1}} \tag{2.3}$$

Este resultado nos lleva a concluir que los polos de la función de transferencia del filtro digital en el plano z se encuentran en $e^{s_k T}$.

En la práctica dada la función de transferencia de un filtro analógico, primero tenemos que transformar a la forma 2.1, para luego reemplazar el resultado 2.3.

Los resultados que se tendrán a continuación surgen de comparar las expresiones 2.1 y 2.3.

2.1.2 ALGUNOS RESULTADOS

Luego de la conversión de filtros hemos visto que la frecuencia analógica se transforma en ω, donde $\omega = \Omega T$, llamándose esta, frecuencia digital. Donde T es el intervalo de muestreo de la señal digital.

Así para este tipo de conversión, las frecuencias analógica y digital mantienen una relación lineal:

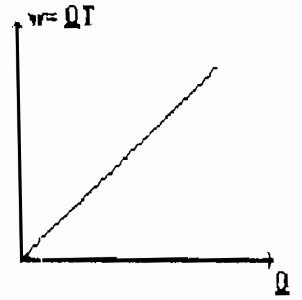


Figura 2.3: Relación Lineal de las Frecuencias Analógica y Digital

donde ω se expresa en [rad], Ω en [rad/seg] y T en unidades arbitrarias de tiempo.

El espectro de una señal digital debido al muestreo, tiene una repetición periódica con período $\omega = 2\pi$. Debemos saber que el punto $\omega = 2\pi$ corresponde a la frecuencia de muestreo, esto es:

$$\omega = 2\pi = \Omega T$$
$$2\pi = 2\pi f T$$
$$1 = fT$$

como $T = 1/fm$, donde fm es la frecuencia de muestreo, entonces:

$$f = fm$$

En la figura 2.4 se indica el espectro de una señal analógica, donde f_M es la frecuencia máxima del espectro.

Si esta señal es digitalizada con una frecuencia de muestreo fm, dado que el espectro es repetitivo cada 2π, podemos tener:

a) Cruce de espectros, si $fm < 2f_M$
b) Tangencia de espectros, si $fm = 2f_M$

c) Banda de separación entre espectros, si $f_m > 2f_M$

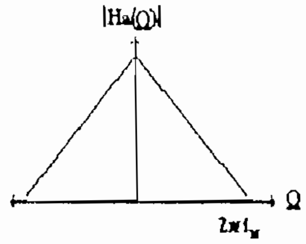


Figura 2.4: Respuesta de Frecuencia de una Señal Analógica

En el caso a), existirá una superposición de los espectros, ocasionando una grave distorsión. Este fenómeno se le conoce con el nombre de Superposición o Cruce de Espectros, "ALIASING". (Observar la figura 2.5)

Estos últimos resultados están basados en el " Teorema del Muestreo", que dice que para evitar el aliasing la frecuencia mínima de muestreo debe ser por lo menos 2 veces la frecuencia máxima contenida en la señal.

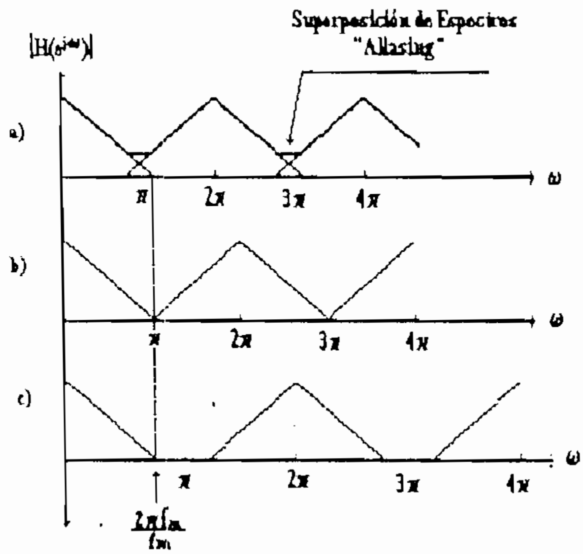


Figura 2.5: Posibles Espectros de una Señal Digital

La respuesta de frecuencia de un filtro analógico con frecuencia analógica Ω que va desde 0 hasta ∞, determina un espectro de frecuencia infinito.

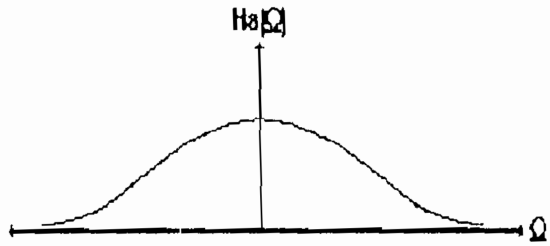


Figura 2.6: Respuesta de Frecuencia de un Filtro Analógico

Si a este filtro lo convertimos en digital, obviamente producirá cruce de espectros "aliasing".

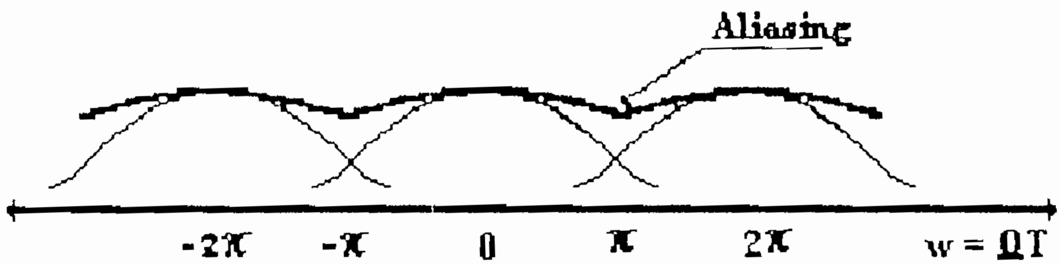
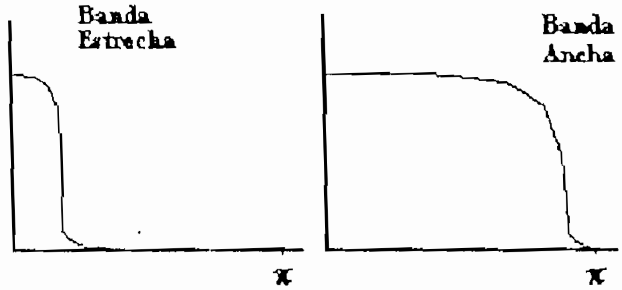


Figura 2.7: Cruce de Espectros, "Aliasing"

Este es un inconveniente que se tendrá al utilizar este tipo de conversión, por lo tanto es recomendable usar esta conversión para filtros de banda estrecha o limitada.



SI NO

Figura 2.8: Respuestas de Frecuencia de filtros Pasa-bajos

Por la misma razón se deberá tener cuidado al usar esta conversión en filtros elimina-banda.

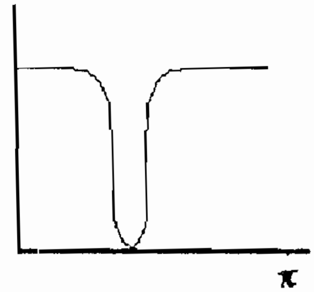


Figura 2.9: Respuesta de Frecuencia de un Filtro Elimina-banda

En la práctica veremos que el fenómeno del aliasing se presenta en forma específica para algunos casos.
En la figura 2.10 se muestra la respuesta de frecuencia de un filtro digital, y la relación que existe con el plano z.

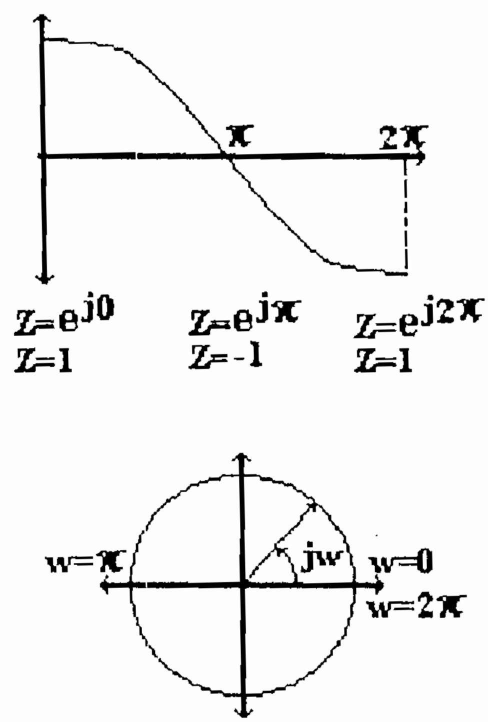


Figura 2.10: Respuesta de Frecuencia de un Filtro Digital y la relación con el Plano "Z"

El círculo unitario obtenido en el plano "z" a partir de la variación de Ω entre $-\pi T$ y πT en el plano "s", se lo ilustra en la figura 2.11.

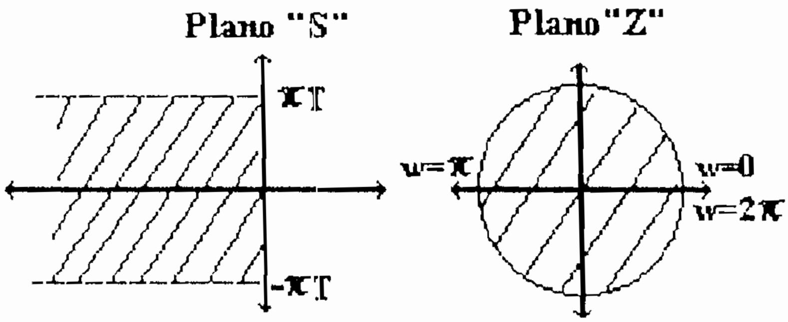


Figura 2.11: Obtención del Círculo Unitario en el Plano " Z "

La regla de que los puntos que se encuentren en el semiplano izquierdo del plano S, se ubicarán dentro del círculo unitario en el plano Z, es válida solo para los polos. Para los ceros no necesariamente se cumple. Por ejemplo:

Sea:

$$H_a(s) = \frac{s+a}{(s+a)^2+b^2}$$

la función de transferencia de un filtro analógico que tiene un cero en s=-a.

Descomponiendo en fracciones parciales:

$$H_a(s) = \frac{\frac{1}{2}}{s+aj+b} + \frac{\frac{1}{2}}{s+a-jb}$$

aplicando la regla de transformación a cada una de las fracciones parciales de primer grado, tenemos:

$$H_a(s) = \frac{\frac{1}{2}}{1-e^{-aT-jbT}z^{-1}} + \frac{\frac{1}{2}}{1-e^{-aT+jbT}z^{-1}}$$

$$= \frac{1}{2}\;\frac{1-e^{-aT+jbT}z^{-1}+1-e^{-aT-jbT}z^{-1}}{(1-e^{-aT-jbT}z^{-1})\,(1-e^{-aT+jbT}z^{-1})}$$

$$= \frac{1-e^{-aT}\cos(bT)\,z^{-1}}{(1-e^{-aT-jbT}z^{-1})\,(1-e^{-aT+jbT}z^{-1})}$$

Efectivamente este resultado nos hace notar que el cero aquí obtenido no obedece a la regla de transformación válida para los polos.

Estos resultados surgen de comparar las expresiones 2.1 y 2.3

2.2 CONVERSION BASADA EN LA SOLUCION NUMERICA DE ECUACIONES DIFERENCIALES.

Este método de conversión se basa en obtener la ecuación de diferencias que caracterice al filtro digital, a partir de una aproximación de la ecuación diferencial que caracteriza al filtro analógico.

Básicamente se trata de aproximar la derivada de una función a la pendiente que forman dos muestras consecutivas de esa función. Sea f(t) una función a la que se la ha digitalizado y tiene muestras cada intervalo T=1, f(nT). Entonces la derivada de f(t) evaluada en cualquier punto t=nT se la puede aproximar en diferencias de dos maneras:

<u>Retrospectiva:</u>

$$\frac{df(t)}{dt}/_{t=nT} \rightarrow \frac{f(n)-f(n-1)}{T} = \nabla^{(1)}[f(n)]$$

es decir que se ha restado el valor presente y el anterior. ∇ es la forma de notar este tipo de diferencias.

Prospectiva:

$$\frac{df(t)}{dt}/_{t=nT} \rightarrow \frac{f(n+1)-f(n)}{T} = \Delta^{(1)}[f(n)]$$

Siendo Δ el símbolo para este tipo de diferencias. En esta aproximación se restan los valores posterior y presente, esto puede hacer que el sistema sea inestable, por lo que no es recomendable su uso.

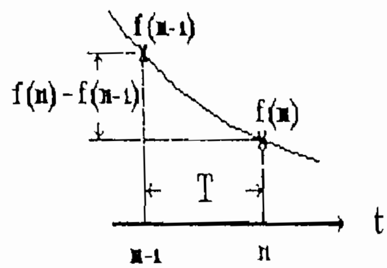


Figura 2.12: Aproximación de la derivada
de una función

En la figura 2.12 se puede observar en que consiste la aproximación y como será tanto mejor cuanto mas alta sea la frecuencia de muestreo, es decir, cuando $T \rightarrow 0$.

Este análisis se puede generalizar para derivadas de orden mayor:

$$\frac{d^k f(t)}{dt^k} = \frac{d}{dt}\left(\frac{d^{k-1}}{dt^{k-1}} f(t)\right) \rightarrow \nabla^{(k)}[f(n)] = \nabla^{(1)}[\nabla^{(k-1)}[f(n)]]$$

También, se define:

$$\nabla^{(0)}[f(n)] = f(n)$$

2.2.1 FORMULA DE TRANSFORMACION

Caracterizado un filtro analógico con su ecuación diferencial:

$$\sum_{k=0}^{N} C_k \frac{d^k y_a(t)}{dt^k} = \sum_{k=0}^{M} f_k \frac{d^k x_a(t)}{dt^k}$$

obtendremos la función de transferencia del filtro digital correspondiente.

En la ecuación diferencial reemplacemos las derivadas por diferencias:

$$\sum_{k=0}^{N} C_k \nabla^{(k)} [y(n)] = \sum_{k=0}^{M} f_k \nabla^{(k)} [x(n)]$$

donde $y(n) = y_a(nT)$ y $x(n) = x_a(nT)$.

Obteniendo la transformada Z:

$$Z\{\nabla^{(1)}[x(n)]\} = Z\{\frac{x(n)-x(n-1)}{T}\}$$

$$= (\frac{X(z)-z^{-1}X(z)}{T})$$

$$= (\frac{1-z^{-1}}{T})\; X(z)$$

Igualmente podemos generalizar:

$$Z\{\nabla^{(k)}[x(n)]\} = (\frac{1-z^{-1}}{T})^k X(z)$$

Aplicando esta relación de transformación:

$$\sum_{k=1}^{N} C_k (\frac{1-z^{-1}}{T})^k Y(z) = \sum_{k=1}^{M} f_k (\frac{1-z^{-1}}{T})^k X(z)$$

donde $Z\{ y(n) \} = Y(z)$ y $Z\{ x(n) \} = X(z)$.

Por lo tanto:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=1}^{M} f_k(\frac{1-z^{-1}}{T})^k}{\sum_{k=1}^{N} C_k(\frac{1-z^{-1}}{T})^k} \tag{2.4}$$

Comparando la función de transferencia del filtro analógico:

$$H_a(s) = \frac{\sum_{k=0}^{M} f_k s^k}{\sum_{k=0}^{N} C_k s^k}$$

con el H(z) encontrado con este método de conversión, se deduce la siguiente relación de transformación:

$$s \rightarrow \frac{1-z^{-1}}{T}$$

$$z = \frac{1}{1-sT}$$

para analizar la transformación que sufre el eje imaginario, sea s=jΩ, empleando la relación de transformación:

$$z = \frac{1}{1-j\Omega T} \tag{2.5}$$

Se demostrará que la misma no transforma el eje imaginario en el círculo unitario en el plano z.

2.2.2 DESARROLLO DE LA TRANSFORMACION DEL EJE jΩ DEL PLANO S EN EL PLANO Z.

Haciendo algunos artificios en la expresión 2.5, tenemos:

$$\begin{aligned} z &= \frac{1}{2}[\frac{1+1+j\Omega T-j\Omega T}{1-j\Omega T}] \\ &= \frac{1}{2}[\frac{1-j\Omega T+1+j\Omega T}{1-j\Omega T}] \\ &= \frac{1}{2}[1+\frac{1+j\Omega T}{1-j\Omega T}] \end{aligned} \tag{2.6}$$

por otra parte 2.5 en forma polar es:

$$z = \frac{1}{\sqrt{1+(\Omega T)^2}} \angle arctg(\Omega T)$$

Sea $\theta = arctg(\Omega T) \rightarrow \Omega T = tg(\theta)$

Si este resultado reemplazamos en la siguiente identidad trigonométrica:

$$tg(2\theta) = \frac{2tg(\theta)}{1-tg^2(\theta)}$$

$$\rightarrow \quad tg(2\theta) = \frac{2\Omega T}{1-(\Omega T)^2} \tag{2.7}$$

Ahora sea:

$$e^{j2\theta} = \cos(2\theta) + j\ sen(2\theta) \tag{2.8}$$

reemplazando las siguientes identidades con el resultado 2.7, en 2.8, tenemos:

Identidades:

$$sen(2\theta) = \frac{tg(2\theta)}{\sqrt{1+tg^2(2\theta)}}$$

$$\cos(2\theta) = \frac{1}{\sqrt{1+tg^2(2\theta)}}$$

$$e^{j2\theta} = \frac{1}{\sqrt{1+\left(\frac{2\Omega T}{1-(\Omega T)^2}\right)^2}} + j\frac{\frac{2\Omega T}{1-(\Omega T)^2}}{\sqrt{1+\left(\frac{2\Omega T}{1-(\Omega T)^2}\right)^2}}$$

simplificando esta expresión:

$$e^{j2\theta} = \frac{1+j\frac{2\Omega T}{1-(\Omega T)^2}}{\sqrt{\frac{1-2(\Omega T)^2+(\Omega T)^4+4(\Omega T)^2}{[1-(\Omega T)^2]^2}}}$$

$$e^{j2\theta} = \frac{1+j\frac{2\Omega T}{1-(\Omega T)^2}}{\frac{1+(\Omega T)^2}{1-(\Omega T)^2}}$$

$$= \frac{1-(\Omega T)^2+j2\Omega T}{1+(\Omega T)^2}$$

$$= \frac{(1+j\Omega T)^2}{(1+j\Omega T)(1-j\Omega T)}$$

por lo tanto:

$$e^{j2\theta} = \frac{1+j\Omega T}{1-j\Omega T}$$

reemplacemos este resultado en 2.6, entonces:

$$z = \frac{1}{2}[1+e^{j2\theta}]$$

$$= \frac{1}{2}[1+e^{j2\operatorname{arctg}(\Omega T)}]$$

Con este resultado podemos decir que el eje imaginario en el plano s, se transforma en una circunferencia de radio ½, desplazada hacia la derecha sobre el eje real la distancia de ½.

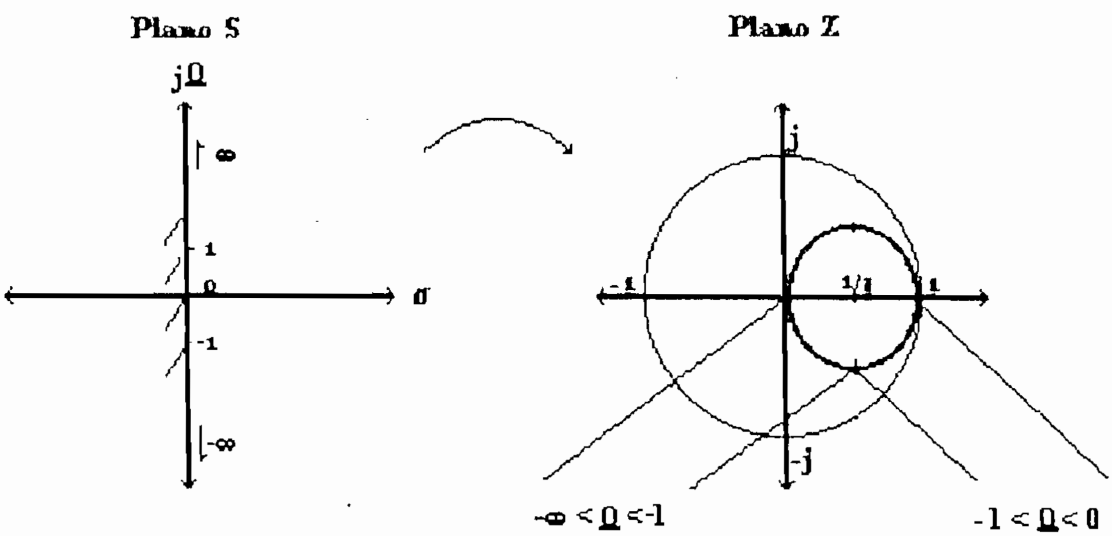


Figura 2.13: Transformación del eje "jΩ"

Ecuación del círculo en el plano x, y:

$$(x - \tfrac{1}{2})^2 + y^2 = \tfrac{1}{4} \quad \rightarrow \quad \begin{array}{l} \text{centro: } (\tfrac{1}{2},\ 0) \\ \text{radio: } \tfrac{1}{2} \end{array}$$

De hecho también el semiplano izquierdo del plano s se transforma en el interior del círculo de radio ½ en el plano z. Esto garantiza que la estabilidad del filtro se mantenga, ya que el círculo de radio ½ está dentro del círculo unitario. Cuanto mas cerca estemos del círculo unitario, es decir en la zona $z = 1 + j0$, el filtro tiene un buen comportamiento, y esto se logra con una frecuencia alta de muestreo.

2.2.3 ALGUNAS CONCLUSIONES.

1) Este tipo de transformación será tanto mejor cuanto mas alta sea la frecuencia de muestreo, a través de la cual se obtienen las diferencias retrospectivas.

2) No se conserva la relación lineal entre la frecuencia analógica y digital, es decir produce un ALABEO de frecuencia.

$$w = 2 \operatorname{arctg} (\Omega T)$$

3) Solo puede utilizarse para filtros pasa-bajos de banda limitada.

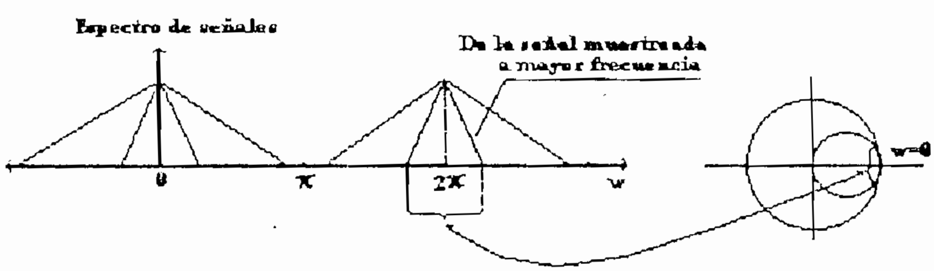


Figura 2.14: Transformación del Espectro al Plano " Z "

No es que se reduce el espectro, sino que con relación a la distancia de 0 á 2π, se ve reducida.

En la figura 2.14 se puede ver que cuando mas alta es la frecuencia de muestreo, los puntos transformados del eje $j\Omega$, caen mas cerca en una zona mas próxima al círculo unitario.

2.3 CONVERSION POR TRANSFORMADA BILINEAL

Este tipo de conversión se obtiene por la solución numérica de la ecuación diferencial a través de una integración numérica por el método de los trapecios.

La integración numérica por el método de los trapecios para cualquier función f(t), consiste en determinar el área bajo la curva comprendida entre dos rectas verticales separadas por una distancia T. La figura formada por estos límites, de la cual se calculará el área, la aproximaremos a un trapecio.

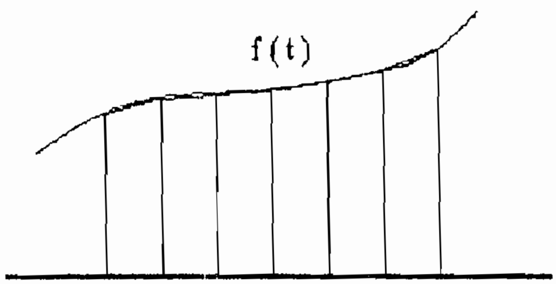


Figura 2.15: Función f(t) dividida en Trapecios

Sea $y_a'(t)$ la función a integrar, de la definición de integrales tenemos:

$$y_a(t) = \int_0^t y_a'(t)\,dt + y_a(0)$$

De esta manera la integral definida entre los límites: " nT " y " (n-1)T ", será igual.

$$\int_{(n-1)T}^{nT} y_a'(t)\,dt = y_a(nT) - y_a[(n-1)T] \qquad (2.9)$$

La aproximación de esta integral, determinada por el área del trapecio (vea figura 2.16), es:

$$\int_{(n-1)T}^{nT} y_a'(t)\,dt = \left(\frac{y_a'[(n-1)T] + y_a'(nT)}{2}\right) T$$

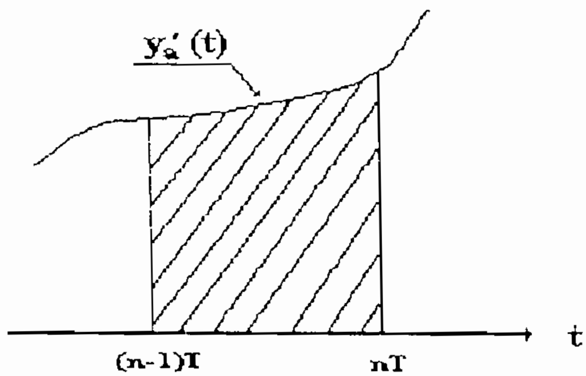


Figura 2.16: Aproximación de la Integral al Area del Trapecio

Esta aproximación remplazándola en 2.9 y despejando el término $y_a(nT)$, tendremos:

$$y_a(nT) = y_a[(n-1)T] + \left(\frac{y_a'[(n-1)T] + y_a'(nT)}{2}\right) T \qquad (2.10)$$

2.3.1 FORMULA DE TRANSFORMACION

Sea un filtro analógico cuya ecuación diferencial y función de transferencia son:

$$C_1\, y_a'(t) + C_0\, y_a(t) = d_0\, x_a(t)$$

$$y_a'(t) = -\frac{C_0}{C_1} y_a(t) + \frac{d_0}{C_1} x_a(t)$$

$$H_a(s) = \frac{d_0}{C_1 s + C_0} \qquad (2.11)$$

donde $y_a(0) = 0$.

Si la ecuación diferencial la remplazamos en el resultado 2.10 obtenido de la integración numérica, obtendremos:

$$y_a(nT) = y_a[(n-1)T] + \frac{T}{2}\Big[-\frac{C_0}{C_1} y_a[(n-1)T] + \frac{d_0}{C_1} x_a[(n-1)T] - \frac{C_0}{C_1} y_a(nT) + \frac{d_0}{C_1} x_a(nT) \Big]$$

En esta expresión haremos el cambio de la función analógica en la digital así:

$$y_a(nT) = y(n)$$
$$y_a'(nT) = y'(n)$$
$$y_a'[(n-1)T] = y'(n-1)$$

$$y(n) = y(n-1) + \frac{T}{2}\Big[-\frac{C_0}{C_1} y(n-1) + \frac{d_0}{C_1} x(n-1) - \frac{C_0}{C_1} y(n) + \frac{d_0}{C_1} x(n) \Big]$$

Tomando la transformada Z:

$$Y(z) = Z^{-1} Y(z) + \frac{T}{2}\Big[-\frac{C_0}{C_1} Y(z)\,(1+z^{-1}) + \frac{d_0}{C_1} X(z)\,(1+z^{-1}) \Big]$$

trabajando un poco en la expresión, tenemos:

$$Y(z) - Z^{-1} Y(z) + \frac{T}{2}\Big[-\frac{C_0}{C_1} Y(z)\,(1+z^{-1}) \Big] = \frac{T}{2}\Big[\frac{d_0}{C_1} X(z)\,(1+z^{-1}) \Big]$$

$$\frac{Y(z)}{X(z)} = \frac{\frac{Td_0}{2C_1}(1+z^{-1})}{1-z^{-1}+\frac{TC_0}{2C_1}(1+z^{-1})}$$

dividiendo el numerador y el denominador para:

$$(1+z^{-1})\frac{T}{2C_1}$$

Entonces la función de transferencia del filtro digital convertido será:

$$H(z) = \frac{d_0}{C_1\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}+C_0} \tag{2.12}$$

Comparando la expresión 2.12 con 2.11, se deduce que la relación de transformación para este tipo de conversión es:

$$s = \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})} \tag{2.13}$$

Se puede demostrar que esta relación de transformación es válida para una ecuación diferencial de grado n-ésimo.

2.3.2 ALABEO DE FRECUENCIA

Al igual que en el caso de la conversión basada en la solución numérica de ecuaciones diferenciales, en este tipo de conversión también se produce alabeo de frecuencia. Esto lo podemos demostrar reemplazando en 2.13, z por $e^{j\omega}$.

$$s = \frac{2}{T}\frac{(1-e^{-j\omega})}{(1+e^{-j\omega})}$$

$$= \frac{2}{T}\frac{e^{-j\omega/2}(e^{j\omega/2}-e^{-j\omega/2})2j}{e^{-j\omega/2}(e^{j\omega/2}+e^{-j\omega/2})2j}$$

$$= \frac{2}{T}j\frac{sen(\omega/2)}{\cos(\omega/2)}$$

$$= \frac{2j}{T}tg(\omega/2)$$

Para el eje imaginario: $\sigma = 0$

$$s = j\Omega = j(2/T).tg(\omega/2)$$

$$\rightarrow \quad \Omega = (2/T)\ tg(\omega/2)$$

de esta expresión despejemos la frecuencia digital ω:

$$\omega = 2\ arctg(\Omega T/2)$$

Este alabeo de frecuencia es ventajoso porque evita que se crucen los espectros, pero produce grave distorsión en la fase del filtro y por esto la transformada bilineal no se la puede utilizar para transformar al dominio digital un filtro analógico de fase lineal.

En la figura 2.17 se muestra que el espectro de un filtro pasa-bajos en el dominio analógico por mas que se extienda hacia el infinito, luego de la conversión en el dominio digital nunca se excederá de $\omega = \pi$. La zona de mayor alabeo es la siguiente:

$$\Omega s < \Omega < \infty \qquad \rightarrow \qquad \omega_s < \omega < \pi$$

Esto garantiza que la conversión con la transformación bilineal no produzca cruce de espectros ("aliasing"). Por esta razón este tipo de conversión no tiene limitaciones en cuanto a la forma del espectro del filtro analógico a convertir.

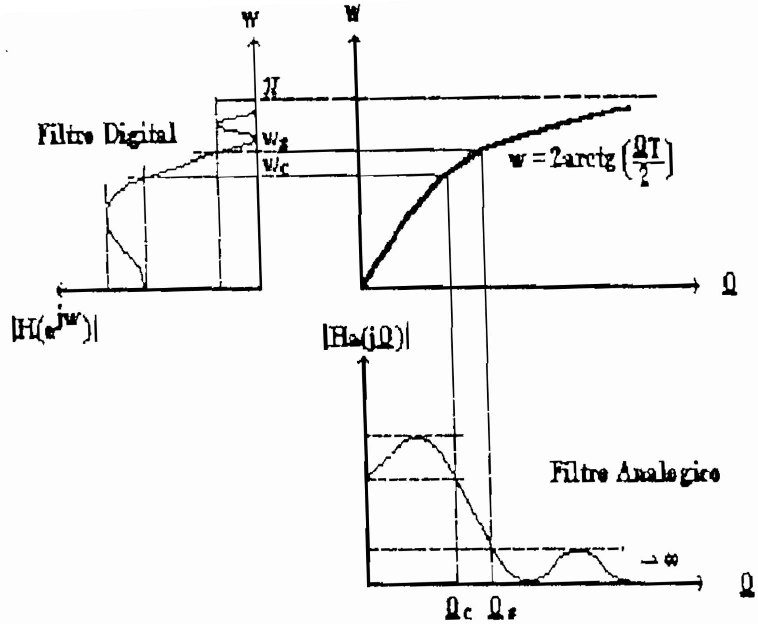


Figura 2.17: Alabeo de Frecuencia

Para transformar un filtro analógico en un filtro digital utilizando la transformada bilineal, se debe previamente PREALABEAR la frecuencia del filtro analógico. Esto garantizará que las especificaciones del filtro analógico se transmitan al filtro digital.

Por decir, sea $\Omega c'$ la frecuencia de corte del filtro analógico, entonces para encontrar la frecuencia prealabeada utilizamos la función inversa de la función que produce el alabeo.

$$\Omega c = (2/T).tg(\Omega c'/2)$$

2.3.3 DESARROLLO DE LA TRANSFORMACION DEL EJE $j\Omega$ DEL PLANO S EN EL PLANO Z.

De la expresión 2.13 vamos a despejar la variable z, entonces:

$$z = \frac{1+\frac{sT}{2}}{1-\frac{sT}{2}}$$

los puntos que están sobre el eje imaginario se caracterizan por: s = jΩ, luego reemplazando:

$$z = \frac{1+\frac{j\Omega T}{2}}{1-\frac{j\Omega T}{2}}$$

si a este resultado le representamos en forma polar el numerador y del denominador, tenemos:

$$z = \frac{\sqrt{1+(\frac{\Omega T}{2})^2}}{\sqrt{1+(\frac{\Omega T}{2})^2}} \frac{\underline{/arctg(\Omega T/2)}}{\underline{/-arctg(\Omega T/2)}}$$

por lo tanto:

$$z = 1 \; \underline{/2.arctg(\Omega T/2)}$$

este resultado nos conduce a obtener en el plano z el circulo unitario.

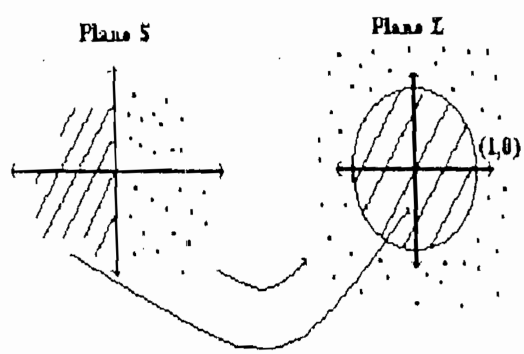


Figura 2.18: Transformación del Plano "S" al "Z"

De hecho todo el semiplano izquierdo se transforma en el interior del círculo unitario:

$$z = \frac{1+\frac{T}{2}(\sigma+j\Omega)}{1-\frac{T}{2}(\sigma+j\Omega)}$$

la condición $\sigma<0$ caracteriza al semiplano izquierdo:

$$z = \frac{1-\frac{T}{2}|\sigma|+\frac{T}{2}j\Omega}{1+\frac{T}{2}|\sigma|+\frac{T}{2}j\Omega}$$

sacando el módulo de z tenemos:

$$|z| = \frac{\sqrt{(1-\frac{T}{2}|\sigma|)^2+(\frac{T}{2}\Omega)^2}}{\sqrt{(1+\frac{T}{2}|\sigma|)^2+(\frac{T}{2}\Omega)^2}}$$

en esta expresión el denominador siempre será mayor que el numerador, por lo tanto $|z|<1$, es decir está dentro del círculo unitario.

En el caso contrario, el semiplano derecho caracterizado por $\sigma>0$ se transforma en el exterior del círculo unitario.

$$\sigma > 0 \quad \rightarrow \quad |z| = \frac{\sqrt{(1+\frac{T}{2}|\sigma|)^2+(\frac{T}{2}\Omega)^2}}{\sqrt{(1-\frac{T}{2}|\sigma|)^2+(\frac{T}{2}\Omega)^2}}$$

aquí el numerador siempre será mayor que el denominador, por lo tanto $|z|>1$.

2.4 CONVERSION POR INVARIANCIA DE PULSO [2.2]

Este método se basa en que dada la respuesta a la función pulso y(t) de un filtro analógico, se establece como respuesta a la función pulso digital del filtro digital a convertirse, a las muestras tomadas en y(t) cada intervalo T.

De esta manera podemos decir que las respuestas a la función pulso del filtro analógico y la del filtro digital, se mantienen invariantes.

La función pulso se la define como:

$$x(t): \quad \begin{cases} 1 & 0 \le t < T_d \\ 0 & t < 0;\ t \ge T_d \end{cases}$$

Debemos aclarar que Td en este caso es el intervalo de duración en el que la función pulso es igual a 1, muy distinto al intervalo de muestreo de la señal digital T. La relación que estos tienen es:

$$T_d = M.T$$

donde M es un número entero que representa al número de muestras tomadas dentro del intervalo T. Así la forma digital de la función pulso, será:

$$x(n): \quad \begin{cases} 1 & 0 \le n < M \\ 0 & n < 0;\ n \ge M \end{cases}$$

En la figura 2.19 se ilustra x(t) y x(n).

Las ventajas que tiene el elegir a la función pulso como la señal de entrada, en nuestro análisis son:

- Un pulso rectangular con esta descripción, aparece efectivamente en muchos sistemas de comunicación. De esta manera es más real y exacta la elección.

- El resultado de la transformación es simple y fácilmente aplicable. De hecho la forma simple del pulso conduce a una simple resolución matemática en la derivación de la transformación.

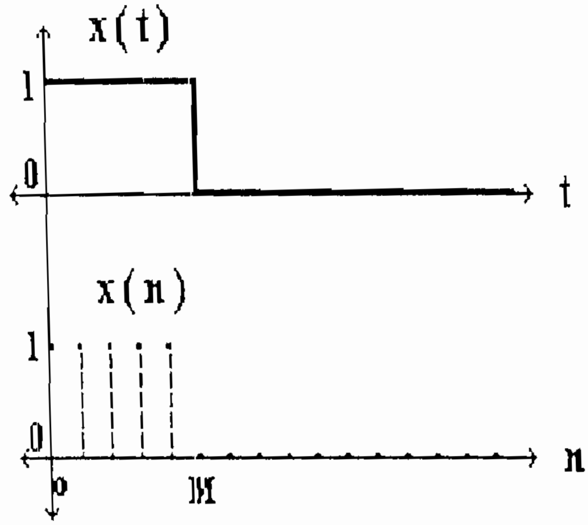


Figura 2.19: Función Pulso Analógica y Digital

Las desventajas serían que :

- Si el actual sistema simulado, usa una forma de pulso diferente, no es evidente que esta transformación conduzca a la mas exacta simulación.
- Un pulso rectangular no tiene banda limitada, así que inevitablemente estará presente el fenómeno aliasing.

2.4.1 FORMULA DE TRANSFORMACION.

De igual manera, a partir de la función de transferencia de un filtro analógico, llegaremos a determinar la función de transferencia equivalente para un filtro digital.

Sea la función de transferencia del filtro analógico igual a (2.1):

$$H_a(s) = \sum_{k=1}^{N} \frac{A_k}{s-s_k}$$

a partir de ésta podemos encontrar la respuesta impulsiva $h_a(t)$, aplicando la transformada inversa de Laplace. Por tanto:

$$h_a(t) = \sum_{k=1}^{N} A_k e^{s_k t} \mu(t)$$

Por otro lado la respuesta de un filtro está dada por la integral de convolución:

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h_a(t-\tau)\, d\tau$$

Reemplazando en esta integral, x(t) igual a la función pulso, tenemos:

$$y(t) : \begin{cases} \int_0^T h_a(t-\tau)\, d\tau & t \geq T \\ \int_0^t h_a(t-\tau)\, d\tau & 0 \leq t \leq T \\ 0 & t < 0 \end{cases}$$

efectivamente la última línea de este resultado, expresa la causalidad del filtro.

Si reemplazamos el valor de $h_a(t)$, tenemos:

$$y(t) : \begin{cases} \int_0^{T_d} \sum_{k=1}^{N} A_k e^{s_k (t-\tau)}\, d\tau & t \geq T_d \\ \int_0^{t} \sum_{k=1}^{N} A_k e^{s_k (t-\tau)}\, d\tau & 0 \leq t \leq T_d \end{cases}$$

estas integrales son iguales a:

Para $t \geq T_d$

$$y(t) = \sum_{k=1}^{N} A_k e^{s_k t} \int_0^{T_d} (e^{-s_k \tau})\, d\tau$$

$$= \sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_k t} (e^{-s_k T_d} - e^{0})$$

$$= \sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_k t} (e^{-s_k T_d} - 1) \tag{2.14}$$

Para $0 \leq t \leq T_d$, se tiene un resultado parecido al anterior solo que los límites de la integral son distintos:

$$y(t) = \sum_{k=1}^{N} A_k e^{s_k t} \int_0^{t} (e^{-s_k \tau})\, d\tau$$

$$= \sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_k t} (e^{-s_k t} - 1) \tag{2.15}$$

Aplicando la invariancia de pulso

$$y(n) = y_a(nT)$$

a los resultados 2.14 y 2.15, tenemos:

$$y(n) = \sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_k nT} (e^{-s_k MT} - 1) \qquad n \geq M$$

$$y(n) = \sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_k nT} (e^{-s_k nT} - 1) \qquad 0 \leq n < M$$

El proceso a seguir para obtener la función de transferencia será tomar la transformada Z de y(n) y x(n), para luego relacionarles así:

$$Y(z) \,/\, X(z)$$

sabiendo que efectivamente este resultado es la función de transferencia del filtro digital.

Entonces:

a) Para y(n)

$$Y(z) = \sum_{n=0}^{\infty} y(n)\; z^{-n}$$

$$Y(z) = \sum_{n=0}^{M-1}\sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_kTn}(e^{-s_kTn}-1)\; z^{-n} + \sum_{n=M}^{\infty}\sum_{k=1}^{N} \frac{A_k}{-s_k} e^{s_kTn}(e^{-s_kTM}-1)\, z^{-n}$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} - \sum_{n=0}^{M-1} e^{s_kTn}\; z^{-n} + \sum_{n=M}^{\infty} e^{s_kT(n-M)} z^{-n} - \sum_{n=M}^{\infty} e^{s_kTn} z^{-n} \right]$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + \sum_{n=M}^{\infty} e^{s_kT(n-M)} z^{-n} - \sum_{n=0}^{\infty} e^{s_kTn} z^{-n} \right]$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + e^{-s_kTM}\left(\sum_{n=0}^{\infty} e^{s_kTn} z^{-n} - \sum_{n=0}^{M-1} e^{s_kTn} z^{-n} \right) - \sum_{n=M}^{\infty} e^{s_kTn} z^{-n} \right]$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + (e^{-s_kTM}-1) \sum_{n=0}^{\infty} e^{s_kTn} z^{-n} - e^{-s_kTM} \sum_{n=0}^{M-1} e^{s_kTn} z^{-n} \right]$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + (e^{-s_kTM}-1) \sum_{n=0}^{\infty} e^{s_kTn} z^{-n} - e^{-s_kTM} \frac{(e^{s_kT} z^{-1})^{M}-1}{e^{s_kT} z^{-1}-1} \right]$$

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + (e^{-s_kTM}-1) \sum_{n=0}^{\infty} e^{s_kTn} z^{-n} + \frac{z^{-M}-e^{-s_kTM}}{1-e^{s_kT} z^{-1}} \right]$$

reemplazando la siguiente equivalencia:

$$\frac{1}{1 - e^{s_kT} z^{-1}} = \sum_{n=0}^{\infty} e^{s_kTn} z^{-n}$$

tenemos:

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[\sum_{n=0}^{M-1} z^{-n} + \frac{z^{-M}-1}{1-e^{s_kT} z^{-1}} \right]$$

b) Para x(n)

$$X(z) = \sum_{n=0}^{\infty} x(n)\, z^{-n}$$

$$X(z) = \sum_{n=0}^{M-1} z^{-n}$$

$$X(z) = \frac{z^{-M}-1}{z^{-1}-1}$$

Entonces con estos resultados ya podemos tener la función de transferencia:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[1 + \frac{z^{-1}-1}{1-e^{s_k T} z^{-1}} \right]$$

$$H(z) = \sum_{k=1}^{N} \frac{A_k}{-s_k} \left[-\frac{z^{-1}(1 - e^{s_k T})}{1-e^{s_k T} z^{-1}} \right] \qquad (2.16)$$

2.4.2 ALGUNOS RESULTADOS.

Los resultados siguientes han sido conocidos a partir de comparar las funciones 2.3 y 2.16. Podríamos decir que esta nueva transformación es una modificación del método de la invariancia de impulso.

Como vemos los polos de ambos métodos son idénticos, entonces podemos decir que con esta nueva trasformación también se mantiene la estabilidad de los filtros luego de la conversión. De hecho los polos del plano izquierdo del plano s, se trasforman en polos ubicados dentro del circulo unitario en el plano z.

Al respecto de los ceros, no son los mismos para las dos funciones de transferencia, sin embargo si hacemos $|s_k T| << 1$, entonces tenemos:

$$1 - e^{s_k T} \approx -s_k T$$

quedando en el numerador de cada función parcial el siguiente término:

$$A_k \ T \ z^{-1}$$

de esa manera estos ceros han sido cuidadosamente aproximados a los ceros de la función de transferencia 2.3, para valores pequeños de T.

Como vemos el término z^{-1} está presente en el numerador, simplemente es un retardo que retraza la señal pero sin alterarla. La respuesta de un filtro a cualquier entrada no singular debe ser cero para el tiempo $t=0^{+}$. De ahí que $y(n=0)$ debe ser cero y esta correspondencia es impuesta por el término z^{-1} en el numerador.

Luego de la conversión del filtro la ganancia permanece igual. Esto podemos mostrar reemplazando $s=0$ en 2.1 y $z=1$ en 2.16, por lo tanto:

$$Ha(0) = \sum_{k=1}^{N} \frac{A_k}{-s_k} = H(1)$$

Sea $s_k = \alpha_k + j\Omega_k$, la transformación de estos polos al plano z es $z_k = e^{s_k T}$. Si $\Omega_k T$ es bastante grande los polos en el plano z pueden ser aproximados a pequeños ángulos correspondientes a valores pequeños de $\Omega_k T$. Esta aproximación se hará a no ser que T sea seleccionado lo suficientemente pequeño.
Para evitar el aliasing es necesario que $\Omega_k T < \pi$.

REFERENCIAS:

[2.1] A.V.OPPENHEIM AND R.W.SCHAFER, Digital Signal Processing Prentice-Hall, 1975.

[2.2] FLOYD M. GARDNER, "A Transformation for Digital Simulation of Analog Filters", IEEE Transaction on Comunications , Vol COM-34, NO.7, July 1986.

CAPITULO 3

TRANSFORMACIONES DEL FILTRO PASA-BAJOS A OTRO TIPO DE FILTRO

[3.1] Hasta aquí nuestro desarrollo ha pretendido analizar el diseño de filtros digitales pasa-bajos normalizados a partir de los correspondientes filtros analógicos.

En este capítulo queremos hacer aplicable los resultados obtenidos a cualquier tipo de filtros.

Cuando se va a diseñar un filtro lo primero que debemos saber es si este filtro es pasa-bajos, pasa-altos, pasa-banda o elimina-banda. Luego es necesario conocer las frecuencias de corte y los valores de las atenuaciones correspondientes.

Para hacer posible la transformación de un filtro pasa-bajos en el filtro requerido, podemos utilizar los métodos de transformación, de los cuales tenemos: Transformación Analógica-Analógica y Transformación Digital-Digital.

Se puede realizar el diseño del filtro digital, a partir de los filtros analógicos normalizados pasa-bajos. Un filtro analógico normalizado pasa-bajos cumple con:

a) Para los filtros de Butterworth la frecuencia de corte a -3dB debe ser igual a 1 [rad/seg].
b) Para los filtros de Chevishev la frecuencia de corte de la banda de paso Ωp debe ser igual a 1 [rad/seg].
c) Para los filtros Elípticos las frecuencias de corte cumplen con:

$$\sqrt{(\Omega p . \Omega s)} = 1 \text{ [rad/seg]}$$

Para los filtros Elípticos es indispensable el empleo del filtro analógico normalizado pasa-bajos.

En cambio, un filtro <u>digital normalizado</u> pasa-bajos es aquel cuyas frecuencias de corte ω se especifican en fracciones de π, es decir en fracciones de la mitad de la frecuencia de muestreo. Por ejemplo si deseamos diseñar un filtro digital pasa-bajos cuya frecuencia de corte de la banda de paso sea 1000 Hz y la frecuencia de muestreo sea 5000 hz, debemos primero determinar el valor de la frecuencia de corte del filtro normalizado correspondiente.

$$\omega = 2\pi . f/fm$$
$$\omega = 2\pi . 1000/5000$$
$$\omega = 0.4\ \pi \quad [rad]$$

3.1 TRANSFORMACION ANALOGICA - ANALOGICA

Básicamente este método de transformación consiste en realizar el cambio de banda antes de la digitalización del filtro.

Entonces para diseñar un filtro digital utilizando esta transformación se debe seguir los siguientes pasos:

a) Diseñar el filtro analógico pasa-bajos de orden N
b) Efectuar la transformación de banda analógico-analógico
c) Digitalizar el filtro analógico

La función de transferencia del filtro analógico antes de ser transformado tiene como variable compleja a s, la transformación consiste en encontrar una función de transferencia cuya variable compleja sea $\bar{s}$, y cumpla con los requerimientos del filtro deseado. Para esto se necesita reemplazar s en el filtro no transformado por una función de $\bar{s}$.

Esta función de $\bar{s}$ es específica para cada tipo de transformación de banda, y depende además de las frecuencias de corte deseadas.

3.1.1 TRASFORMACION PASA-BAJOS A PASA-BAJOS

Para realizar esta transformación bastará multiplicar por una constante que solo cambie los valores de las frecuencias de corte, porque la forma de la respuesta de frecuencia es la misma. Así considerando $s = j\Omega$:

$s/j = \Omega a$ [rad/seg] frecuencia de corte antes de la transformación

$\bar{s}/j = \Omega u$ [rad/seg] frecuencia de corte deseada

dividiendo estas igualdades tenemos:

$$s / \bar{s} = \Omega a / \Omega u$$

si reemplazamos u por $\Omega u/\Omega a$, entonces:

$$s = \bar{s} / u \qquad (3.1)$$

este último resultado viene a ser la función de transformación que se deberá reemplazar en la función de transferencia del filtro no transformado.

Si el filtro no transformado es el filtro analógico normalizado, entonces:

$$\Omega a = 1 \text{ [rad/seg]}$$

y:

$$u = \Omega u$$

esto quiere decir que la constante u es igual a la frecuencia de corte del filtro deseado siempre que obtengamos el filtro transformado a partir de un filtro normalizado.

En la figura 3.1 se muestra gráficamente este tipo de transformación. Todos los gráficos que se indiquen en esta sección corresponderán a los de Butterworth.

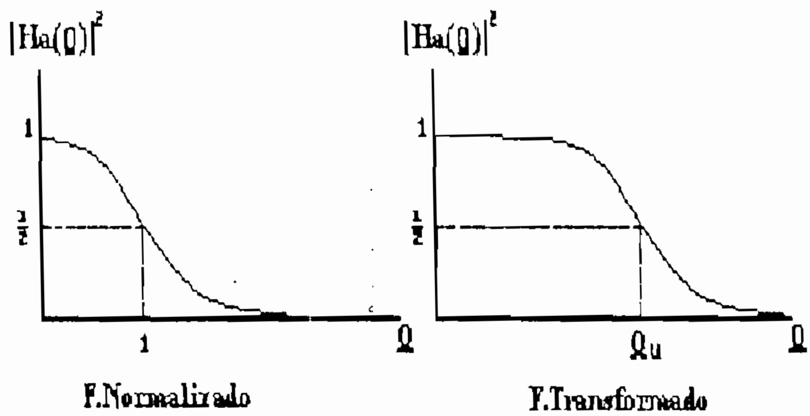


Figura 3.1: Transformación de un Filtro Pasa-bajos

Se debe tomar en cuenta que, la constante u, que se utiliza en la transformación, escala a todas las frecuencias de la misma manera.

3.1.2 TRASFORMACION PASA-BAJOS A PASA-ALTOS

En esta transformación de banda la forma de la respuesta de frecuencia es diferente. En la figura 3.2 vemos ilustrada la respuesta de transferencia antes y después de la transformación.

Esta transformación se logrará intercambiando la variable s por la variable $1/\bar{s}$, y se incluirá también la constante de escalamiento u para alcanzar el valor de las frecuencias de corte deseadas. Entonces la fórmula de transformación será:

$$s = u / \bar{s} \qquad (3.2)$$

De esta manera la constante u es igual a la frecuencia de corte deseada, si el diseño se hace a partir de un filtro normalizado.

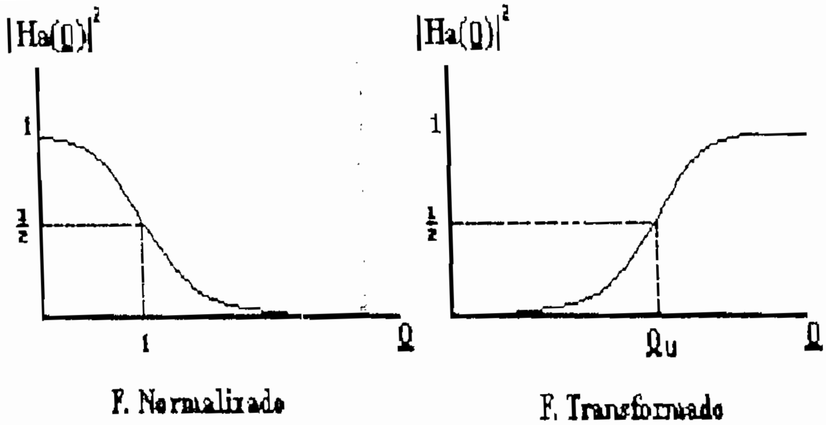


Figura 3.2: Transformación a un Filtro Pasa-altos

3.1.3 TRASFORMACION PASA-BAJOS A PASA-BANDA

La fórmula de transformación utilizada aquí se ha obtenido a partir de las dos fórmulas anteriores. Considerando que la respuesta de frecuencia de un filtro pasa-banda sería como tener la respuesta de frecuencia de un filtro pasa-altos seguida de la respuesta de un filtro pasa-bajos. Sea Ω_l la frecuencia de corte inferior, esta corresponderá a la respuesta de un filtro pasa-altos; y sea Ωu la frecuencia de corte superior que corresponderá a la respuesta de un filtro pasa-bajos.

Entonces tomando las fórmulas de transformación anteriores y reemplazando u por $\Omega u / \Omega a$, y l por $\Omega_l/\Omega a$, tendremos:

$$s = \bar{s} / u \qquad y \qquad s = l / \bar{s}$$

Restando estas relaciones:

$$0 = \bar{s} / u - l / \bar{s}$$

entonces:

$$\bar{s}^2 = u.l \tag{3.3}$$

Considerando:

$$s/j = 1 \text{ [rad/seg]} \quad \text{frecuencia de corte normalizada}$$

$$s = j$$

$$s^2 = -1$$

$$1 = -s^2$$

Y luego se suma 1 a cada miembro:

$$2 = 1 - s^2$$

$$2/s = (1 - s^2)/s$$

$$2/s = 1/s - s$$

reemplazando 3.1 y 3.2 :

$$\frac{2}{s} = \frac{\bar{s}}{l} - \frac{\bar{s}}{u}$$

$$\frac{2}{s} = \frac{\bar{s}.u - \bar{s}.l}{u.l}$$

despejando s y reemplazando 3.3

$$s = \frac{2u.l}{\bar{s}(u-l)}$$

$$s = \frac{\bar{s}^2 + u.l}{\bar{s}(u-l)} \tag{3.4}$$

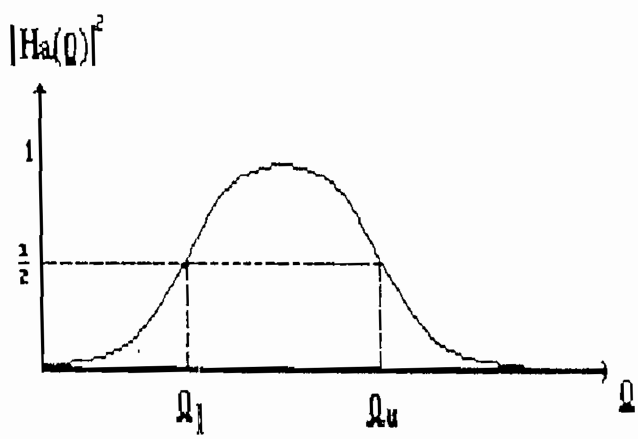


Figura 3.3: Respuesta de Frecuencia del Filtro Pasa-banda

En la figura 3.3 podemos ver la forma que adopta el filtro transformado pasa-banda.

3.1.4 TRASFORMACION PASA-BAJOS A ELIMINA-BANDA

La forma de la respuesta de frecuencia de este tipo de filtro resulta ser lo contrario de lo que representa un filtro pasa-banda. Entonces basta cambiar el eje Ω por $1/\Omega$, por lo tanto la fórmula de transformación puede ser análogamente determinada, invirtiendo la expresión 3.4. Entonces:

$$s = \frac{\bar{s}(u - l)}{\bar{s}^2 + u.l}$$

La figura 3.4 representa la respuesta de frecuencia del filtro transformado elimina-banda. Podemos verificar que esta respuesta podría formarse de la respuesta de un filtro pasa-bajos seguida de un filtro pasa-altos, exactamente lo contrario de lo que se dijo para el filtro pasa-banda.

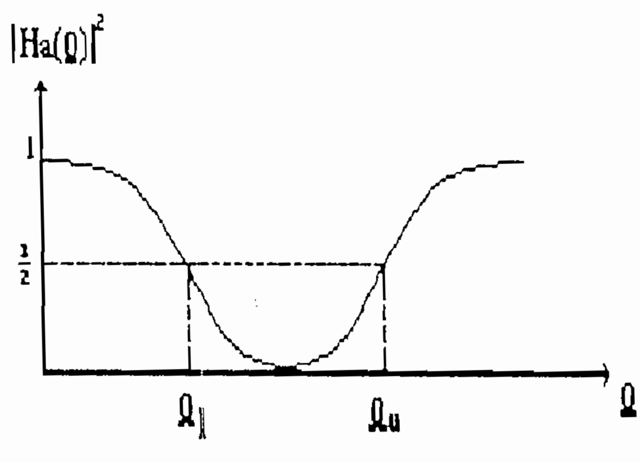


Figura 3.4: Respuesta de frecuencia del Filtro Elimina-banda

Estas transformaciones pueden ser usadas para trasladar la banda de paso de cualquier filtro analógico sea este de Butterworth, Chevishev tipo I ó tipo II, ó Elíptico.

En la práctica dadas las especificaciones de un filtro debemos determinar el valor de las contantes u y l, y en base a estas, determinar las frecuencias de corte del filtro pasa-bajos no transformado necesario para cumplir con el filtro deseado.

3.2 TRANSFORMACION DIGITAL - DIGITAL

Básicamente este método de transformación consiste en realizar el cambio de banda después de la digitalización del filtro.

Para diseñar un filtro digital utilizando esta técnica se debe seguir los siguientes pasos:

a) Diseñar el filtro analógico pasa-bajos de orden N
b) Digitalizar el filtro analógico pasa-bajos
c) Efectuar la transformación de banda digital-digital

La función de transferencia del filtro digital antes de ser transformado asume a z^{-1} como variable, la transformación consiste en encontrar una función de transferencia con variable $\check{z}^{-1}$, que cumpla con los requerimientos del filtro deseado. Para esto se necesita reemplazar z^{-1} en el filtro no transformado por una función de $\check{z}^{-1}$.

$$z^{-1} = G(\check{z}^{-1})$$

Las características que debe cumplir la función G de transformación son:

a) Ser una función racional
b) Ser tal que los filtros estables y causales en el plano z se transformen en filtros estables y causales en el plano $\check{z}$.
c) El círculo unitario en el plano z se debe transformar en el círculo unitario en el plano $\check{z}$.

En base a estas características se establece que:

$$G(\check{Z}^{-1}) = \pm\prod_{k=1}^{n} \frac{\check{Z}^{-1}-\alpha_k}{1-\alpha_k\check{Z}^{-1}} \qquad (3.5)$$

La contante α_k debe ser menor que 1 para garantizar la estabilidad del filtro. Esta constante depende de la frecuencia de corte del filtro antes y después de la transformación. Asumiremos que θ es la frecuencia de corte digital del filtro pasa-bajos no transformado y que ω es la frecuencia del filtro digital transformado.

3.2.1 TRASFORMACION PASA-BAJOS A PASA-BAJOS

Esta transformación se realiza, tomando la expresión 3.5 con signo positivo y n=1. Entonces:

$$z^{-1} = G(\check{Z}^{-1}) = \frac{\check{Z}^{-1}-\alpha}{1-\alpha\check{Z}^{-1}}$$

Sustituyendo $z = e^{j\theta}$ y $\check{Z} = e^{j\omega}$, tenemos:

$$e^{-j\theta} = \frac{e^{-j\omega}-\alpha}{1-\alpha e^{-j\omega}} \qquad (3.6)$$

Si despejamos ω, obtendremos una expresión que dependerá de la frecuencia de corte del filtro no transformado θ y del valor de la constante α.

$$\omega = \arctan\left(\frac{(1-\alpha^2)\,sen\theta}{2\alpha+(1+\alpha^2)\cos\theta}\right)$$

Haciendo variar a θ con valores de 0 a π, y dependiendo del valor de α, podemos tener algunas curvas de ω.

En la figura 3.5 se indica que cuando $\alpha=0$ no existe transformación de frecuencia, es decir que $\theta=\omega$. Pero para $-1<\alpha<0$ y $0<\alpha<1$ podemos obtener transformaciones de frecuencia en aumento o disminución de la frecuencia de corte respectivamente, respecto de la frecuencia de corte del filtro no transformado θ.

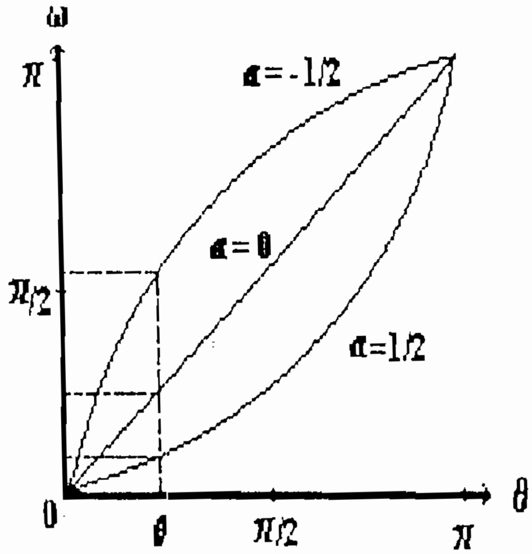


Figura 3.5: ω en función de θ para algunos valores de α

Retomando la expresión 3.6 despejaremos α.

$$e^{-j\theta} - \alpha e^{-j(\theta+\omega)} = e^{-j\omega} - \alpha$$

$$\alpha = \frac{e^{-j\omega} - e^{-j\theta}}{1 - e^{-j(\theta+\omega)}}$$

$$\alpha = \frac{sen\left(\frac{\theta-\omega}{2}\right)}{sen\left(\frac{\theta+\omega}{2}\right)} \tag{3.7}$$

Con esta expresión podemos determinar el valor de α que intervendrá en la función $G(z^{-1})$, a partir de las frecuencias de corte del filtro antes y después de transformarlo.

3.2.2 TRASFORMACION PASA-BAJOS A PASA-ALTOS

Consideraremos ahora que ω es la frecuencia de corte del filtro pasa-altos deseado (Vea la figura 3.6). Para lograr esta transformación basta con tomar la función $G(z^{-1})$ con signo negativo y n=1, entonces:

$$G(z^{-1}) = -\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$$

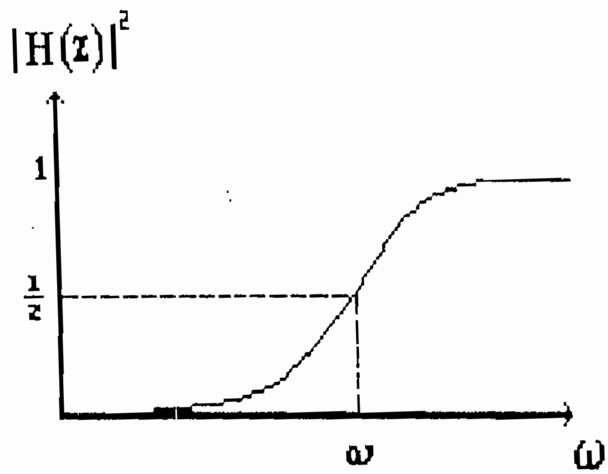


Figura 3.6: Filtro transformado pasa-altos

Realizando el mismo procedimiento que en el caso anterior, tenemos que el valor de α queda determinado por:

$$\alpha = \frac{\cos\left(\frac{\theta-\omega}{2}\right)}{\cos\left(\frac{\theta+\omega}{2}\right)} \qquad (3.8)$$

Para realizar la transformación obtenemos la función $G(z^{-1})$ con el valor de α dado por 3.8, para luego reemplazar z^{-1} por $G(z^{-1})$ en la función de transferencia del filtro no transformado.

Por lo general el coseno de la resta es mayor que el coseno de la suma de dos ángulos, por lo tanto el valor de α será mayor que 1, pudiendo provocar inestabilidad en el filtro.

Curiosamente algunos autores suelen reemplazar en la función $G(\check{z}^{-1})$, el inverso de la expresión 3.8, para evitar la inestabilidad.

3.2.3 TRASFORMACION PASA-BAJOS A PASA-BANDA

La respuesta de frecuencia de un filtro pasa-banda se especifica con dos frecuencias de corte. Llamaremos ω_1 y ω_2 respectivamente a las frecuencias de corte inferior y superior. Ver la figura 3.7.

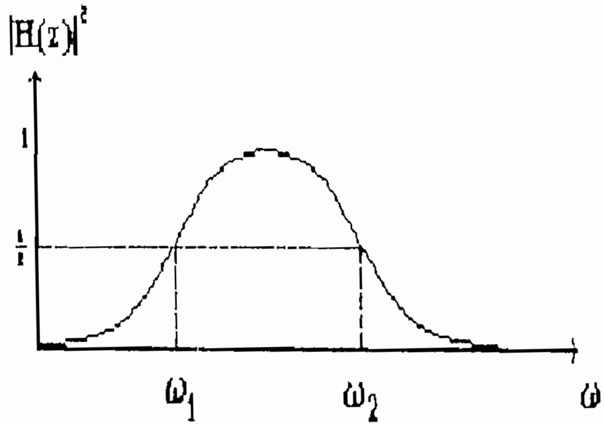


Figura 3.7: Filtro transformado pasa-banda

Para determinar la función de transformación $G(\check{z}^{-1})$ utilizaremos la expresión 3.5 con signo negativo y n=2.

$$G(\check{z}^{-1}) = -\frac{\check{z}^{-1}-\alpha_1}{1-\alpha_1\check{z}^{-1}}\cdot\frac{\check{z}^{-1}-\alpha_2}{1-\alpha_2\check{z}^{-1}}$$

$$G(\check{z}^{-1}) = -\frac{\check{z}^{-2}-(\alpha_1+\alpha_2)\check{z}^{-1}+\alpha_1\alpha_2}{1-(\alpha_1+\alpha_2)\check{z}^{-1}+\alpha_1\alpha_2\check{z}^{-2}}$$

Sustituyendo las siguientes equivalencias:

$$\alpha_1+\alpha_2 = \frac{2\alpha k}{k+1}$$

$$\alpha_1\alpha_2 = \frac{k-1}{k+1}$$

donde α y k son constantes a ser determinadas. Entonces nos queda:

$$G(\check{Z}^{-1}) = -\frac{\check{Z}^{-2}-\left(\frac{2\alpha k}{k+1}\right)\check{Z}^{-1}+\frac{k-1}{k+1}}{1-\left(\frac{2\alpha k}{k+1}\right)\check{Z}^{-1}+\frac{k-1}{k+1}\check{Z}^{-2}} \qquad (3.9)$$

Para determinar las constantes α y k en función a las frecuencias seguiremos un proceso igual al empleado anteriormente, considerando que ahora tenemos dos frecuencias de corte. Los resultados obtenidos son:

$$\alpha = \frac{\cos\left(\frac{\omega_2+\omega_1}{2}\right)}{\cos\left(\frac{\omega_2-\omega_1}{2}\right)}$$

$$k = \cot\left(\frac{\omega_2-\omega_1}{2}\right)\tan\left(\frac{\theta}{2}\right)$$

Con esto tenemos la función de transformación 3.9 y podemos reemplazarla en la función de transferencia del filtro no transformado.

3.2.4 TRASFORMACION PASA-BAJOS A ELIMINA-BANDA

Esta transformación se comporta de alguna manera en una forma dual con respecto a la fórmula anterior. De hecho es por que el filtro elimina-banda tiene un efecto opuesto al filtro pasa-banda. Observemos la figura 3.8.

La función de transformación proviene de la expresión 3.5 con n=2 y signo positivo, así:

$$G(\check{Z}^{-1}) = \frac{\check{Z}^{-1}-\alpha_1}{1-\alpha_1\check{Z}^{-1}}\cdot\frac{\check{Z}^{-1}-\alpha_2}{1-\alpha_2\check{Z}^{-1}}$$

$$G(\check{Z}^{-1}) = \frac{\check{Z}^{-2}-(\alpha_1+\alpha_2)\,\check{Z}^{-1}+\alpha_1\alpha_2}{1-(\alpha_1+\alpha_2)\,\check{Z}^{-1}+\alpha_1\alpha_2\check{Z}^{-2}}$$

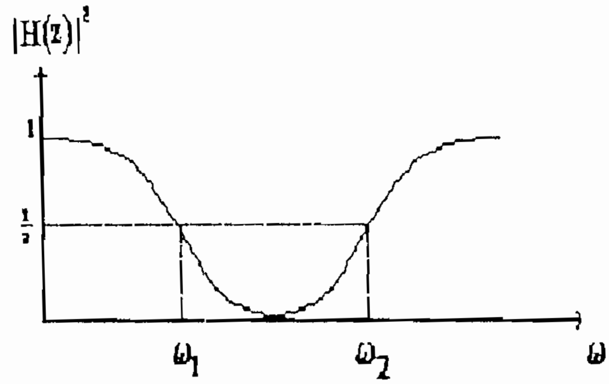


Figura 3.8: Filtro transformado elimina-banda

sustituyendo las siguientes equivalencias:

$$\alpha_1+\alpha_2 = \frac{2\alpha}{k+1}$$

$$\alpha_1\alpha_2 = \frac{1-k}{1+k}$$

donde α y k son constantes a ser determinadas. Entonces nos queda:

$$G(\check{Z}^{-1}) = \frac{\check{Z}^{-2}-\left(\frac{2\alpha}{1+k}\right)\check{Z}^{-1}+\frac{1-k}{1+k}}{1-\left(\frac{2\alpha}{1+k}\right)\check{Z}^{-1}+\frac{1-k}{1+k}\check{Z}^{-2}} \tag{3.10}$$

Las contantes α y k en función a las frecuencias quedan determinadas así:

$$\alpha = \frac{\cos\left(\frac{\omega_2+\omega_1}{2}\right)}{\cos\left(\frac{\omega_2-\omega_1}{2}\right)}$$

$$k = \tan\left(\frac{\omega_2-\omega_1}{2}\right)\tan\left(\frac{\theta}{2}\right)$$

Con esto tenemos la función de transformación 3.10 y podemos reemplazarla en la función de transferencia del filtro no transformado.

En la práctica, para poder cumplir con las frecuencias de corte deseadas de un filtro se debe determinar correctamente la función de transformación, y con esta, se debe encontrar las frecuencias de corte del filtro pasa-bajos no transformado de tal manera que al transformarlo cumpla con las frecuencias de corte requeridas.

a) Los filtros pasa-bajos y pasa-altos, tienen dos frecuencias de corte que los caracterizan. Estas frecuencias son: la frecuencia de paso (fp) y la de supresión (fs). Para el filtro transformado estas frecuencias son conocidas. Se asume una frecuencia de paso para el filtro no transformado y con la frecuencia (fp) del filtro transformado determinamos la función de transformación de banda utilizando los procedimientos descritos en este capítulo. Luego con la frecuencia (fs) del filtro transformado y la función de transformación encontramos la frecuencia de supresión para el filtro no transformado. Esto se basa en que cada transformación tiene una sola función específica que transformará a todas las frecuencias del filtro no transformado, por lo tanto conociendo la función de transformación y las frecuencias que deseamos obtener, podemos determinar que filtro pasa-bajos no transformado nos conviene diseñar para obtener el filtro deseado.

b) Los filtro pasa-banda y elimina-banda, se especifican con cuatro frecuencias, es decir dos por cada banda de transición. Las funciones aquí descritas para la transformación de banda tanto en el método analógico-analógico como en el digital-digital, se han determinado basándose en que cada frecuencia del filtro no transformado corresponda a dos frecuencias del filtro transformado dispuestas simétricamente.

Supongamos que un filtro pasa-banda transformado tiene a (fls) y (flp) como frecuencias de supresión y de paso respectivamente, de la banda de transición inferior; y tiene a (fup) y (fus) como frecuencias de paso y de supresión respectivamente, de la banda de transición superior; y necesitamos saber cual es el filtro pasa-bajos no transformado que nos proporcionará el filtro transformado deseado. Entonces asumimos una frecuencia de paso para el filtro no transformado y con (flp) y (fup) determinamos la función de transformación. Para encontrar el valor de la frecuencia de supresión del filtro no transformado no hace falta conocer las dos frecuencias (fls) y (fus), nos basta conocer una de ellas y la otra ya sabemos que es simétrica. Se determina la posición del eje de simetría con el promedio de las dos frecuencias simétricas conocidas, así: $f_{eje\ de\ simetria} = [fup+flp] / 2$

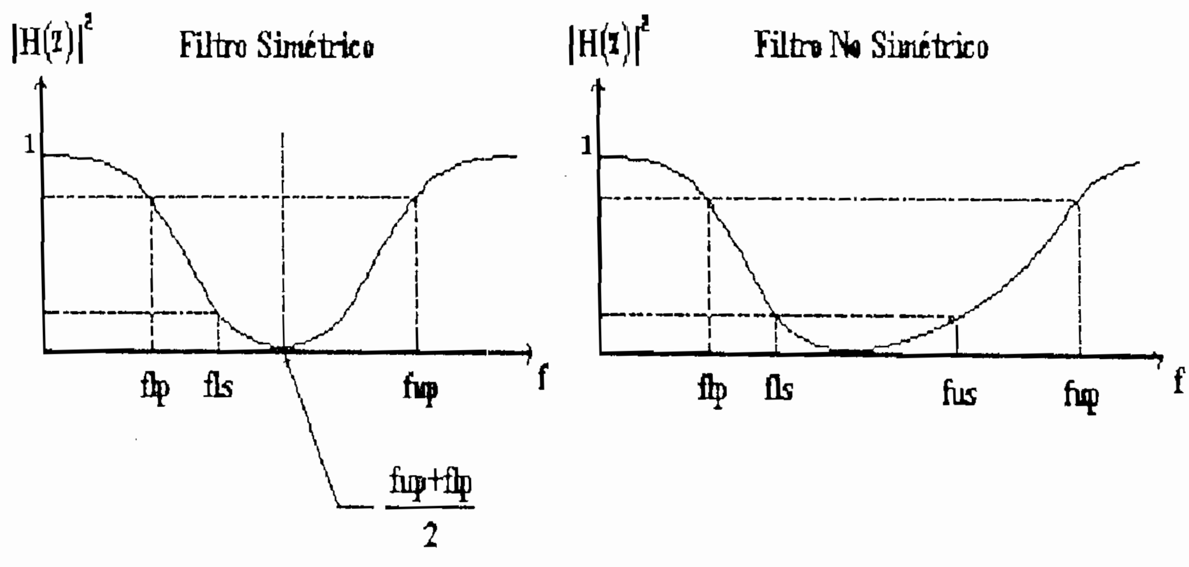


Figura 3.9: Respuestas de Frecuencia de filtros elimina-banda.

Este resultado nos lleva a concluir que solo se puede diseñar filtros pasa-banda y elimina-banda simétricos (caracterizados por tres frecuencias), y que para el diseño de filtros no simétricos (caracterizados por las cuatro frecuencias) se debe encontrar un filtro simétrico que se aproxime realizando ajustes. Ver la figura 3.9.

REFERENCIA:

[3.1] A.V.OPPENHEIM AND R.W.SCHAFER, Digital Signal Processing
Prentice-Hall, 1975.

CAPITULO 4

SOFTWARE DE LA SIMULACION DEL FILTRO

4.1 DESCRIPCION DEL PROGRAMA A DESARROLLAR

Con lo anteriormente estudiado procedemos al desarrollo de un software que simulará el filtraje de una señal previamente almacenada.

Nuestro análisis previo a la obtención de un filtro consiste en diseñar un filtro analógico, realizar su conversión a digital y transformarlo a la banda deseada de filtrado.

Se conocen cuatro métodos de diseño de filtros analógicos, tres métodos de conversión en digital y dos métodos de transformación de banda, el programa a desarrollar permite utilizar cualquiera de estos métodos, hasta obtener el filtro deseado. Estos métodos serán escogidos según la necesidad que se tenga para realizar el filtrado de una señal.

Debemos considerar que, la simulación de la transformación de banda analógica-analógica se realizará únicamente para el método de conversión digital "Transformada Bilineal", y con carácter ilustrativo, ya que la transformación digital-digital es mas sencilla y trabaja con las mismas ventajas.

Previamente el programa iniciará pidiendo datos, tales como la selección de los métodos mencionados anteriormente y los requerimientos de la respuesta de frecuencia del filtro que se quiera obtener. Entre los requerimientos es necesario especificar los valores de las atenuaciones de las bandas de paso y de las bandas de supresión, las frecuencias de corte de paso y supresión de las bandas de transición que hubieren, y, la frecuencia de muestreo de la señal para efectuar la normalización del filtro en el dominio digital.
Obviamente que para poder definir la respuesta de frecuencia

del filtro se deberá saber que el filtro es pasa-bajos, pasa-altos, pasa-banda ó elimina-banda.

Luego de conocer los parámetros iniciales, se efectúa todo el cálculo matemático según lo estudiado, hasta obtener la señal filtrada.

Para garantizar la corrección del diseño se deberá determinar las respuestas de frecuencia analógica y digital, para que una vez graficadas puedan ser analizadas.

Un punto importante es el manejo de los archivos para almacenar la señal antes y después de filtrarla. También podremos archivar las respuestas de frecuencia, con el único propósito de efectuar un análisis comparativo.

Los resultados que puedan graficarse se los debe hacer adecuadamente, de tal forma que se facilite el análisis de los mismos. Podríamos graficar las respuestas de frecuencia analógica, digital e incluso la respuesta de frecuencia del filtro analógico luego de ser transformado con el método de transformación analógico-analógico. También la señal antes y después de filtrarla, la cual si es muy larga se deberá visualizar por tramos.

Para facilitar el manejo del programa se provee de un sistema de menús, que evitará sobre todo el ingreso de datos erróneos.

El programa a desarrollar consta de dos partes: Programa Principal y Subrutinas. Debido al gran número de subrutinas, estas se encuentran repartidas en tres módulos. Estas partes serán explicadas con mayor detenimiento en los siguientes numerales.

4.2 DESARROLLO DEL PROGRAMA PRINCIPAL

El programa principal se encuentra localizado en el módulo principal llamado "FDIGITAL.BAS".

Básicamente todo el manejo del programa se efectúa con la utilización de un menú principal. De esta manera el programa principal, dependiendo de la selección que se haya realizado, llama a la subrutina adecuada que realizará la tarea respectiva. Esto lo realiza mediante un SELECT CASE de la variable men. Esta variable es proporcionada por la subrutina "menu", que tienen la tarea de seleccionar.

En este módulo además, se definen las variables, las constantes, y se declaran todas las subrutinas. También se define el tipo de pantalla que se utilizará, con SCREEN 12, especificamos un tipo de pantalla VGA que proporciona una pantalla de texto de 30 líneas y 80 columnas, y una pantalla de gráficos de 480 x 640 pixels (puntos) los cuales pueden adoptar hasta 16 colores diferentes. Esta pantalla nos conviene sobretodo por el manejo de pixel de varios colores, puesto que algunos gráficos para el análisis comparativo han sido dibujados en un mismo recuadro, y podrán distinguirse gracias al uso de colores.

Una parte importante que se tiene presente en el programa principal es la declaración de las teclas <F2> y <F10>.
La tecla <F2> se utiliza para conocer las características del filtro a diseñarse y la tecla <F10> para llamar al menú principal. El uso de <F10> resulta altamente conveniente, puesto que muchas veces uno se arrepiente de haber activado alguna opción, y para volver al menú debería esperar que se cumpla la tarea.

Para evitar que el programa se detenga cuando se intente abrir un archivo que no existe o cuando se realice un cálculo indeterminado, hemos activado la sentencia ON ERROR. Cada vez que se detecta un error el programa se dirige a la etiqueta "archivoerror", y ejecuta instrucciones que indican al usuario que se ha registrado un error.
Existe una subrutina que inicializa los parámetros y variables del programa. Esta subrutina es llamada al inicio del programa principal, y tiene el nombre de "valoresiniciales".

4.3 DESARROLLO DE SUBRUTINAS

En el programa se ha dado una vital importancia al uso de las subrutinas, ganando con esto una estructuración adecuada del sistema. El programa principal, como se describió anteriormente, llama a cada subrutina a realizar una tarea específica. El programa tiene 36 subrutinas. Las mismas que han sido nombradas con títulos específicos que ayuden a la implementación del programa.

Entre las subrutinas podemos distinguir tres grupos de acuerdo a las funciones que desempeñan. Estas son:

4.3.1 SUBRUTINAS DE INGRESO DE DATOS

Las subrutinas que especifican el filtro a diseñarse son:

- TIPO
- DATOS
- TIPOANALOGICO
- TIPOCONVERSION
- TRANSFORMACION

4.3.1.1 <u>Subrutina tipo</u>

Esta subrutina tiene como función la de dar a conocer si el filtro a diseñarse es pasa-bajos, pasa-altos, pasa-banda o elimina-banda. Para facilitar la selección, se preparará una pantalla dividida en cuatro partes, donde en cada parte se dibuja la respuesta de frecuencia de uno de los tipos de banda a escogerse.

La selección se la hace con las flechas, distinguiéndose la banda escogida por tener invertidos los colores del dibujo.
La información extraída se la llevará en la variable <u>tip</u>.

Mientras se escoge la banda, estaremos en un lazo (DO...WHILE), y saldremos de este al presionar <ENTER>.

4.3.1.2 Subrutina datos

Esta subrutina sirve para ingresar las especificaciones de la respuesta de frecuencia del filtro elegido. Estos datos incluyen las frecuencias de corte, de paso y de supresión, tomadas en Hz, las atenuaciones de cada banda en dB y la frecuencia de muestreo dada también en Hz.

Para que el ingreso de datos resulte fácil, se dibujará en la pantalla la respuesta de frecuencia seleccionada, sobre la cual se indican las especificaciones a ser tomadas. Para dibujar la respuesta de frecuencia seleccionada utilizamos un SELECT CASE de la variable tip.

Si es un filtro pasa-bajos o pasa-altos, se realiza el siguiente ciclo de trabajo:

a. Se ubica el cursor para tomar la frecuencia de muestreo.
b. Se coloca el cursor adecuadamente para tomar la atenuación de la banda de paso.
c. Se coloca el cursor adecuadamente para tomar la atenuación de la banda de supresión.
d. Se ubica el cursor en la frecuencia de corte inferior. Esta frecuencia corresponde a la de paso en el filtro pasa-bajos y a la de supresión en el filtro pasa-altos.
e. Se ubica el cursor en la frecuencia de corte superior. Esta frecuencia corresponde a la de supresión en el filtro pasa-bajos y a la de paso en el filtro pasa-altos.

Si el filtro es pasa-banda o elimina-banda, el ciclo de trabajo es parecido al anterior, con la diferencia de que, se aumenta el ingreso de una frecuencia de corte más, que especifica a la frecuencia de paso de la segunda banda de transición.

Para aceptar un dato primero se lo analiza. El análisis del dato consiste en revisar que las frecuencias superiores sean mayores que las inferiores o viceversa, que la máxima frecuencia especificada no sobrepase a mitad de la frecuencia de muestreo, y que la atenuación de la banda de paso sea mayor a la atenuación de la banda de supresión o viceversa según los casos. En los filtros elimina-banda se prevee que las frecuencias de supresión se superpongan, condicionando a la frecuencia de paso de la segunda banda de transición así:

$$fup > f_{\text{eje de simetría}} + (flp - fls)$$

El ciclo de trabajo esta dentro de un lazo infinito, y se mueve de un punto a otro al presionar la tecla <ENTER>. Para salir del lazo se utiliza la tecla <F10>, que nos llevará al menú.

4.3.1.3 Subrutina tipoanalogico

Se encarga de recibir la especificación del tipo de filtro analógico en base al cual se realizará el diseño del filtro digital.

Utilizando el proceso de sub-menús, esta subrutina utiliza un SELECT CASE de la variable tipana, que indica cual ha sido la selección. Así cualquiera que sea el valor de tipana, invierte el color del nombre del tipo analógico escogido. Cada cambio de selección se hace aumentando o disminuyendo el valor de la variable tipana, luego que se detecte la tecla de <FLECHA ARRIBA> o <FLECHA ABAJO>, respectivamente.

Esto está dentro de un lazo DO... WHILE, que al detectar la tecla <ENTER>, sale del lazo y termina la subrutina.

Lo descrito en esta subrutina se repite para todas las subrutinas que presentan sub-menús.

4.3.1.4 <u>Subrutina tipoconversion</u>

Se desarrollará esta subrutina para especificar el tipo de conversión de filtro analógico en digital, que se desea utilizar. Utiliza el método de sub-menús para realizar la selección, y lleva la información en la variable <u>tipconv</u>.

4.3.1.5 <u>Subrutina transformacion</u>

Esta subrutina realiza el escogitamiento del tipo de transformación de banda que se desea emplear.

Con la instrucción IF...THEN se condiciona el uso de esta subrutina, para que se pueda usar solo cuando se utilice la transformada bilineal.

Esta subrutina también utiliza el método de sub-menús. La información extraída en esta subrutina se guardará en la variable <u>trans</u>.

Para el ingreso de otros datos adicionales, tenemos las siguientes subrutinas:

- ARCHIVO
- VALORESINICIALES

4.3.1.6 <u>Subrutina archivo</u>

Esta subrutina ha sido dedicada al manejo de archivos.

Se presenta una pantalla donde se puede ingresar el nombre de los archivos que contienen a:

- La señal a filtrarse
- La señal filtrada
- La respuesta de frecuencia analógica
- La respuesta de frecuencia digital

Al terminar de especificar los archivos, se procede a:

1. Si se ha cambiado el archivo de la señal a filtrarse, se inspecciona el archivo. Esto consiste en determinar el número de datos guardados y la máxima amplitud que tienen. La subrutina "inspeccion" se encarga de hacerlo. Luego se filtra la señal, calculando y(n). Si el archivo de la señal a filtrarse tiene la terminación *.wav, el programa cambia este formato al formato que entiende el programa. Luego realiza la inspección y el filtrado. Para finalizar, la señal filtrada es cambiada al formato inicial *.wav.

2. Si se ha cambiado el archivo de la señal filtrada, se efectúa de nuevo el filtrado y se va guardando el resultado en este archivo.

3. Si se ingresa el nombre del archivo en el que se guardará la respuesta de frecuencia analógica del filtro, se guarda esta respuesta tomando los datos del arreglo, ha(i), si la transformación de banda es digital-digital y, hat(i), si la transformación de banda es analógica-analógica.

4. Esta tarea es parecida a la anterior, con la diferencia de que se guarda la respuesta de frecuencia digital. Los datos a guardar se toman del arreglo hi(i) cuando se usa la conversión de invariancia de impulso, hb(i) si se usa la transformada bilineal, y hp(i) cuando la conversión utilizada es la invariancia de pulso.

Siempre, antes de filtrar una señal se llama a esta subrutina.

4.3.1.7 Subrutina valoresiniciales

Para que el programa pueda empezar a funcionar se debe inicializar algunos parámetros y variables. Esta subrutina es llamada dentro del programa principal, y deja, definidas las características y las especificaciones de la respuesta de frecuencias de un filtro cualquiera tomado al azar. También especifica un archivo para la señal de entrada, para la salida, para la respuesta de frecuencia analógica y para la respuesta de frecuencia digital.

De esta manera se prevé que por imprudencia se desee simular un filtro sin antes haber señalado las características de este.

Existen otras subrutinas que ingresan datos pero propiamente del programa, es decir, para indicarle ciertas sentencias o tareas que deseamos que realice. Estas son:

- GRAFICOS
- MENU

4.3.1.8 Subrutina graficos

Esta subrutina se utilizará para realizar las siguientes tareas:

1. Graficar en toda la pantalla y con cuadrículas, dos respuestas de frecuencia que sean de interés para un análisis comparativo.
2. Indicar cuales son las respuestas de frecuencia que se desean ver, llamándolas por el nombre del archivo en el cual se guardaron.
3. Obtener un gráfico, tanto de las respuestas de frecuencia como de la señal antes y después de filtrarla.

Las tareas son elegidas mediante el método de sub-menús, asignando a la variable comp el valor correspondiente.

4.3.1.9 Subrutina menu

De esta subrutina habíamos hablado en el desarrollo del programa principal. Consiste en proporcionar a la variable men la indicación de la tarea que se quiere ejecutar. Para hacerlo utiliza el método de sub-menús. Este sub-menú es algo diferente a los demás, porque a más de seleccionar las opciones con las <FLECHAS>, podemos seleccionar presionando las letras resaltadas.

4.3.2 SUBRUTINAS QUE REALIZAN CALCULOS

Para efectuar la simulación de un filtro digital, se han de realizar una serie de procesos de cálculos matemáticos, los cuales están destinados a hacerse por medio de subrutinas específicas. Estas subrutinas están organizadas por otra subrutina llamada "simulacion".

4.3.2.1 Subrutina simulacion

Esta, se encarga de llamar adecuadamente a cada una de las subrutinas a efectuar los cálculos necesarios para obtener el filtro.

Primero llama a las subrutinas de cálculo previo a la obtención del filtro analógico. Luego se encarga de determinar la respuesta de frecuencia del filtro analógico.

Posteriormente dependiendo del tipo de conversión del filtro a digital, llama a un grupo de subrutinas para obtener la función de transferencia y la respuesta de frecuencia del filtro digital.

Finalmente llama a la subrutina "archivo" que invita a verificar la localización de archivos y luego efectúa el filtraje de la señal utilizando los coeficientes de la función de transferencia.

Las subrutinas que efectúan los cálculos previos para el diseño del filtro analógico, son:

- WAPS
- ORDEN
- POLOS

4.3.2.2 Subrutina waps

Esta subrutina determina los valores de las frecuencias de corte del filtro analógico a diseñarse y los valores de las constantes que intervienen en las funciones de transformación de banda.

El procedimiento a seguir, para cumplir con las especificaciones de frecuencia del filtro digital deseado, consiste, en determinar cual es el filtro analógico adecuado que al efectuar la transformación de banda nos de como resultado el filtro pedido.

Para esto, tomamos una de las frecuencias de corte del filtro digital y asumiendo una frecuencia de corte para el filtro no transformado, encontramos una constante y formamos una función de transformación. Ahora con la función de transformación obtenida y las frecuencias de corte del filtro digital especificado, determinaremos el resto de frecuencias del filtro no transformado.

Para realizar esta tarea, primero debemos identificar si la transformación de banda es digital-digital o analógica-analógica. Luego es necesario distinguir si los filtros analógicos son Elípticos, pues estos tienen un trato especial.

Entonces tenemos distintos casos, así:

CASO 1 : Filtros No Elípticos con Transformación Analógica-Analógica.

En este caso se deberá tomar en cuenta el prealabeo de las frecuencias, considerando que la transformación analógica-analógica se utilizó solo con la transformada bilineal.
Asumimos un valor para la frecuencia de paso del filtro no transformado, wap:

$$wap = 1 \quad [rad]$$

Esto asumiremos en todos los casos, excepto en los filtros pasa-bajos y en los filtros Elípticos.

a) Para filtros pasa-bajos.

No se realiza la transformación de banda. Por lo tanto la asignación de las variables wap y was se las realiza directamente de la normalización de las frecuencias del filtro digital. Así:

$$wap = 2\ fl\ \pi\ /\ fm$$
$$was = 2\ fu\ \pi\ /\ fm$$

donde:
- fl es la frecuencia de corte de la banda de paso en Hz.
- fu es la frecuencia de corte de la banda de supresión en Hz.
- fm es la frecuencia de muestreo de la señal digital en Hz.

Estas expresiones al momento no incluyen todavía el prealabeo de frecuencia.

b) Para filtros pasa-altos.

Sabemos que están definidos por dos frecuencias de corte, la frecuencia de supresión <u>fl</u> y la de paso <u>fu</u>.

Prealabeamos la frecuencia <u>wap</u> asumida anteriormente, (wap=1), así:

$$w = 2\ \mathrm{TAN}(wap/2)$$

Ahora tomamos el valor de la frecuencia de paso especificada, la normalizamos y la prealabeamos:

$$\hat{w} = 2\ \mathrm{TAN}((2\cdot fu\cdot \pi/fm)/2)$$
$$\hat{w} = 2\ \mathrm{TAN}(fu\cdot \pi/fm)$$

Retomando la expresión 3.2, y despejando el valor de la constante <u>u</u> tenemos:

$$u = w\cdot \hat{w}$$

reemplazando las expresiones obtenidas tenemos:

$$u = 4\cdot \tan(wap/2)\cdot \tan(fu\cdot \pi/fm)$$

esta expresión es la que se utiliza en el programa para calcular el valor de la contante <u>u</u>. Análogamente esta constante también puede ser determinada así:

$$u = 4\cdot \tan(was/2)\cdot \tan(fl\cdot \pi/fm)$$

donde <u>fl</u> se conoce, pero <u>was</u> la frecuencia de supresión del filtro analógico pasa-bajos no transformado no se conoce. Despejando de esta última expresión, tenemos:

$$was = 2\ \arctan\left(\frac{u}{4\ \tan\left(\frac{fl\pi}{fm}\right)}\right)$$

Esta es la expresión que utiliza el programa para determinar el valor de <u>was</u>.

c) Para filtros pasa-banda.

Primero haremos un análisis de la función de transformación dada por la expresión 3.4. para poder despejar las contantes u y l.

Definamos dos contantes. Sea:

$$Bo = u - l$$
$$wo = u \cdot l \tag{4.1}$$

reemplazando estas constantes en la expresión 3.4 tenemos:

$$s = \frac{\bar{S}^2 + wo}{\bar{S}\,Bo}$$

ahora reemplacemos las siguientes igualdades: $s = j\omega$, $\bar{s} = j\bar{\omega}$

$$j\omega = \frac{-\bar{\omega}^2 + wo}{j\bar{\omega}\,Bo}$$

Cabe recalcar que las variables con la barra superior corresponden al filtro transformado, y las variables sin la barra superior corresponden al filtro no transformado.

Trabajando la última expresión, podemos llegar a:

$$\bar{\omega}^2 - Bo\omega\bar{\omega} - wo = 0$$

aplicando la fórmula de solución de la ecuación de segundo grado tenemos:

$$\bar{\omega} = \frac{\omega Bo}{2} \pm \sqrt{\left(\frac{\omega Bo}{2}\right)^2 + wo^2}$$

reemplazando ω por $\pm\omega_p$, tendremos cuatro posibles soluciones para $\bar{\omega}$, dos de las cuales serán menores que cero, y deberán ser desechadas. Por lo tanto las soluciones válidas son:

$$\varpi_{p1} = -\frac{\omega_p Bo}{2} + \sqrt{\left(\frac{\omega_p Bo}{2}\right)^2 + wo^2} \qquad (4.2)$$

$$\varpi_{p2} = \frac{\omega_p Bo}{2} + \sqrt{\left(\frac{\omega_p Bo}{2}\right)^2 + wo^2} \qquad (4.3)$$

Estos resultados nos conducen a verificar que por cada valor de frecuencia del filtro no transformado, se obtienen dos frecuencias en el filtro transformado. De esta manera a partir de la frecuencia de la banda de supresión ω_s, podremos obtener las frecuencias de supresión de las bandas de transición del filtro pasa-banda $\hat{\omega}_{s1}$ y $\hat{\omega}_{s2}$.

Restando 4.3 menos 4.2 obtendremos una expresión para Bo, así:

$$Bo = \frac{\varpi_{p2} - \varpi_{p1}}{\omega_p} \qquad (4.4)$$

Ahora sumando 4.2 y 4.3, y luego reemplazando la expresión 4.4, podremos obtener una expresión para wo, así:

$$wo = \varpi_{p2}\varpi_{p1} \qquad (4.5)$$

Conocidas las constantes Bo y wo, y haciendo uso de la definición de estas dadas por las expresiones 4.1, se forma un sistema de dos ecuaciones con dos incógnitas, u y l.

Resolviendo este sistema se llega a tener:

$$u = \frac{Bo + \sqrt{Bo^2 + 4wo}}{2} \qquad (4.6)$$

$$l = \frac{-Bo + \sqrt{Bo^2 + 4wo}}{2} \qquad (4.7)$$

Por el momento la única expresión que nos es útil en la determinación de las frecuencias de corte del filtro no transformado, es la 4.4. Como vemos esta expresión utiliza frecuencias digitales normalizadas, entonces normalicemos nuestras frecuencias (Ver la figura 4.1):

$$\omega_p = wap = 1 \ [rad] \rightarrow \hat{\omega}_{p1} = 2 \cdot fu \cdot \pi \ / \ fm \ [rad]$$
$$\hat{\omega}_{p2} = 2 \cdot fup \cdot \pi \ / \ fm \ [rad]$$

$$\omega_s = was = ? \ [rad] \rightarrow \hat{\omega}_{s1} = 2 \cdot fl \cdot \pi \ / \ fm \ [rad]$$
$$\hat{\omega}_{s2} = 2 \cdot fus \cdot \pi \ / \ fm \ [rad]$$

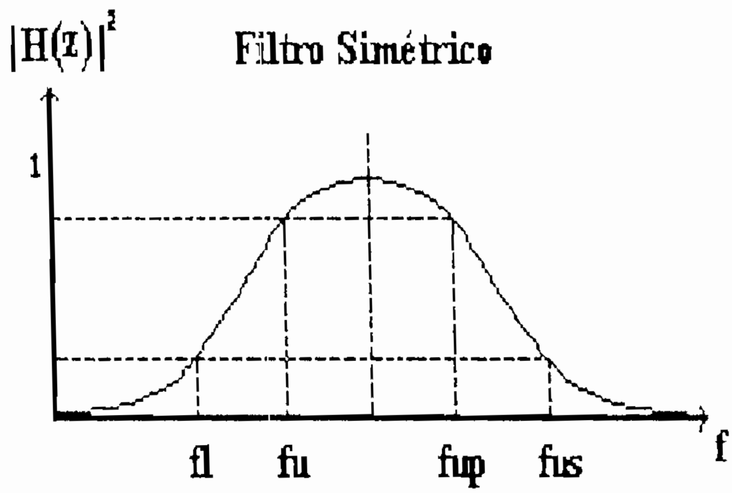


Figura 4.1: Filtro Pasa-Banda

donde la frecuencia <u>fus</u> se puede determinar por la simetría que tiene la respuesta de frecuencia. Entonces:

$$fus = fup + (fu - fl)$$

Entonces reemplazando en 4.4 estos valores previamente prealabeados, obtenemos:

$$BO = \frac{\tan\left(\frac{fup \cdot \pi}{fm}\right) - \tan\left(\frac{fu \cdot \pi}{fm}\right)}{\tan\left(\frac{wap}{2}\right)} \qquad (4.8)$$

$$Bo = \frac{\tan\left(\frac{(fup+fu-fl).\pi}{fm}\right) - \tan\left(\frac{fl.\pi}{fm}\right)}{\tan\left(\frac{was}{2}\right)} \quad (4.9)$$

Despejando was en la expresión 4.9, tenemos:

$$wap = 2.\arctan\left(\frac{\tan\left(\frac{(fup+fu-fl).\pi}{fm}\right) - \tan\left(\frac{fl.\pi}{fm}\right)}{Bo}\right) \quad (4.10)$$

En el programa se determina Bo con la expresión 4.8, y luego utilizando este valor encontramos Bo con la expresión 4.10.

d) Para filtros elimina-banda.

Tomando la función de transformación determinada en la sección 3.1.4, y siguiendo un proceso similar al anterior, se llega a determinar que:

$$Bo = (\varpi_{p2} - \varpi_{p1}).\omega_p \quad (4.11)$$

$$wo = \varpi_{p2}\varpi_{p1} \quad (4.12)$$

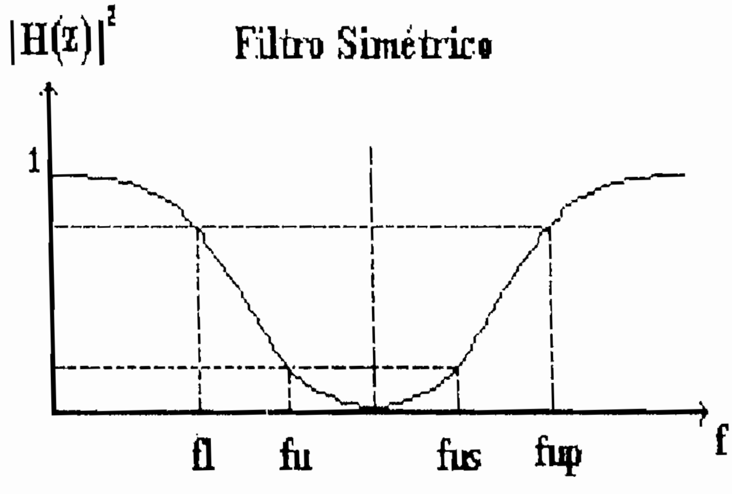


Figura 4.2: Filtro Elimina-Banda

normalicemos las frecuencias de corte del filtro elimina-banda visto en la figura 4.2:

$$\omega_p = wap = 1 \quad [rad] \qquad \hat{\omega}_{p1} = 2\cdot fl\cdot\pi \,/\, fm \quad [rad]$$

$$\hat{\omega}_{p2} = 2\cdot fup\cdot\pi \,/\, fm \quad [rad]$$

$$\omega_s = was = ? \quad [rad] \qquad \hat{\omega}_{s1} = 2\cdot fu\cdot\pi \,/\, fm \quad [rad]$$

$$\hat{\omega}_{s2} = 2\cdot fus\cdot\pi \,/\, fm \quad [rad]$$

donde la frecuencia <u>fus</u> se puede determinar por la simetría que tiene la respuesta de frecuencia. Entonces:

$$fus = fup - (\ fu - fl\ )$$

reemplazando en 4.11 tenemos:

$$Bo = 4.\left(\tan\left(\frac{fup.\pi}{fm}\right) - \tan\left(\frac{fl.\pi}{fm}\right)\right).\tan\left(\frac{wap}{2}\right) \tag{4.13}$$

$$Bo = 4.\left(\tan\left(\frac{(fup-fu+fl).\pi}{fm}\right) - \tan\left(\frac{fu.\pi}{fm}\right)\right).\tan\left(\frac{was}{2}\right) \tag{4.14}$$

de esta última expresión despejamos <u>was</u>:

$$was = 2.\arctan\left(\frac{Bo}{4.\left(\tan\left(\frac{(fup-fu+fl).\pi}{fm}\right) - \tan\left(\frac{fu.\pi}{fm}\right)\right)}\right) \tag{4.15}$$

En el programa se utiliza la expresión 4.13 para calcular la constante <u>Bo</u> y con este resultado calculamos la frecuencia <u>was</u> con la expresión 4.15

Las constantes <u>u</u> y <u>l</u> se calcularán con las expresiones 4.6 y 4.7, puesto que las constantes <u>Bo</u> y <u>wo</u> se definieron de la misma manera que en el filtro pasa-banda.

<u>CASO 2</u> : Filtros Elípticos con Transformación Analógica-Analógica.

En el diseño de filtros analógicos elípticos, no intervienen las frecuencias de corte en forma absoluta, pero si en forma relativa. La especificación de frecuencia para este tipo de filtros esta dada por la constante k (ksola en el programa), que es igual a la relación de las frecuencias de corte, así:

$$ksola = wap \; / \; was$$

reemplazando las frecuencias digitales con los valores normalizamos, y simplificando tenemos:

$$ksola = fl \; / \; fu$$

Esto significa que las frecuencias de corte del filtro analógico elíptico serán:

$$wap = \sqrt{(ksola)}$$
$$was = 1 \; / \; \sqrt{(ksola)}$$

A partir de este filtro y gracias a las transformaciones de banda podremos obtener cualquier tipo de filtro.
De hecho esto implica que aún los filtros pasa-bajos requerirán de la transformación de banda.

Procederemos a calcular solo las constantes de las funciones de transformación, ya que los valores wap y was ya son conocidos:

a) Para filtros pasa-bajos.

De la relación 3.1, y siguiendo los procesos de los casos anteriores, tenemos que:

$$u = \hat{\omega}_{p1} \; / \; wap$$
$$u = 2 \cdot \tan(fl \cdot \pi / fm) \; / \; wap$$
$$u = 2 \cdot \tan(fl \cdot \pi / fm) \; / \; \sqrt{(ksola)}$$

Esta última expresión es la que se emplea en el programa.

En este caso no se realiza el prealabeo de la frecuencia wap, porque la digitalización del filtro la obtendremos a partir del filtro transformado, y no a partir del filtro no transformado.

Aclararemos que en el caso anterior sí se prealabeó las frecuencias wap y was , pero se lo hizo para prever la utilización de subrutinas que tratan a los filtros sin tomar en cuenta las transformaciones de banda.

b) Para filtros pasa-altos.

$$u = \sqrt{(ksola)} \cdot 2 \cdot \tan(fu \cdot \pi / fm)$$

c) Para filtros pasa-banda.

$$Bo = \frac{2.(\tan(\frac{fup.\pi}{fm}) - \tan(\frac{fu.\pi}{fm}))}{\sqrt{ksola}}$$

d) Para filtros elimina-banda.

$$Bo = 2.(\tan(\frac{fup.\pi}{fm}) - \tan(\frac{fl.\pi}{fm})).\sqrt{ksola}$$

CASO 3 : Filtros No Elípticos con Transformación Digital-Digital.

En este caso puesto que la transformación es digital-digital, es decir que la conversión de analógico a digital ya se realizó, no hace falta considerar el prealabeo.

a) Para filtros pasa-bajos.

Para este tipo de filtros no se realiza la transformación de banda, por lo tanto, $\alpha = 0$ y las frecuencias digitales wap y was se las obtiene a partir de la normalización de las frecuencias conocidas en Hz. Así:

$$wap = 2 \cdot fl \cdot \pi \;/\; fm$$

$$was = 2 \cdot fu \cdot \pi \;/\; fm$$

b) Para filtros pasa-altos.

En la sección 3.2.2, se mencionó que se utilizaría el inverso de la expresión 3.8, para realizar la transformación de banda.

Entonces calculamos la constante α (alfasola en el programa), con:

$$alfasola = \frac{\cos\left(\frac{wap}{2} + \frac{fu.\pi}{fm}\right)}{\cos\left(\frac{wap}{2} - \frac{fu.\pi}{fm}\right)} \qquad (4.16)$$

similarmente podemos calcular alfasola con:

$$alfasola = \frac{\cos\left(\frac{was}{2} + \frac{fl.\pi}{fm}\right)}{\cos\left(\frac{was}{2} - \frac{fl.\pi}{fm}\right)}$$

De esta última expresión debemos obtener el valor de was, pero como no se puede despejar la variable was, recurriremos al siguiente proceso: Evaluamos esta expresión realizando incrementos de la variable was en una milésima de radián, hasta que coincida con el valor determinado por la expresión 4.16. El último valor adoptado por was, es el que se buscaba.

En el programa se determina el valor de alfasola con la expresión 4.16, y luego se realiza el proceso para encontrar el valor de was.

c) Para filtros pasa-banda.

La función de transformación para este tipo de filtros, tiene dos constantes a determinarse, α y k, (alfaso y kso en el programa). Tomando los resultados de la sección 3.2.3, tenemos:

$$alfaso = \frac{\cos\left(\frac{fup.\pi}{fm} + \frac{fu.\pi}{fm}\right)}{\cos\left(\frac{fup.\pi}{fm} - \frac{fu.\pi}{fm}\right)} \quad (4.17)$$

$$kso = \frac{\tan\left(\frac{wap}{2}\right)}{\tan\left(\frac{fup.\pi}{fm} - \frac{fu.\pi}{fm}\right)} \quad (4.18)$$

Similarmente utilizando las frecuencias de supresión podemos encontrar kso con:

$$kso = \frac{\tan\left(\frac{was}{2}\right)}{\tan\left(\frac{(fup+fu-fl).\pi}{fm} - \frac{fl.\pi}{fm}\right)}$$

Despejando de esta expresión la variable was, tenemos:

$$was = 2.\arctan\left(kso.\tan\left(\frac{(fup+fu-2.fl).\pi}{fm}\right)\right) \quad (4.19)$$

En el programa se utilizan las expresiones 4.17, 4.18 y 4.19

d) Para filtros elimina-banda.

Tomando los resultados de la sección 3.2.4, en el programa, se determinan las constantes de la función de transformación así:

$$alfaso = \frac{\cos\left(\frac{fup.\pi}{fm} + \frac{fl.\pi}{fm}\right)}{\cos\left(\frac{fup.\pi}{fm} - \frac{fl.\pi}{fm}\right)} \quad (4.20)$$

$$kso = \tan\left(\frac{wap}{2}\right).\tan\left(\frac{fup.\pi}{fm} - \frac{fl.\pi}{fm}\right) \quad (4.21)$$

Similarmente utilizando las frecuencias de supresión podemos encontrar kso con:

$$kso = \tan\left(\frac{was}{2}\right)\tan\left(\frac{(fup+fl-fu).\pi}{fm} - \frac{fu.\pi}{fm}\right)$$

Despejando de esta expresión la variable <u>was</u>, tenemos:

$$was = 2.\arctan\left(\frac{kso}{\tan\left(\frac{(fup+fl-2.fu).\pi}{fm}\right)}\right) \tag{4.22}$$

Esta expresión se utilizará en el programa.

<u>CASO 4</u> : Filtros Elípticos con Transformación Digital-Digital.

Para el caso de los filtros elípticos , sabemos que, las frecuencias del filtro analógico no transformado ya están determinadas. De esta manera solo calcularemos las constantes de las funciones de transformación, usando las mismas expresiones que para el caso anterior, con la diferencia que en lugar de la variable wap reemplazaremos su equivalente:

$$wap = \sqrt{(ksola)}$$

a) Para filtros pasa-bajos.

Por la determinación implícita de las frecuencias <u>wap</u> y <u>was</u>, se hace indispensable la transformación de banda pasa-bajos a pasa-bajos. Con la expresión 3.7, determinamos la constante α:

$$alfasola = \frac{sen\left(\frac{\sqrt{ksola}}{2} - \frac{fl.\pi}{fm}\right)}{sen\left(\frac{\sqrt{ksola}}{2} + \frac{fl.\pi}{fm}\right)}$$

b) Para filtros pasa-altos.

$$alfasola = \frac{\cos\left(\frac{\sqrt{ksola}}{2} + \frac{fu.\pi}{fm}\right)}{\cos\left(\frac{\sqrt{ksola}}{2} - \frac{fu.\pi}{fm}\right)}$$

c) Para filtros pasa-banda.

$$alfaso = \frac{\cos\left(\frac{fup.\pi}{fm} + \frac{fu.\pi}{fm}\right)}{\cos\left(\frac{fup.\pi}{fm} - \frac{fu.\pi}{fm}\right)}$$

Esta expresión es idéntica a la utilizada en el filtro no elíptico.

$$kso = \frac{\tan\left(\frac{\sqrt{ksola}}{2}\right)}{\tan\left(\frac{fup.\pi}{fm} - \frac{fu.\pi}{fm}\right)}$$

d) Para filtros elimina-banda.

$$alfaso = \frac{\cos\left(\frac{fup.\pi}{fm} + \frac{fl.\pi}{fm}\right)}{\cos\left(\frac{fup.\pi}{fm} - \frac{fl.\pi}{fm}\right)}$$

Esta expresión es idéntica a la utilizada en el no elíptico.

$$kso = \tan\left(\frac{\sqrt{ksola}}{2}\right).\tan\left(\frac{fup.\pi}{fm} - \frac{fl.\pi}{fm}\right)$$

4.3.2.3 <u>Subrutina orden</u>.

Esta subrutina sirve para determinar el orden que tiene el filtro a diseñarse. Conocido el orden podemos saber el número de secciones que se deben construir para simular el filtro. La mayoría de las secciones son de segundo grado, pero cuando el orden es impar se complementa con una sección de primer grado.

El orden del filtro depende fundamentalmente del tipo de filtro analógico. También se debe tomar en cuenta los casos en los cuales se debe realizar el prealabeo, un olvido de esta consideración puede llevar a determinar un valor de erróneo del orden.

Primeramente calcularemos las constantes de atenuación ε y λ a partir de los valores especificados en decibelios A y B. Para esto utilizaremos las expresiones 1.3 y 1.4 respectivamente. Estos valores son comunes para todos los tipos de filtros analógicos.

Para los filtros de Butterworth, calculamos el orden con la expresión 1.2. Luego la frecuencia Ωc será determinada con la expresión 1.5 (en función de ε) en el caso de no utilizar la transformada bilineal, y con la expresión 1.6 (en función de λ) cuando se utilice la transformada bilineal. Lo que se quiere con esto es reducir un poco el ancho de banda para el caso de la invariancia de pulso e impulso, debido a las limitaciones que este tipo de conversión presenta. En cambio cuando se usa la transformada bilineal se toma la banda más ancha porque no hay restricciones, además como está prealabeada la respuesta digital cae más rápido que la analógica. Para mayor claridad vea la figura 1.5. La constante Ωc en el programa será reconocida por wac.

Para los filtros de Chevishev, determinamos el orden con la expresión 1.24, para el tipo I y tipo II. Luego calculamos la constante a, para el tipo I en función de ε con 1.21 y para el tipo II en función de λ con 1.27. Este valor posteriormente se utilizará en la determinación de polos. En el programa la constante a está representada por la variable alf.

Para los filtros Elípticos, determinamos el orden con la expresión dada en el formulario páginas 59,60 y 61.
Previamente a este cálculo se debe determinar las constante D, k', q_o, y q, en este orden. En el programa estas constantes toman los siguientes nombres:

D	:	D
k'	:	kprima
q_o	:	qo
q	:	q

Sabemos que el orden del filtro determinado con las fórmulas para cada caso, no siempre es un entero. Entonces tomamos el entero superior mas cercano, así:

n = INT(nt) + 1

donde nt es el valor teórico calculado del orden.
Finalmente analizamos la paridad. Dividimos el valor del orden n para dos, luego a este resultado le restamos su parte entera, por tanto si el último resultado es distinto de cero, quiere decir que el orden es impar, caso contrario es par.
El resultado de la paridad se lo lleva en la variable p, si p=1 el orden es par y si p=2 el orden es impar.

4.3.2.4 Subrutina polos.

Esta subrutina se encarga de calcular la función de transferencia del filtro analógico. En el caso de los filtros de Butterworth y Chevishev tipo I, calcularemos los polos y la constante del numerador. En el caso de los filtros de Chevishev tipo II y Elípticos, se determina además los ceros, para conocer por completo la función de transferencia.

Como se vio en el capitulo 1, los polos son números complejos, que se encuentran en el denominador de la función de transferencia. Además siempre encontramos pares conjugados de números complejos, esto ayudará posteriormente a la formación de secciones.

Sea a+jb un polo con su conjugado igual a a-jb, entonces la función de transferencia será:

$$H(s) = \frac{K}{[s - (a+jb)]\,[s - (a-jb)]}$$

Esta función aparenta ser compleja, por los coeficientes complejos que tiene. Ahora multiplicando los dos factores del denominador, tenemos:

$$H(s) = \frac{K}{s^2 - s(a+jb+a-jb) + (a+jb)(a-jb)}$$

$$H(s) = \frac{K}{s^2 - s(2a) + (a^2+b^2)}$$

Si nos fijamos han desaparecido las partes imaginarias de la expresión. Esta expresión se la conoce como " Una sección de segundo grado", debido que la variable s tiene exponente 2.

Esta operación será efectuada con todos los pares conjugados de polos, a fin de convertir la función de transferencia en un producto de secciones de segundo grado.

El número de polos existentes en una función de transferencia es igual al orden del filtro n, y el número de secciones cuadradas será igual a la mitad de n. Cuando el orden es impar el número de secciones es igual a la mitad de (n+1), de donde todas las secciones son cuadradas excepto una que es de primer grado.

Cuando el orden es impar, aparece un polo que es una cantidad real y que coincide con él semi-eje real negativo del plano complejo "S". Este polo es el que genera la única sección de primer grado. Así:

$$H(s) = \frac{1}{s-s_r}$$

donde s_r es el polo real.

Esta subrutina se encarga de calcular los polos y además los coeficientes de las secciones. En el programa se interpreta estos coeficientes con las siguientes variables:

- a (parte real del polo) por: s(i).real

- b (parte imaginaria del polo) por: s(i).imag
- (-2·a) (coeficiente de s) por: gama(1,i)
- (a^2+b^2) (término independiente) por: gama(2,i)
- s_r (polo real) por: s(sec+1).real

donde i varía desde 1 hasta sec, donde sec es el número de secciones cuadráticas de la función. Esta variable fue determinada en el momento de analizar la paridad en la subrutina "orden".

La función de transferencia quedaría determinada así:

$$H(s) = \frac{kanaloga}{(s^2+\gamma_{1,1}s+\gamma_{2,1})\ldots(s^2+\gamma_{1,sec}s+\gamma_{2,sec})\ldots(s+\gamma_{2,sec+1})}$$

donde: $\gamma_{1,i}$ = gama(1,i)

$\gamma_{2,i}$ = gama(2,i)

Dependiendo del tipo de filtro analógico realizamos los cálculos. Para esto utilizaremos un SELECT CASE de la variable tipana, así como se indica en el diagrama de flujo 4.1:

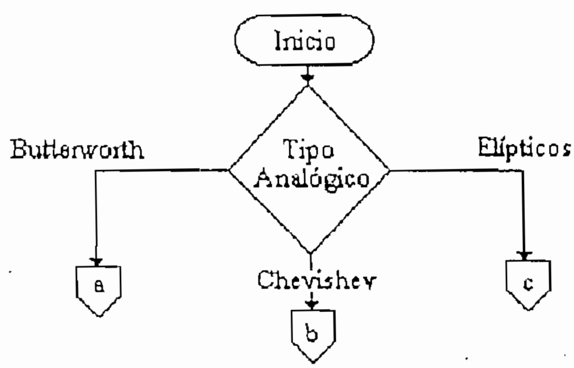


Diagrama 4.1: Subrutina Polos

CASO 1: Para los filtros de Butterworth.

Determinamos los polos con la expresión encontrada en el numeral 1.1.1, y luego formamos las secciones de segundo grado. Si n es impar determinamos la sección de primer grado. Posteriormente determinamos la constante del numerador, que en el programa le hemos llamado kanaloga. Esta constante puede ser determinada de dos maneras, una de ellas es la indicada en la sección 1.1.2 de este trabajo; la otra se expone a continuación:

Tomemos la primera función de transferencia obtenida para el tipo de Butterworth, así:

$$|H(s)|^2 = \frac{1}{1 + \left(\frac{s}{j\Omega c}\right)^{2n}}$$

$$|H(s)|^2 = \frac{(j\Omega c)^{2n}}{(j\Omega c)^{2n} + s^{2n}}$$

Extrayendo la raíz cuadrada a los dos miembros, tenemos:

$$|\,H(s)\,| = \frac{(j\Omega c)^{n}}{\sqrt{(j\Omega c)^{2n} + s^{2n}}}$$

De esta expresión podemos deducir que la constante del numerador es igual a:

$$\text{kanaloga} = (j\cdot\Omega c)^{n}$$

Y como se trata del módulo de la función de transferencia, kanaloga siempre va a ser positiva. Por lo tanto:

$$\text{kanaloga} = (\Omega c)^{n}$$

De esta forma es que determinaremos en el programa la kanaloga, donde Ωc es reemplazada por la variable wac.

Para mayor claridad indicaremos como se opera el "CASO 1", con el diagrama de flujo 4.2.

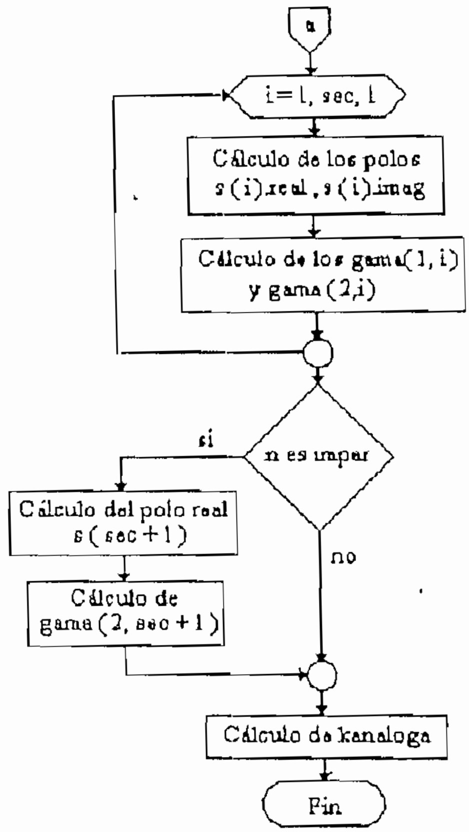


Diagrama 4.2: Caso 1 (Sub.Polos)

CASO 2,3: Para los filtros de Chevishev.

Calculamos los polos con las expresiones 1.22 y 1.23, que corresponden al filtro tipo I. En el programa las variables adoptan los siguientes nombres:

- Ωp por wap
- α por alf
- k por i
- σ_k por s(i).real
- Ω_k por s(i).imag

Seguidamente calculamos los coeficientes de las secciones de

segundo grado, y luego el de la sección de primer grado si la hay.
Para calcular la kanaloga evaluaremos la función de transferencia en s = 0. Tómese en cuenta que la amplitud de H(0) depende de la paridad del orden (revise la sección 1.2.2)

Para n impar tenemos:

$$H(0) = 1 = \frac{kanaloga}{gama(2,1)\ gama(2,2)\ldots gama(2,sec+1)}$$

por lo tanto:

kanaloga = gama(2,1)·gama(2,2)··· gama(2,sec+1)

Y para n par tenemos:

$$H(0) = attp(adi) = \frac{kanaloga}{gama(2,1)\ gama(2,2)\ \ldots\ gama(2,sec)}$$

donde attp(adi) es el valor adimensional de la atenuación de paso. Este valor queda determinado en función del valor dado en decibelios, así:

[- attp(dB)] = 20·log(attp(adi))

entonces:

attp(adi) = 10^(-attp(dB)/20) (4.23)

el valor de kanaloga queda determinado por:

kanaloga = attp(adi)·gama(2,1)·gama(2,2)··· gama(2,sec+1)

Si el caso es determinar los polos del filtro tipo II, los obtendremos a partir de los polos del filtro tipo I ya determinados aquí. La relación que guardan los polos del filtro tipo I y el tipo II se ve en la sección 1.3.2. , y en base a esta realizaremos el cálculo. Debemos tomar en cuenta que para el filtro tipo II ya no existen solo polos, sino que intervienen también los ceros. Los ceros se obtienen de la expresión 1.26, y estos se encuentran también con su respectivo conjugado. Esto quiere decir que ahora las secciones de segundo grado incluirán un cero de segundo orden igual a:

$$C_k^2 = \left(\frac{was}{sen\left(\frac{2i+n-1}{2n}\right)\pi}\right)^2$$

donde en el programa:

- c(i) reemplaza a C_k^2
- was reemplaza a Ωs
- i reemplaza a k

Entonces cada sección cuadrada queda así:

$$\frac{s^2 + c(i)}{s^2 + gama(1,i)\,s + gama(2,i)}$$

La sección de primer orden, simplemente no tiene cero. Esta consideración se hace debido a que el cero obtenido para esta sección con la expresión 1.26 es igual a infinito.

Luego de determinar los coeficientes gama determinaremos el polo real y luego la constante kanaloga. La constante kanaloga se determinará a partir de H(0), como se indica en la sección 1.3.3.

En el programa las multiplicaciones sucesivas se las realiza mediante un lazo de repetición. Partimos de un kanaloga igual 1, luego a este le multiplicamos el primer término gama(2,i)/c(i), luego a este resultado le multiplicamos el segundo término gama(2,i)/c(i), etc, hasta llegar a i igual a sec. En caso de ser n impar a todo este resultado se le multiplica el término gama(2,sec+1).

El caso 2 queda representado en el diagrama de flujo 4.3 y el caso 3 por el diagrama 4.4, así:

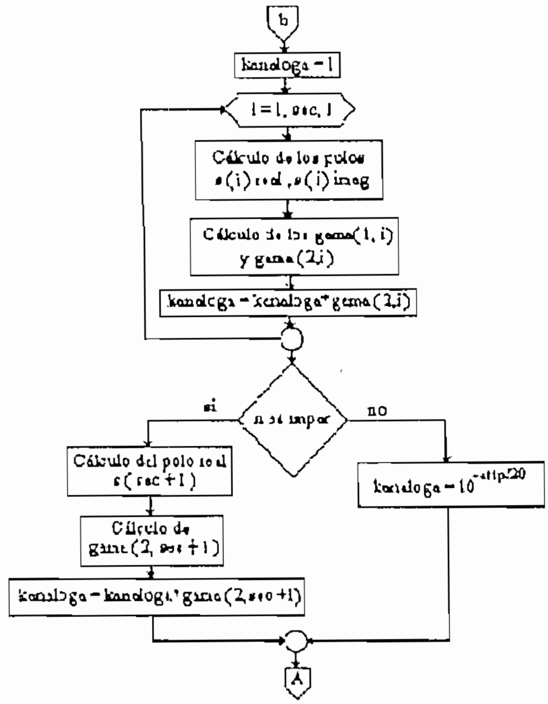


Diagrama 4.3: Caso 2 (Sub.Polos)

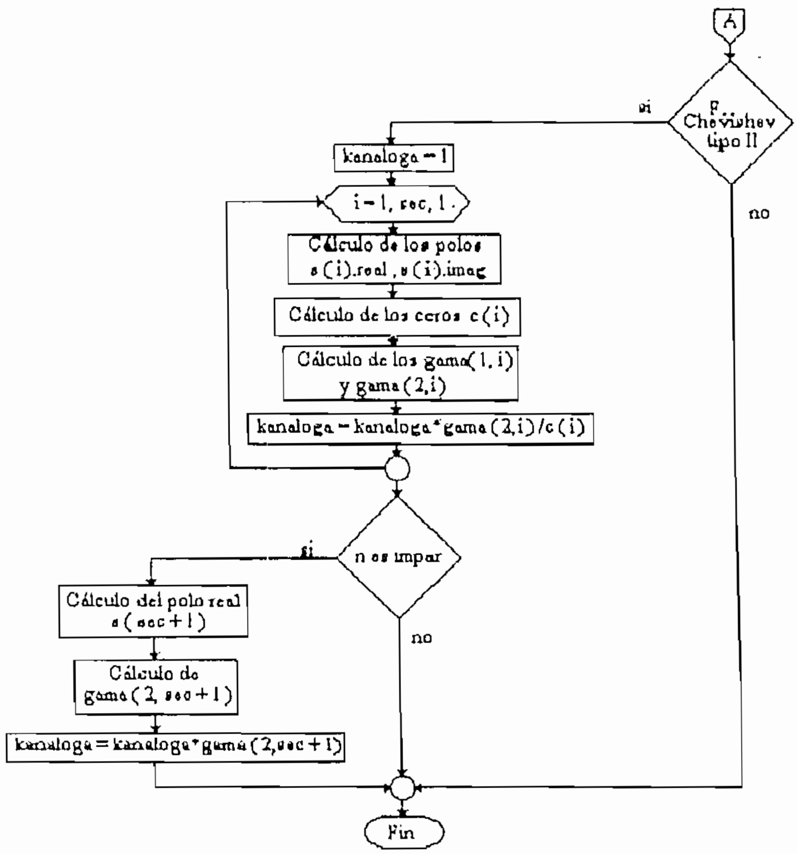


Diagrama 4.4: Caso 3 (Sub.Polos)

CASO 4: Para los filtros Elípticos.

Para este caso el cálculo de los polos no se reduce a la utilización de una sola fórmula, sino que para llegar al resultado tenemos que determinar algunas variables previas. Siguiendo el proceso descrito en la sección 1.4, conseguiremos una función de transferencia normalizada, cuya forma es similar a la del filtro Chevishev tipo II. En la función:

$$Ha(s) = \frac{Ho}{Do(s)} \prod_{i=1}^{r} \frac{s^2 + A_{0i}}{s^2 + B_{1i}s + B_{0i}}$$

reemplazamos nuestras variables, así:

- c(i) por A_{0i}
- gama(1,i) por B_{1i}
- gama(2,i) por B_{0i}
- Kanaloga por Ho
- sec por r

La sección de primer grado que aparece cuando el orden es impar, esta incluida en la función Do(s), donde el polo real es igual a $-\sigma_o$.

Calculamos primero **Λ** (En el programa: LAMBDA), luego, para calcular σ_o (sigmacero) previamente debemos determinar el valor de las series infinitas que ésta tiene. Estas series infinitas se las determina por medio de un lazo de repetición DO...WHILE, hasta que los términos de la serie sean menores al valor de 0.000001. El valor de 0.000001 es tomado arbitrariamente.

Determinamos W (doblev) y luego, realizamos un lazo de repetición FOR...NEXT, que va desde i=0 hasta i=sec. Dentro de esta lazo se determinarán en el siguiente orden los valores de:

- Ω_i en el programa: omega.
- V_i en el programa: v

- c(i)
- gama(2,i)
- gama(1,i)
- s(i).real
- s(i).imag

El cálculo de omega depende de la paridad del orden, por lo tanto se hace necesario utilizar un sentencia IF...ELSE...END.

Como se observa, los polos son determinados a partir de los coeficientes de la sección de segundo grado, de manera inversa a los procesos anteriores.

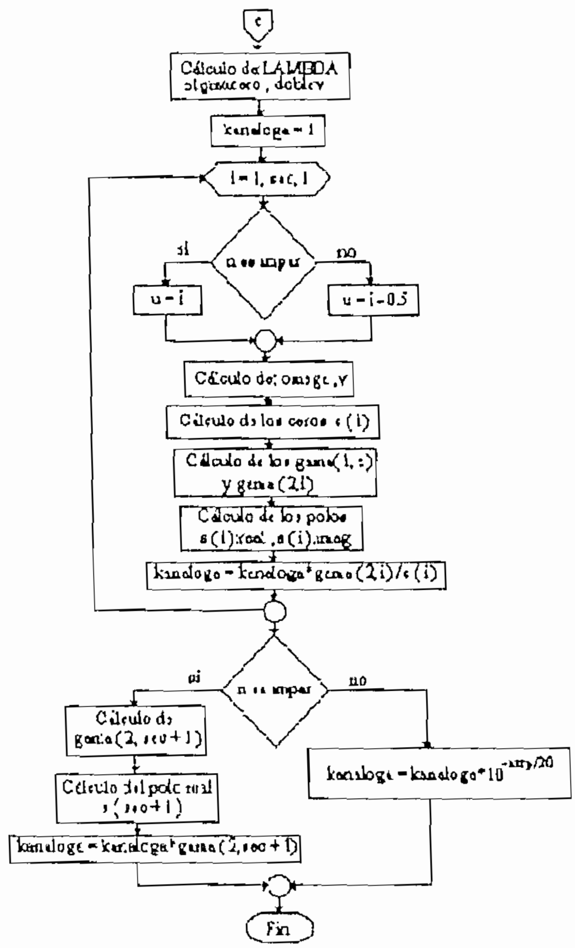


Diagrama 4.5: Caso 4 (Sub.Polos)

Teniendo los valores de gama(2,i) y c(i), el cálculo de la kanaloga se efectúa así:

$$kanaloga = \prod_{i=1}^{6\cdot c} \frac{gama(2,i)}{c(i)}$$

Cuando n es impar se multiplica el polo real al valor encontrado de kanaloga, y cuando n es par se multiplica por la constante 10^(attp/20). Esta constante interviene por la presencia del rizado en la banda de paso, tal como se hizo para el filtro Chevishev tipo I.

En el diagrama 4.5 se indican estas funciones de una forma mas clara.

4.3.2.5 Subrutina frecuenciana

Hasta aquí en las subrutinas previas, se ha determinado la función de tranferencia del filtro analógico en base del cual se diseñará el filtro digital.

Esta subrutina se encarga de calcular el módulo de la respuesta de frecuencia del filtro analógico.

Para esto vamos ha reemplazar s por jω en la función de transferencia obtenida.

En el caso que la función tiene solo polos, al reemplazar s por jω, el denominador de la k-ésima sección de segundo grado queda así:

$$[-\omega^2] + gama(1,k)\cdot j\omega + gama(2,k)$$

separando la parte real y la imaginaria, tenemos:

$$[\ gama(2,k) - \omega^2\ ] + \ j\cdot gama(1,k)\cdot\omega$$

el módulo de esta expresión es:

$$\sqrt{[gama(2,k) - \omega^2]^2 + [gama(1,k)\ \omega]^2}$$

El resultado general del denominador es igual al producto de las secciones de k=1 á k=sec. En el caso de n impar se debe incluir el término de la sección de primer grado, cuyo módulo es igual a:

$$\sqrt{[gama(2,sec+1)]^2 + [\omega]^2}$$

Para obtener la función de transferencia, debemos dividir la contante <u>kanaloga</u> para el resultado del denominador. Para <u>n</u> par la función es:

$$|H(w)| = \frac{kanaloga}{\prod_{k=1}^{sec} \sqrt{[gama(2,k) - \omega^2]^2 + [gama(1,k)\,\omega]^2}}$$

La evaluación de la respuesta de frecuencia se la hace reemplazando ω por $2\pi i / 304$, donde i va desde 1 hasta 304. De esta manera el rango de evaluación de la respuesta de frecuencia es de $\omega=2\pi/304$ [rad/seg] á $\omega=2\pi$ [rad/seg].

La respuesta de frecuencia se guarda en un arreglo de 304 puntos llamado <u>ha(i)</u> . La función se evalúa cada intervalo de $2\pi/304$.

Cuando la función de transferencia tiene polos y ceros, al resultado de <u>ha(i)</u> obtenido hasta aquí, se lo debe multiplicar un factor <u>b</u> . Este factor necesariamente depende del valor de los ceros. En la subrutina polos se había visto que para el caso de las funciones que tienen ceros, cada k-ésima sección de segundo grado, posee en el numerador el siguiente término:

$$s^2 + c(k)$$

reemplazando s por $j\omega$, tenemos:

$$[-\omega^2] + c(k)$$

esta expresión es real, por lo tanto es igual a su módulo. Al multiplicar los términos de k=1 hasta k=sec, obtendremos el factor b. En el diagrama 4.6 se encuentra mas claro lo dicho.

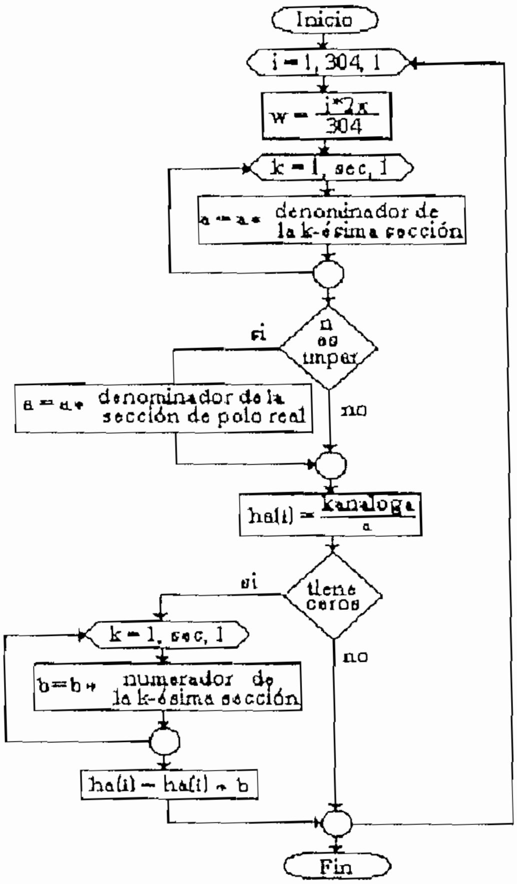


Diagrama 4.6: Subrutina "frecuenciana"

Terminado el diseño de los filtros analógicos, procedemos a digitalizar este filtro. Dependiendo del método de conversión elegido, el proceso es distinto.

Para el caso de las conversiones por Invariancia de Impulso o Pulso, los cálculos a realizarse tienen cierta similitud; existen dos subrutinas que utilizan mutuamente ambas conversiones, estas son:

- FPARCIALES
- TRANSFRECUENCIAPULSO

Las subrutinas utilizadas con la Transformación Bilineal son exclusivas de esta, porque se incluye el alabeo de frecuencia y la transformación de banda analógica-analógica.

Por medio de un SELECT CASE de la variable tipconv separamos el trabajo:

CASO 1,3: Utilizando conversión por Invariancia

4.3.2.6 Subrutina fparciales

El propósito de esta subrutina es determinar la función de transferencia del filtro analógico en términos de fracciones parciales.

Para ello como:

$$H(s) = \frac{kanaloga}{(s-s_1)(s-s_2)\ldots(s-s_n)} = \frac{a_1}{s-s_1} + \frac{a_2}{s-s_2} + \ldots + \frac{a_n}{s-s_n}$$

el proceso de encontrar las fracciones parciales se reduce a determinar el valor de los coeficientes $a_1, a_2, \ldots, a_n$. Esta expresión es válida solo para el caso en que el grado del numerador es menor al grado del denominador.

Para determinar el coeficiente j-ésimo a_j, multiplicamos la función H(s) por el término $(s-s_j)$, así:

$$\frac{kanaloga(s-s_j)}{(s-s_1)\ldots(s-s_j)\ldots(s-s_n)} = \frac{a_1(s-s_j)}{s-s_1} +\ldots \frac{a_j(s-s_j)}{s-s_j} \ldots + \frac{a_n(s-s_j)}{s-s_n}$$

simplificando, tenemos:

$$\frac{kanaloga}{(s-s_1)\,..\quad..\,(s-s_n)} = \frac{a_1(s-s_j)}{s-s_1} + ..\ a_j\ .. + \frac{a_n(s-s_j)}{s-s_n}$$

ahora evaluemos la igualdad en $s=s_j$:

$$\frac{kanaloga}{(s_j-s_1)\,..\,(s_j-s_{j-1})\,(s_j-s_{j+1})\,..\,(s_j-s_n)} = a_j$$

de esta manera quedan determinados los coeficientes de la función en fracciones parciales.

Para el caso en que las funciones de transferencia tienen ceros, siguiendo el mismo proceso, la expresión de a_j queda así:

$$\frac{kanaloga(s_j^2+c_1)\,(s_j^2+c_2)\,..\,(s_j^2+c_{aec})}{(s_j-s_1)\,..\,(s_j-s_{j-1})\,(s_j-s_{j+1})\,..\,(s_j-s_n)} = a_j$$

El proceso de este cálculo se complica debido a que los polos son números complejos, lo que implica todo un desarrollo de operaciones con complejos.

En el programa todo este desarrollo se lo hace por partes, utilizando las expresiones anteriores:

1. Determinamos los términos (s_j-s_i) del denominador, donde i varía entre 1 y n, exceptuando el caso en que i=j. Estos términos los hemos llamado ini(m), donde m esta en el rango de 1 y (n-1).

2. Multiplicamos los términos ini(m), para obtener el denominador, considerando que es un producto de complejos. Este resultado se lo asigna a la variable den(j), donde j indica que pertenece al denominador de la expresión para determinar el coeficiente a_j.

3. Si la función de transferencia es de solo polos, el proceso termina en racionalizar la expresión compleja:

$$\frac{kanaloga}{(den_j.real) + j(den_j.imag)}$$

no confundir, el j de la parte imaginaria del complejo, con j subíndice de la variable den . El resultado de los a_j, se almacena en la variable aa(j) .

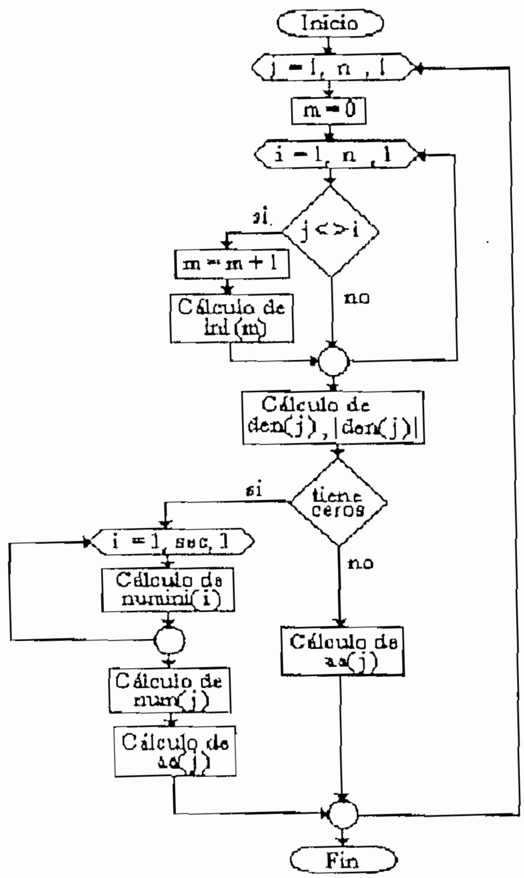


Diagrama 4.7: Subrutina "fparciales"

4. Si la función de transferencia incluye ceros, determinamos las términos s_j^2 - c(i), y los guardamos en la variable numini(m). Luego efectuamos el producto de complejos entre los términos numini(m) y la constante kanaloga , el resultado lo asignamos a la variable num(j). Posteriormente realizamos la

siguiente división para obtener el resultado final:

$$(a_j.real) + j(a_j.imag) = \frac{(num_j.real) + j(num_j.imag)}{(den_j.real) + j(den_j.imag)}$$

En los resultados obtenidos podemos darnos cuenta que los coeficientes aa(j) son pares de complejos conjugados. En el caso de ser n impar aparece un coeficiente aa(j) real.

Apreciemos lo descrito en el diagrama de flujo 4.7

Para el caso de tener ceros en el numerador y de que el orden del filtro sea par, nos daremos cuenta que el grado del numerador es igual al grado del denominador, entonces debemos tomar como numerador al residuo de la división de estos. El proceso de encontrar el residuo, se lo puede evitar, porque al realizar las operaciones para determinar los coeficientes de las fracciones parciales, da igual el tomar como numerador el mismo de la función de transferencia que el residuo, con la consideración de que un término del resultado de las fracciones parciales es la constante <u>kanaloga</u> , es decir:

$$H(s) = kanaloga + \frac{a_1}{s-s_1} + \frac{a_2}{s-s_2} + \ldots + \frac{a_n}{s-s_n}$$

Esto se tomará muy en cuenta el momento de encontrar la respuesta de frecuencia digital.

Determinadas las fracciones parciales, se debe reemplazar directamente la expresión 2.3 para el caso de Invariancia de Impulso o la expresión 2.16 para el caso de la Invariancia de Pulso. Utilizando la instrucción IF...THEN, separamos el trabajo. Cuando utilicemos Invariancia de Impulso llamaremos a la subrutina "transformacionimpulso" y cuando utilicemos la Invariancia de Pulso llamaremos a la subrutina "transformacionpulso".

4.3.2.7 <u>Subrutina transformacionimpulso</u>

Esta subrutina se encarga de determinar la función de transferencia del filtro digital utilizando la Invariancia de Impulso. Esta función ya incluye la transformación de banda, por lo que demanda un tratamiento especial para cada tipo de banda.

Inicialmente se digitaliza el filtro pasa-bajos analógico. La función de transferencia resultante digital H(z) esta formada por la suma de secciones de segundo grado cuya forma es:

$$\frac{\beta_{0,i} + \beta_{1,i} z^{-1}}{1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2}}$$

en el programa los coeficientes son calculados con los siguientes nombres:

- $\alpha_{1,i}$ con alfa(1,i)
- $\alpha_{2,i}$ con alfa(2,i)
- $\beta_{0,i}$ con beta(0,i)
- $\beta_{1,i}$ con beta(1,i)

Para obtener estos coeficientes tomaremos dos fracciones parciales de la función de transferencia analógica, cuyos coeficientes sean complejos conjugados, así:

$$\frac{a_{re} + ja_{im}}{s - (s_{re} + js_{im})}, \frac{a_{re} - ja_{im}}{s - (s_{re} - js_{im})}$$

ahora estos términos los reemplazamos por la expresión 2.3:

$$\frac{a_{re} + j a_{im}}{1 - e^{(s_{re}+js_{im})} z^{-1}}, \frac{a_{re} - j a_{im}}{1 - e^{(s_{re}-js_{im})} z^{-1}}$$

en este momento se cambia de la variable s a la variable z. Multiplicando estos términos tenemos:

$$\frac{2a_{re} + [-2\, e^{s_{re}}(a_{re} COS\, s_{im} + a_{im} SEN\, s_{im})]\, z^{-1}}{1 + (-2\, e^{s_{re}} COS\, s_{im})\, z^{-1} + e^{2 s_{re}} z^{-2}}$$

de este resultado podemos concluir que, para la i-ésima sección de la función digitalizada del filtro pasa-bajos, los coeficientes son iguales a (presentaremos los resultados en función de las variables utilizadas en el programa):

$$beta(0,i) = 2\, aa(i)_{re}$$

$$beta(1,i) = -2\, e^{s(i)_{re}} [aa(i)_{re} COS\, s(i)_{im} + aa(i)_{im} SEN\, s(i)_{im}]$$

$$alfa(1,i) = -2\, e^{s(i)_{re}} COS\, s(i)_{im}$$

$$alfa(2,i) = e^{2 s(i)_{re}}$$

en el programa se evalúan los coeficientes de todas las secciones cuadradas que existen, mediante un lazo de repetición FOR...NEXT donde i varía de 1 hasta sec.

Para el caso en que n es impar, la sección de primer grado es:

$$\frac{aa(sec+1)}{1+[-e^{s(sec+1)}]}$$

entonces los coeficientes son:

$$beta(0,sec+1) = aa(sec+1)$$

$$alfa(1,sec+1) = -e_{s(sec+1)}$$

A partir de esta función de transferencia, se determina la función de transferencia del filtro con transformación de banda. Para esto se tienen cuatro casos, dependiendo del tipo de banda del filtro a obtenerse, así:

a) Para los filtros pasa-bajos.

En realidad este caso tiene importancia solo para los filtros Elípticos, porque necesitan de esta transformación de banda, para lograr que el filtro de $\Omega c=1$ [rad] cumpla con los requerimientos de frecuencia dados. Sin embargo, para el caso de los filtros no elípticos se ejecuta esta parte del programa haciendo alfasola = 0.

Reemplazando la función $G(\check{z}^{-1})$ obtenida en la sección 3.2.1, por z^{-1}, tenemos una función de transferencia cuyas secciones de segundo grado son de la forma:

$$\frac{kpulso(i)\,[1+\beta_{0,i}\check{z}^{-1}+\beta_{1,i}\check{z}^{-2}]}{1+\alpha_{1,i}\check{z}^{-1}+\alpha_{2,i}\check{z}^{-2}} \qquad (4.24)$$

los coeficientes α y β son recalculados en función de la constante alfasola y de los valores α y β antes calculados. La variable kpulso(i) es un factor que multiplica a la sección i-ésima. Los nuevos valores de α y β, y el valor de kpulso(i) se determinan así:

$$\beta_{0,i} = \frac{-2\beta_{0,i}alfasola+\beta_{1,i}(1+alfasola^2)}{\beta_{0,i}-\beta_{1,i}alfasola}$$

$$\beta_{1,i} = \frac{alfasola[\beta_{0,i}alfasola-\beta_{1,i}]}{\beta_{0,i}-\beta_{1,i}alfasola}$$

$$\alpha_{1,i} = \frac{-2\,alfasola\,(1+\alpha_{2,i})+\alpha_{1,i}(1+alfasola^2)}{1-\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2-\alpha_{1,i}alfasola+\alpha_{2,i}}{1-\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

$$kpulso(i) = \frac{\beta_{0,i}-\beta_{1,i}alfasola}{1-\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

La sección de primer grado tiene la forma siguiente:

$$\frac{kpulso(sec+1)\,\beta_{0,sec+1}}{1+\alpha_{1,sec+1}z^{-1}} \tag{4.25}$$

donde:

$$\beta_{0,sec+1} = -alfasola$$

$$\alpha_{1,sec+1} = \frac{\alpha_{1,sec+1} - alfasola}{1-\alpha_{1,sec+1}alfasola}$$

$$kpulso(sec+1) = \frac{\beta_{0,sec+1}}{1-\alpha_{1,sec+1}alfasola}$$

b) Para los filtros pasa-altos

Reemplazando la función de transformación de banda $G(z^{-1})$ que se determinó en la sección 3.2.2, obtenemos una función de transferencia $H(z)$ cuyas secciones son de la forma 4.24. Los coeficientes <u>α</u> , <u>ß</u> y <u>kpulso</u> de la i-ésima sección se los determina con las expresiones siguientes:

$$\beta_{0,i} = \frac{-2\beta_{0,i}alfasola - \beta_{1,i}(1+alfasola^2)}{\beta_{0,i}+\beta_{1,i}alfasola}$$

$$\beta_{1,i} = \frac{alfasola[\beta_{0,i}alfasola + \beta_{1,i}]}{\beta_{0,i}+\beta_{1,i}alfasola}$$

$$\alpha_{1,i} = \frac{-2\,alfasola(1+\alpha_{2,i}) - \alpha_{1,i}(1+alfasola^2)}{1+\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2 + \alpha_{1,i}alfasola + \alpha_{2,i}}{1+\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

$$kpulso(i) = \frac{\beta_{0,i}+\beta_{1,i}alfasola}{1+\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^3}$$

Los coeficientes de la sección de primer grado, que es igual a la 4.25, se los calcula con:

$$\beta_{0,sec+1} = -alfasola$$

$$\alpha_{1,sec+1} = \frac{-\alpha_{1,sec+1} - alfasola}{1 + \alpha_{1,sec+1}\,alfasola}$$

$$kpulso(sec+1) = \frac{\beta_{0,sec+1}}{1 + \alpha_{1,sec+1}\,alfasola}$$

Estas expresiones, respecto a las expresiones determinadas para los filtros pasa-bajos, solo difieren en el signo de algunos términos.

c) Para los filtros pasa-banda

Para el caso de los filtros pasa-banda y elimina-banda, al reemplazar la función de transformación $G(\check{z}^{-1})$ respectiva, se obtiene una función de transferencia cuyas secciones son de la forma:

$$\frac{kpulso(i)\,[1 + \beta_{0,i}\check{z}^{-1} + \beta_{1,i}\check{z}^{-2} + \beta_{2,i}\check{z}^{-3} + \beta_{3,i}\check{z}^{-4}]}{1 + \alpha_{1,i}\check{z}^{-1} + \alpha_{2,i}\check{z}^{-2} + \alpha_{3,i}\check{z}^{-3} + \alpha_{4,i}\check{z}^{-4}} \qquad (4.26)$$

puesto que la función de transformación de banda es de segundo grado, las secciones de la función de transferencia que eran de segundo grado, han sido convertidas en secciones de cuarto grado.

Los coeficientes $\underline{\alpha}$, $\underline{\beta}$ y kpulso(i), dependen de los valores de $\underline{\alpha}$ y $\underline{\beta}$ determinados inicialmente, y de las constantes <u>v1</u> y <u>v2</u> que se las determina a partir de los valores <u>kso</u> y <u>alfaso</u> calculados en la subrutina "waps", así:

$$v1 = \frac{-2\,alfaso\,kso}{kso + 1}$$

$$v2 = \frac{kso - 1}{kso + 1}$$

entonces los coeficientes de la i-ésima sección quedan determinados por:

$$\beta_{0,i} = \frac{v1\,[2\,\beta_{0,i} - \beta_{1,i}(1+v2)]}{\beta_{0,i} - \beta_{1,i}v2}$$

$$\beta_{1,i} = \frac{(2\,v2 + v1^2)\,\beta_{0,i} - \beta_{1,i}(1 + v1^2 + v2^2)}{\beta_{0,i} - \beta_{1,i}v2}$$

$$\beta_{2,i} = \frac{v1\,[v2\,(\beta_{0,i} - \beta_{1,i}) + (\beta_{0,1}v2 - \beta_{1,i})]}{\beta_{0,i} - \beta_{1,i}v2}$$

$$\beta_{3,i} = \frac{v2\,(\beta_{0,i}v2 - \beta_{1,i})}{\beta_{0,i} - \beta_{1,i}v2}$$

$$\alpha_{1,i} = \frac{v1\,[2 - \alpha_{1,i}(1+v2) + 2\,v2\,\alpha_{2,i}]}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{2,i} = \frac{(2\,v2 + v1^2)(1 + \alpha_{2,i}) - \alpha_{1,i}(1 + v1^2 + v2^2)}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{3,i} = \frac{v1\,[2\,v2 - \alpha_{1,i}(1 + v2) + 2\,\alpha_{2,i}]}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{4,i} = \frac{v2^2 - \alpha_{1,i}v2 + \alpha_{2,i}}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$kpulso(i) = \frac{\beta_{0,i} - \beta_{1,i}v2}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

la sección del polo real que ahora es de segundo grado tiene la forma de la expresión 4.24. Los coeficientes están determinados por:

$$\beta_{0,sec+1} = v1$$

$$\beta_{1,sec+1} = v2$$

$$\alpha_{1,sec+1} = \frac{v1\,(1 - \alpha_{1,sec+1})}{1 - \alpha_{1,sec+1}\,v2}$$

$$\alpha_{2,sec+1} = \frac{v2 - \alpha_{1,sec+1}}{1 - \alpha_{1,sec+1}\,v2}$$

$$kpulso(sec+1) = \frac{\beta_{0,sec+1}}{1 - \alpha_{1,sec+1}\,v2}$$

d) Para los filtros elimina-banda

Las constantes v1 y v2 , son iguales a:

$$v1 = \frac{-2\,alfaso}{kso + 1}$$

$$v2 = \frac{1 - kso}{kso + 1}$$

los coeficientes de la i-ésima sección quedan determinados por:

$$\beta_{0,i} = \frac{v1\,[2\,\beta_{0,i} + \beta_{1,i}(1+v2)]}{\beta_{0,i} + \beta_{1,i}\,v2}$$

$$\beta_{1,i} = \frac{(2\,v2 + v1^2)\,\beta_{0,i} + \beta_{1,i}(1 + v1^2 + v2^2)}{\beta_{0,i} + \beta_{1,i}\,v2}$$

$$\beta_{2,i} = \frac{v1\,[\,v2\,(\beta_{0,i} + \beta_{1,i}) + (\beta_{0,1}\,v2 + \beta_{1,i})\,]}{\beta_{0,i} + \beta_{1,i}\,v2}$$

$$\beta_{3,i} = \frac{v2\,(\beta_{0,i}\,v2 + \beta_{1,i})}{\beta_{0,i} + \beta_{1,i}\,v2}$$

$$\alpha_{1,i} = \frac{v1\,[2 + \alpha_{1,i}\,(1+v2) + 2\,v2\,\alpha_{3,i}]}{1 + \alpha_{1,i}\,v2 + \alpha_{2,i}\,v2^2}$$

$$\alpha_{2,i} = \frac{(2\,v2 + v1^2)\,(1 + \alpha_{2,i}) + \alpha_{1,i}\,(1 + v1^2 + v2^2)}{1 + \alpha_{1,i}\,v2 + \alpha_{2,i}\,v2^2}$$

$$\alpha_{3,i} = \frac{v1\,[2\,v2 + \alpha_{1,i}\,(1 + v2) + 2\,\alpha_{2,i}]}{1 + \alpha_{1,i}\,v2 + \alpha_{2,i}\,v2^2}$$

$$\alpha_{4,i} = \frac{v2^2 + \alpha_{1,i} v2 + \alpha_{2,i}}{1 + \alpha_{1,i} v2 + \alpha_{2,i} v2^2}$$

$$kpulso(i) = \frac{\beta_{0,i} + \beta_{1,i} v2}{1 + \alpha_{1,i} v2 + \alpha_{2,i} v2^2}$$

los coeficientes de la sección del polo real están determinados por:

$$\beta_{0,sec+1} = v1$$

$$\beta_{1,sec+1} = v2$$

$$\alpha_{1,sec+1} = \frac{v1\,(1 + \alpha_{1,sec+1})}{1 + \alpha_{1,sec+1} v2}$$

$$\alpha_{2,sec+1} = \frac{v2 + \alpha_{1,sec+1}}{1 + \alpha_{1,sec+1} v2}$$

$$kpulso(sec+1) = \frac{\beta_{0,sec+1}}{1 + \alpha_{1,sec+1} v2}$$

En el programa primero se recalculan todos los valores α, luego todos los valores β y por último los coeficientes de la sección del polo real, si la hay.

En el diagrama 4.8 se indica claramente esta subrutina.

4.3.2.8 Subrutina transformacionpulso

Esta subrutina se encarga de determinar la función de transferencia del filtro digital utilizando la Invariancia de Pulso. La función a determinarse incluye la transformación de banda, por lo que demanda un tratamiento especial para cada tipo de filtro.

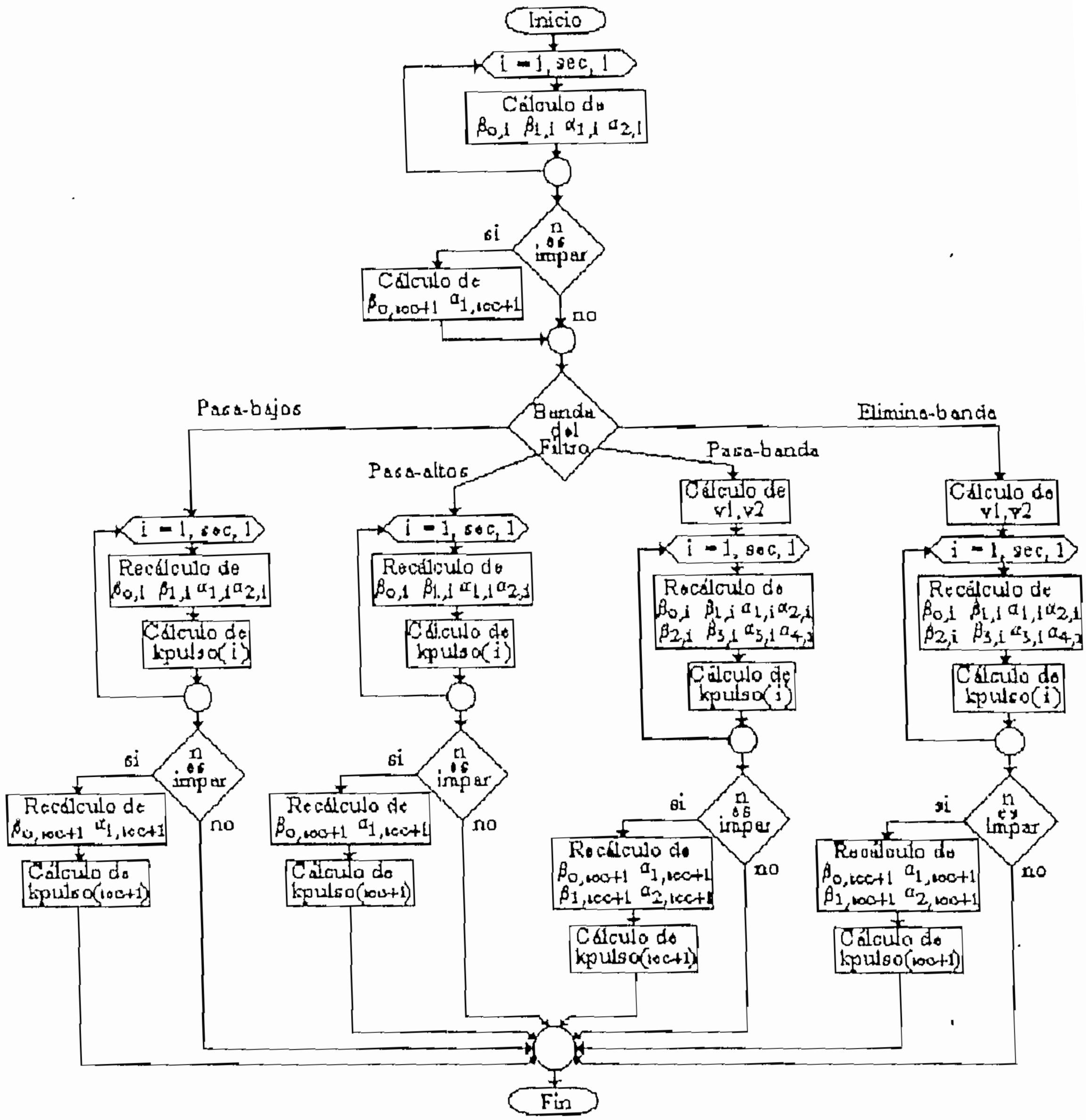


Diagrama 4.8: Subrutina "transformacionimpulso"

La digitalización del filtro por medio de la Invariancia de Pulso consiste en reemplazar los términos de la expresión 2.16 por las fracciones parciales de la función de transferencia analógica, así:

Sean dos fracciones parciales cuyos coeficientes son complejos conjugados entre sí:

$$\frac{a_{re}+ja_{im}}{s-(s_{re}+js_{im})}, \frac{a_{re}-ja_{im}}{s-(s_{re}-js_{im})}$$

entonces reemplazamos:

$$\frac{(a_{re}+ja_{im})\,z^{-1}(1-e^{s_{re}+js_{im}})}{-(s_{re}+js_{im})\,(1-e^{s_{re}+js_{im}})\,z^{-1}}$$

$$\frac{(a_{re}-ja_{im})\,z^{-1}(1-e^{s_{re}-js_{im}})}{-(s_{re}-js_{im})\,(1-e^{s_{re}-js_{im}})\,z^{-1}}$$

en este momento se cambia de la variable s a la variable z. Multiplicando estos términos obtenemos la forma de una sección de segundo grado de la función de trasferencia digital:

$$\frac{\beta_0 z^{-1}+\beta_1 z^{-2}}{1+\alpha_1 z^{-1}+\alpha_2 z^{-2}} \tag{4.27}$$

donde los coeficientes pueden ser calculados así:

$$\beta_0 = \frac{-2\,[a_{re}s_{re}+a_{im}s_{im}]\,[1-e^{s_{re}}COS\,s_{im}]+[a_{im}s_{re}-a_{re}s_{im}]\,e^{s_{re}}SEN\,s_{im}}{s_{re}^2+s_{im}^2}$$

$$\beta_1 = \frac{2\,e^{s_{re}}\,(COS\,s_{im}-e^{s_{re}})\,[a_{re}s_{re}+a_{im}s_{im}]+SEN\,s_{im}\,[a_{im}s_{re}-a_{re}s_{im}]}{s_{re}^2+s_{im}^2}$$

$$\begin{aligned}\alpha_1 &= -2\,e^{s_{re}}COS\,s_{im}\\ \alpha_2 &= e^{2s_{re}}\end{aligned} \tag{4.28}$$

para la k-ésima sección en el programa se tiene por coeficientes a: β(0,k), β(1,k), α(1,k), α(2,k) y cada uno de estos depende de los valores complejos aa(k) y s(k).

Para el caso de existir la sección de primer grado, esta tiene la forma:

$$\frac{\beta_0\, z^{-1}}{1 + \alpha_1 z^{-1}} \tag{4.29}$$

donde:

$$\beta_{0,sec+1} = \frac{-aa(sec+1)\,[1-e^{s(sec+1)}]}{s(sec+1)}$$

se entiende que los valores aa(sec+1) y s(sec+1), son reales.

De esta manera hemos determinado la función de transferencia de un filtro digital sin transformación de banda.

Seguidamente realizaremos la transformación de banda reemplazando la función $G(\check{Z}^{-1})$ encontrada en 3.2 por la variable z.

Las resultados obtenidos son:

a) Para los filtros pasa-bajos:

Este caso es exclusivo de los filtros elípticos. La forma que tienen las secciones de segundo grado es igual a 4.24, cuyos coeficientes se determinan así:

$$\beta_{0,i} = \frac{\beta_{0,i}(1+alfasola^2) - 2\,alfasola\,\beta_{1,i}}{-\beta_{0,i}alfasola + \beta_{1,i}alfasola^2}$$

$$\beta_{1,i} = \frac{-\beta_{0,i}alfasola + \beta_{1,i}}{-\beta_{0,i}alfasola + \beta_{1,i}alfasola^2}$$

$$\alpha_{1,i} = \frac{-2\,alfasola\,(1+\alpha_{2,i}) + \alpha_{1,i}(1+alfasola^2)}{1 - \alpha_{1,i}alfasola + \alpha_{2,i}alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2 - \alpha_{1,i}alfasola + \alpha_{2,i}}{1 - \alpha_{1,i}alfasola + \alpha_{2,i}alfasola^2}$$

$$kpulso(i) = \frac{-\beta_{0,i} alfasola + \beta_{1,i} alfasola^2}{1 - \alpha_{1,i} alfasola + \alpha_{2,i} alfasola^2}$$

La sección del polo real tiene la forma de 4.25 y los coeficientes se los obtiene así:

$$\beta_{0,sec+1} = \frac{-1}{alfasola}$$

$$\alpha_{1,sec+1} = \frac{\alpha_{1,sec+1} - alfasola}{1 - \alpha_{1,sec+1} alfasola}$$

$$kpulso(sec+1) = \frac{-\beta_{0,sec+1} alfasola}{1 - \alpha_{1,sec+1} alfasola}$$

cabe recalcar que los coeficientes α y β se calculan en función de los α y β de la función de transferencia sin transformación de banda.

b) Para los filtros pasa-altos:

Las secciones son de la forma 4.24 para las de segundo grado y de la forma 4.25 para la sección del polo real, igual, que los pasa-bajos. Los coeficientes se los determina de la siguiente manera:

$$\beta_{0,i} = \frac{-\beta_{0,i}(1 + alfasola^2) - 2\, alfasola\, \beta_{1,i}}{\beta_{0,i} alfasola + \beta_{1,i} alfasola^2}$$

$$\beta_{1,i} = \frac{\beta_{0,i} alfasola + \beta_{1,i}}{\beta_{0,i} alfasola + \beta_{1,i} alfasola^2}$$

$$\alpha_{1,i} = \frac{-2\, alfasola(1 + \alpha_{2,i}) - \alpha_{1,i}(1 + alfasola^2)}{1 + \alpha_{1,i} alfasola + \alpha_{2,i} alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2 + \alpha_{1,i} alfasola + \alpha_{2,i}}{1 + \alpha_{1,i} alfasola + \alpha_{2,i} alfasola^2}$$

$$kpulso(i) = \frac{\beta_{0,i} alfasola + \beta_{1,i} alfasola^2}{1 + \alpha_{1,i} alfasola + \alpha_{2,i} alfasola^3}$$

Para la sección del polo real:

$$\beta_{0,sec+1} = \frac{-1}{alfasola}$$

$$\alpha_{1,sec+1} = \frac{-\alpha_{1,sec+1} - alfasola}{1 + \alpha_{1,sec+1} alfasola}$$

$$kpulso(sec+1) = \frac{\beta_{0,sec+1} alfasola}{1+\alpha_{1,sec+1} alfasola}$$

c) Para los filtros pasa-banda:

Para este caso determinamos las constantes <u>v1</u> y <u>v2</u>:

$$v1 = \frac{-2\, alfaso(kso)}{kso + 1}$$

$$v2 = \frac{kso - 1}{kso + 1}$$

donde los valores <u>kso</u> y <u>alfaso</u> fueron determinados en la subrutina "waps".

Los coeficientes <u>ß</u> se los obtiene así:

$$\beta_{0,i} = \frac{v1\,[-\beta_{0,i}(1+v2) \ +2v2\ \beta_{1,i}]}{-\beta_{0,i} v2 + \beta_{1,i} v2^3}$$

$$\beta_{1,i} = \frac{(2\,v2 + v1^2)\,\beta_{1,i} - \beta_{0,i}(1 + v1^2 + v2^2)}{-\beta_{0,i} v2 + \beta_{1,i} v2^3}$$

$$\beta_{2,i} = \frac{-v1\,[-\beta_{0,i}(1+v2) \ + 2\,\beta_{1,i}]}{-\beta_{0,i} v2 + \beta_{1,i} v2^2}$$

$$\beta_{3,i} = \frac{-\beta_{0,i} v2 + \beta_{1,i}}{-\beta_{0,i} v2 + \beta_{1,i} v2^3}$$

los coeficientes $\underline{\alpha}$ así:

$$\alpha_{1,i} = \frac{v1[2 - \alpha_{1,i}(1+v2) + 2\,v2\,\alpha_{2,i}]}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{2,i} = \frac{(2\,v2 + v1^2)(1 + \alpha_{2,i}) - \alpha_{1,i}(1 + v1^2 + v2^2)}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{3,i} = \frac{v1\,[2\,v2 - \alpha_{1,i}(1 + v2) + 2\,\alpha_{2,i}]}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{4,i} = \frac{v2^2 - \alpha_{1,i}v2 + \alpha_{2,i}}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$kpulso(i) = \frac{-\beta_{0,i}v2 + \beta_{1,i}v2^2}{1 - \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

la sección del polo real que ahora es de segundo grado, tiene por coeficientes a:

$$\beta_{0,sec+1} = \frac{v1}{v2}$$

$$\beta_{1,sec+1} = \frac{1}{v2}$$

$$\alpha_{1,sec+1} = \frac{v1\,(1 - \alpha_{1,sec+1})}{1 - \alpha_{1,sec+1}v2}$$

$$\alpha_{2,sec+1} = \frac{v2 - \alpha_{1,sec+1}}{1 - \alpha_{1,sec+1}v2}$$

$$kpulso(sec+1) = \frac{-v2\,\beta_{0,sec+1}}{1 - \alpha_{1,sec+1}v2}$$

Las secciones normales son de cuarto grado y tienen la forma 4.26, mientras que la sección del polo real tiene la forma 4.24. La misma forma tienen las secciones de los filtros elimina-banda.

d) Para los filtros elimina-banda:

Las constantes v1 y v2 , son iguales a:

$$v1 = \frac{-2\,alfaso}{kso + 1}$$

$$v2 = \frac{1 - kso}{kso + 1}$$

los coeficientes de la i-ésima sección de cuarto grado, quedan determinados por:

$$\beta_{0,i} = \frac{v1\,[\beta_{0,i}(1+v2) + 2\,v2\,\beta_{1,i}]}{\beta_{0,i}v2 + \beta_{1,i}v2^2}$$

$$\beta_{1,i} = \frac{(2v2+v1^2)\,\beta_{1,i} + \beta_{0,i}(1+v1^2+v2^2)}{\beta_{0,i}v2 + \beta_{1,i}v2^2}$$

$$\beta_{2,i} = \frac{v1\,[(1+v2)\,\beta_{0,i} + 2\beta_{1,i}]}{\beta_{0,i}v2 + \beta_{1,i}v2^2}$$

$$\beta_{3,i} = \frac{\beta_{0,i}v2 + \beta_{1,i}}{\beta_{0,i}v2 + \beta_{1,i}v2^2}$$

$$\alpha_{1,i} = \frac{v1\,[2 + \alpha_{1,i}(1+v2) + 2\,v2\,\alpha_{2,i}]}{1 + \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{2,i} = \frac{(2\,v2 + v1^2)(1 + \alpha_{2,i}) + \alpha_{1,i}(1 + v1^2 + v2^2)}{1 + \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{3,i} = \frac{v1\,[2\,v2 + \alpha_{1,i}(1 + v2) + 2\,\alpha_{2,i}]}{1 + \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$\alpha_{4,i} = \frac{v2^2 + \alpha_{1,i}v2 + \alpha_{2,i}}{1 + \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

$$kpulso(i) = \frac{\beta_{0,i}v2 + \beta_{1,i}v2^2}{1 + \alpha_{1,i}v2 + \alpha_{2,i}v2^2}$$

Los coeficientes de la sección del polo real están determinados por:

$$\beta_{0,\,nsc+1} = \frac{v1}{v2}$$

$$\beta_{1,sec+1} = \frac{1}{v2}$$

$$\alpha_{1,sec+1} = \frac{v1\,(1 + \alpha_{1,sec+1})}{1 + \alpha_{1,sec+1}\,v2}$$

$$\alpha_{2,sec+1} = \frac{v2 + \alpha_{1,sec+1}}{1 + \alpha_{1,sec+1}\,v2}$$

$$kpulso(\mathrm{sec}+1) = \frac{v2\,\beta_{0,sec+1}}{1 + \alpha_{1,sec+1}\,v2}$$

El diagrama de flujo que corresponde a esta subrutina, es similar al diagrama 4.8, con la diferencia que, en este se considera la excepción en los filtros pasa-bajos no elípticos.

Determinados todos los coeficientes de las secciones de la función de transferencia, y sabiendo que la forma que adquieren estas, es similar a la Invariancia de Impulso, la siguiente subrutina se utiliza en común.

4.3.2.9 Subrutina transfrecuenciapulso

Esta subrutina se encarga de determinar la respuesta de frecuencia digital en base a la función de transferencia determinada en las subrutinas anteriores. Para esto reemplazamos e^{jw} por z. Entonces las variables z^{-1} y z^{-2} se reemplazan por:

z^{-1} por e^{-jw}

z^{-2} por e^{-j2w}

estos mismos reemplazos en forma trigonométrica se los haría, así:

z^{-1} por $\mathrm{COS}(w) + j\ \mathrm{SEN}(w)$

z^{-2} por $\mathrm{COS}(2w) + j\ \mathrm{SEN}(2w)$

Como se puede observar, la introducción de la frecuencia digital w, transforma a la función de transferencia en una expresión compleja. De esta manera el proceso para determinar la respuesta de frecuencia consiste en separar la parte real de la imaginaria, sección por sección, y luego extraer el módulo. Dado que, tanto el numerador como el denominador de una sección son complejos, se debe desarrollar la división de complejos.

a) Para los filtros pasa-bajos y pasa-altos, las secciones tienen la misma forma, excepto para un filtro pasa-bajos no elíptico que utiliza Invariancia de Pulso.

En general para la k-ésima sección:

Parte real del numerador :

$$a = 1 + \beta_{o.k}\,COS(w) + \beta_{1.k}\,COS(2w)$$

Parte imaginaria del numerador

$$b = -\,\beta_{o.k}\,SEN(w) - \beta_{1.k}\,SEN(2w)$$

Parte real del denominador

$$c = 1 + \alpha_{1.k}\,COS(w) + \alpha_{2.k}\,COS(2w)$$

Parte imaginaria del denominador

$$d = -\,\alpha_{1.k}\,SEN(w) - \alpha_{2.k}\,SEN(2w)$$

Por lo tanto la cada sección queda determinada así:

Parte real

$$real = kpulso(k)\,\frac{ac+bd}{c^2 + d^2}$$

Parte imaginaria

$$imag = kpulso(k)\,\frac{bc-ad}{c^2 + d^2}$$

En el programa se utilizan las variables a, b, c, d, real e imag.

En las variables real e imag se van acumulando los valores de todas las secciones por medio de un lazo FOR... NEXT.

Si el filtro analógico a partir del cual se está haciendo el diseño tiene ceros y es de orden par, entonces, se debe añadir a la parte real el valor de la constante kanaloga, así:

real = real + kanaloga

esta implicación surge de tener polinomios de igual grado en el numerador y denominador de la función de transferencia.

Si el orden es impar se debe añadir la parte de la sección de primer grado. Esta parte queda determinada así:

Parte real del numerador :

$$a = 1 + \beta_{o.k}\ COS(w)$$

Parte imaginaria del numerador

$$b = -\ \beta_{o.k}\ SEN(w)$$

Parte real de denominador

$$c = 1 + \alpha_{o.k}\ COS(w)$$

Parte imaginaria del denominador

$$d = -\alpha_{o.k}\ SEN(w)$$

el complejo resultante de la división se lo obtiene como se hizo para las secciones cuadradas.

Entonces con los valores <u>real</u> e <u>imag</u> , ya podemos determinar la respuesta de frecuencia:

$$h(i) = \sqrt{[\ real^2 + imag^2\ ]}$$

En el programa, para la Invariancia de Impulso asignamos el resultado a la variable <u>hi(i)</u>, mientras que para la Invariancia de Pulso asignamos a la variable <u>hp(i)</u>.

Como en el caso que se determinó la respuesta de frecuencia analógica, aquí también se guarda la respuesta de frecuencia en un arreglo de 304 puntos, por medio de un lazo de FOR ...NEXT de la variable i. En los cálculos, w se calcula en cada intervalo de $2\pi/304$ [rad], esto quiere decir que en 304 puntos se determinará un rango de $2\pi/304$ a 2π radianes para w.

En el caso de los filtros pasa-bajos no elípticos que utilizan la Invariancia de Pulso, puesto que no se realiza la transformación de banda, las secciones siguen siendo de la forma 4.27 y 4.29. Por lo tanto la única diferencia con lo realizado hasta aquí, es que la parte real del numerador no incluye la suma del término 1.

b) Los filtros pasa-banda y elimina-banda, tienen secciones de la idéntica forma.

Entonces para la k-ésima sección de estos:

Parte real del numerador

$$a = 1 + \beta_{o.k}\ COS(w) + \beta_{1.k}\ COS(2w) + \beta_{2.k}\ COS(3w) + \beta_{3.k}\ COS(4w)$$

Parte imaginaria del numerador

$$b = -\beta_{0.k}\ \mathrm{SEN}(w) - \beta_{1.k}\ \mathrm{SEN}(2w) - \beta_{2.k}\ \mathrm{SEN}(3w) - \beta_{3.k}\ \mathrm{SEN}(4w)$$

Parte real del denominador

$$c = 1 + \alpha_{1.k}\ \mathrm{COS}(w) + \alpha_{2.k}\ \mathrm{COS}(2w) + \alpha_{3.k}\ \mathrm{COS}(3w) + \alpha_{4.k}\ \mathrm{COS}(4w)$$

Parte imaginaria del denominador

$$d = -\alpha_{1.k}\ \mathrm{SEN}(w) - \alpha_{2.k}\ \mathrm{SEN}(2w) - \alpha_{3.k}\ \mathrm{SEN}(3w) - \alpha_{4.k}\ \mathrm{SEN}(4w)$$

La obtención de real e imag se lo hace igual que en el caso anterior.

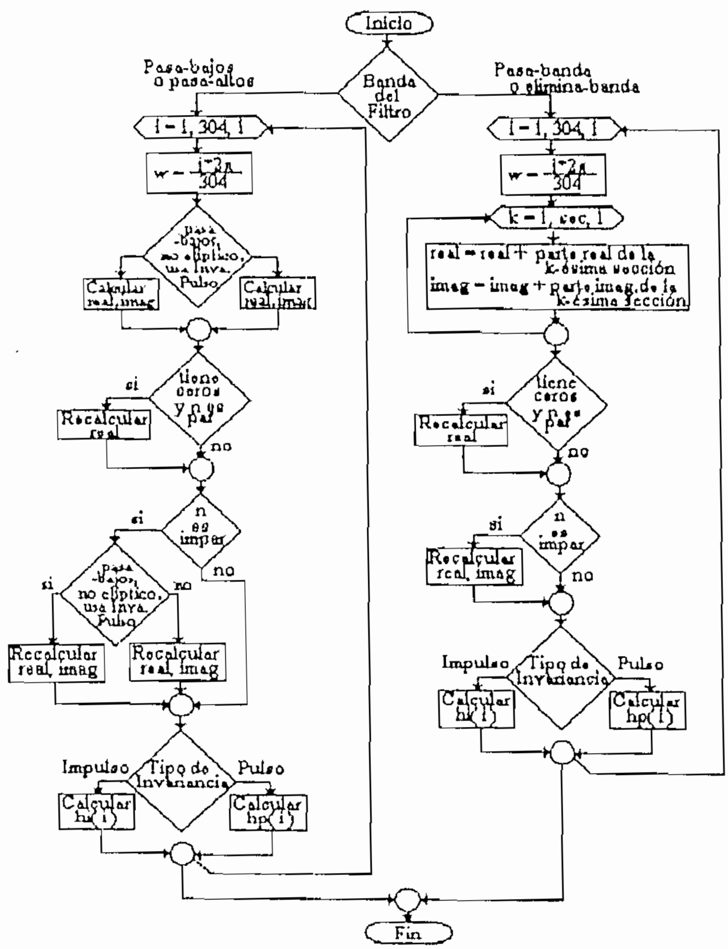


Diagrama 4.9: Subrutina "transfrecuenciapulso"

Luego se debe considerar si se aumenta o no, la variable kanaloga a la parte real . Posteriormente, si n es impar, la sección del polo real se la determina con las mismas expresiones de cálculo de las secciones de segundo grado del caso anterior. Por último asignamos el módulo del complejo:

[real] + j [imag]

a la variable correspondiente de la respuesta de frecuencia.

El diagrama de flujo de esta subrutina es el 4.9.

Con estos cálculos se termina el diseño del filtro digital utilizando Invariancia de Impulso e Invariancia de Pulso.

CASO 2: Utilizando la Transformada Bilineal

En este caso se debe tomar en cuenta que la transformación de banda puede ser: analógica-analógica ó digital-digital.

Las subrutinas desarrolladas son:

- FRECUENCIANAT
- BILINEAL
- FRECUENCIABILINEAL
- TRANSFORMACIONBILINEAL
- TRANSFRECUENCIABILINEAL

4.3.2.10 Subrutina frecuencianat

Esta subrutina se encarga de encontrar la respuesta de frecuencia analógica resultante, al efectuar la transformación de banda analógica-analógica. Esto se lo hace sin determinar la función de transferencia simplificada, sino que por facilidad de encontrar la respuesta de frecuencia se lo hace cambiando la función de transformación de banda $F(\bar{s})$ por s y luego reemplazando $\bar{s}$ por jw, directamente.

Hemos llamado función de transferencia simplificada a la función de transferencia que tiene la forma del producto de secciones con coeficientes reales.

Para esto se utilizan los resultados del apartado 3.1.

Dependiendo del tipo de banda tenemos los siguientes casos:

a) Para los filtros pasa-bajos:

Este caso se ha creado exclusivamente para los filtros elípticos, el motivo ya se lo explicó anteriormente.

Solo en este caso encontraremos una función de transferencia simplificada. Los coeficientes de esta función, dependen de los valores de los polos, los ceros y de la constante <u>kanaloga</u> propios de la función transformada en banda.

Los polos, los ceros, y la constante kanaloga, de la función transformada en banda, son el resultado de recalcular así:

Un polo se lo recalcula de la siguiente manera:

$$s(i) = s(i) \cdot u$$

un cero, como se indica a continuación:

$$c(i) = c(i) \cdot u^2$$

la constante <u>kanaloga</u> así:

$$kanaloga = kanaloga \cdot u^n$$

y en el caso de tener ceros esta contante solo se recalcula así:

$$kanaloga = kanaloga \cdot u$$

porque en el recálculo de los ceros ya se involucró el valor de u^{n-1}.

Los coeficientes (gama) de la función simplificada y la respuesta de frecuencia, se calcula de la misma manera que se hizo en la subrutina "frecuenciana", porque se trata de un filtro pasa-bajos. El motivo de no llamar en este momento a la subrutina "frecuenciana", es que, el resultado se guardaría en la variable ha(i) , y nosotros deseamos almacenar en otra variable (hat(i)), para posteriormente efectuar comparaciones con esta.

La variable hat(i) será también un arreglo de 304 puntos.

b) Para los filtros pasa-altos:

Como no existe ningún recálculo, directamente reemplazamos s por la expresión 3.2.,entonces la función de transferencia es:

$$H(\bar{s}) = \frac{kanaloga}{[u\bar{s}^{-1} - s(1)] \ldots [u\bar{s}^{-1} - s(n)]}$$

cada factor del denominador al reemplazar $\bar{s}$ por $j\omega$ quedará así:

$$u / j\omega - s(i)$$

como s(i) es un número complejo, separaremos la parte real de la imaginaria, quedándonos cada término, así:

$$-s(i)_{re} \quad - \quad j \; [\; u /\omega + s(i)_{im} \;]$$

extrayendo el módulo tenemos:

$$\sqrt{\{ \; s(i)_{re}^2 \; + \; j \; [\; u /\omega + s(i)_{im} \;]^2 \; \}}$$

por lo tanto la respuesta de frecuencia queda determinada:

$$hat(i) = kanaloga \prod_{k=1}^{soc} \frac{1}{\sqrt{s(k)_{re}^2 + [\frac{u}{\omega} + s(k)_{im}]^2}}$$

donde para cada punto de i, ω varía en $2\pi/304$ [rad/seg].

A este valor se le debe agregar un factor v, cuando la función de transferencia tiene ceros. Este factor depende de los ceros y se lo calcula con:

$$v = \prod_{k=1}^{soc} ABS[-\frac{u^2}{\omega^2} + c(k)]$$

c) Para los filtros pasa-banda:

Aquí primero debemos calcular las constantes de transformación u y l con las expresiones 4.6 y 4.7. La constante Bo ya se determinó en la subrutina "waps", mientras que a wo la calcularemos. De 4.5 tenemos:

$$wo = \varpi_{p2} \varpi_{p1}$$

reemplazando las frecuencias de paso prealabeadas:

$$wo = 4TAN(\frac{fup\pi}{fm}) TAN(\frac{fu\pi}{fm})$$

Teniendo ahora la función de transformación 3.4, reemplazamos en la función de transferencia original. Luego reemplazamos s por $j\omega$, separamos la parte real de la imaginaria y sacamos el módulo. La expresión final es igual a:

$$hat(i) = kanaloga \prod_{k=1}^{sec} \frac{1}{\sqrt{[s(k)_{im}\omega(u-l)+ul-\omega^2]^2+[s(k)_{re}\omega(u-l)]^2}}$$

donde kanaloga la recalculamos así:

$$kanaloga = kanaloga\ \omega^n (u-l)^n$$

Para los filtro que incluyen ceros, a este valor se le multiplica el factor v que se determina a continuación:

$$v = \prod_{k=1}^{sec} ABS\ [\omega^4 - (2ul + c(k)\ (u-l)^2)\,\omega^2 + ul^2)]$$

en estos casos el factor que afecta a kanaloga en el recálculo ya va incluido en el factor v . Por tanto es ahora necesario dividir aquel factor para la expresión final de hat(i), así, dependiendo de la paridad tenemos que dividir para:

n par:

$$[\omega^n\ (u-l)^n]$$

n impar:

$$[\omega^{n-1}\ (u-l)^{n-1}]$$

d) Para los filtros elimina-banda:

Calculamos las constantes de transformación u y l con las expresiones 4.6 y 4.7. La constante Bo ya se determinó en la subrutina "waps", mientras que a wo la calcularemos ahora. De 4.5 tenemos:

$$wo = \varpi_{p2}\ \varpi_{p1}$$

reemplazando las frecuencias de paso prealabeadas para este caso obtenemos:

$$wo = 4\,TAN\left(\frac{fup\pi}{fm}\right) TAN\left(\frac{fl\pi}{fm}\right)$$

Teniendo ahora el valor de u y l reemplazamos la función de transformación determinada en 3.1.4, en la función de

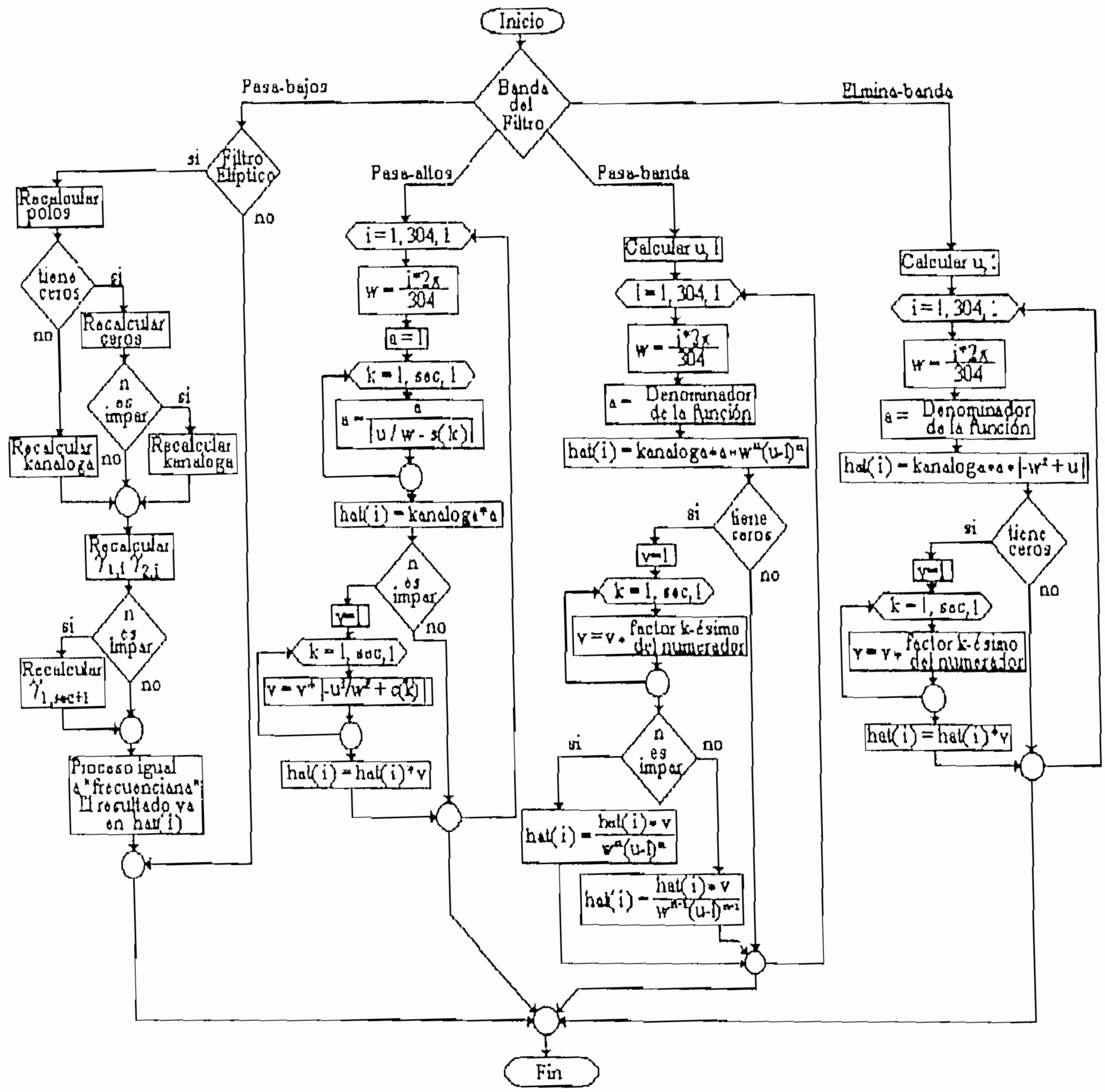


Diagrama 4.10: Subrutina "frecuencianat"

transferencia original. Luego reemplazamos $\bar{s}$ por $j\omega$, separamos la parte real de la imaginaria y sacamos el módulo. La expresión final de hat(i) es igual a:

$$hat(i) = kanaloga \prod_{k=1}^{s+c} \frac{1}{\sqrt{[s(k)_{re}(\omega^2 - ul)]^2 + [s(k)_{im}(\omega^2 - ul) + \omega(u-l)]^2}}$$

donde kanaloga la recalculamos así:

$$kanaloga = kanaloga\ ABS(-\omega^2 + ul)^n$$

Para los filtro que incluyen ceros, a este valor se lo multiplica el factor v que se determina a continuación:

$$v = ABS\left(\prod_{k=1}^{s+c} \frac{-\omega^2 (u-l)^2}{(-\omega^2 + ul)^2 + c(k)}\right)$$

En el diagrama 4.10 podremos aclarar lo dicho en esta subrutina.

4.3.2.11 Subrutina bilineal

Esta subrutina se encarga de digitalizar el filtro analógico pasa-bajos, utilizando el desarrollo de la sección 2.3.

Para esto partimos de la función de transferencia con coeficientes gama , y reemplazamos s por la expresión 2.13.
La función resultante H(z) es igual a:

$$H(z) = kbilineal \prod_{i=1}^{s+c} \frac{1 + \beta_{0,i} z^{-1} + \beta_{1,i} z^{-2}}{1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2}}$$

Los coeficientes ß para la i-ésima sección son iguales a:

$$\beta_{0,i} = 2$$
$$\beta_{1,i} = 1$$

y si la función original tenía ceros el coeficiente $\beta_{0,i}$ es distinto:

$$\beta_{0,i} = \frac{-8 + 2c(i)}{4 + c(i)}$$

Los coeficientes $\underline{\alpha}$ para la i-ésima sección son iguales a:

$$\alpha_{1,i} = \frac{-8 + 2\gamma_{2,i}}{4 + 2\gamma_{1,i} + \gamma_{2,i}}$$

$$\alpha_{2,i} = \frac{4 - 2\gamma_{1,i} + \gamma_{2,i}}{4 + 2\gamma_{1,i} + \gamma_{2,i}}$$

Si se tiene la sección de primer grado, ésta es igual a:

$$\frac{1 + z^{-1}}{1 + \alpha_{1,sec+1} z^{-1}}$$

donde:

$$\alpha_{1,sec+1} = \frac{\gamma_{2,sec+1} - 2}{2 + \gamma_{2,sec+1}}$$

La constante kbilineal es igual a:

$$kbilineal = \frac{kanaloga}{\prod_{i=1}^{sec} [4 + 2\gamma_{1,i} + \gamma_{2,i}]}$$

si existe la sección de primer grado se recalcula así:

$$kbilineal = kbilineal \,/\, \gamma_{2,sec+1}$$

y si la función de transferencia original tenía ceros se recalcula así:

$$kbilineal = kbilineal \prod_{i=1}^{sec} [4 + c(i)]$$

De esta manera hemos determinado la función de transferencia del filtro pasa-bajos digital.

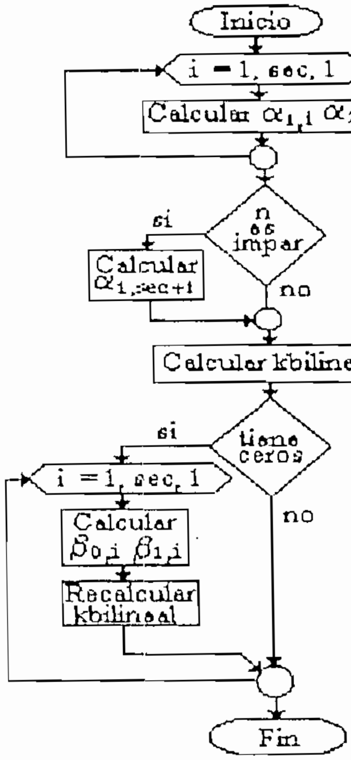


Diagrama 4.11: Subrutina "bilineal"

Esta subrutina es utilizada cuando realizamos la transformación de banda digital-digital, y cuando digitalizamos un filtro pasa-bajos transformado en banda con el método de transformación analógica-analógica.

Veamos el diagrama de flujo 4.11, para aclarar lo dicho de esta subrutina.

4.3.2.12 Subrutina frecuenciabilineal

Esta subrutina se encarga de obtener la respuesta de frecuencia digital de los filtros pasa-bajos o pasa-altos, a partir de los resultados encontrados en la subrutina "bilineal".

Para esto debemos reemplazar z^{-1} por COS(w)-jSEN(w) en las funciones de transferencia digitales.

Antes habíamos determinado que para el filtro pasa-bajos, el numerador de una sección de la función de transferencia digital es igual a:

$$1 + 2\ z^{-1} + z^{-2}$$

factorando tendremos:

$$(\ 1 + z^{-1}\ )^2$$

esto nos lleva a concluir que al multiplicar los numeradores de todas las secciones, inclusive el de la sección de primer grado, obtendremos como resultado en el numerador lo siguiente:

$$(\ 1 + z^{-1}\ )^n$$

Posteriormente veremos que para un filtro pasa-altos esta expresión es igual a:

$$(\ 1 - z^{-1}\ )^n$$

Al reemplazar el valor de z^{-1}, sacar el módulo de ese resultado, y multiplicar la constante <u>kbilineal</u>, tendremos en el numerador lo siguiente:

Para el filtro pasa-bajos:

$$kbilineal\ [\ \sqrt{[1+COS(w)]^2 + SEN(w)^2}\ ]^n$$

para el filtro pasa-altos:

$$kbilineal\ [\ \sqrt{[1-COS(w)]^2 + SEN(w)^2}\ ]^n$$

Ahora nos quedaría determinar el denominador, éste se lo calcula multiplicando el módulo de los denominadores de todas las secciones. El módulo del denominador de la k-ésima sección es igual a:

$$\sqrt{[1+\alpha_{1,k}COS(w)+\alpha_{2,k}COS(2w)]^2 + [\alpha_{1,k}SEN(w)+\alpha_{2,k}SEN(2w)]^2}$$

Cuando la función original ha tenido ceros, el numerador de las secciones se lo determina así:

$$\sqrt{[1+\beta_{0,k}COS(w)+\beta_{1,k}COS(2w)]^2 + [\beta_{0,k}SEN(w)+\beta_{1,k}SEN(2w)]^2}$$

para encontrar el numerador total multiplicaremos los numeradores de todas las secciones mas la constante kbilineal.

En el caso de ser n impar se incluirá la sección de primer grado que será igual a:

Para los filtros pasa-bajos:

$$\frac{1 + z^{-1}}{1 + \alpha_{1,sec+1} z^{-1}}$$

Para los filtros pasa-altos:

$$\frac{1 - z^{-1}}{1 + \alpha_{1,sec+1} z^{-1}}$$

Como resultado tendremos una función que depende de la variable ω, y la respuesta de frecuencia consiste en evaluar esta función para un rango determinado. El valor que adquiere la función al evaluarla en un punto se guarda en la variable <u>hb(i)</u>. Como en los casos anteriores se evalúa la respuesta de frecuencia cada intervalo de $2\pi/304$.

En el diagrama 4.12 se indica con mayor claridad esta subrutina.

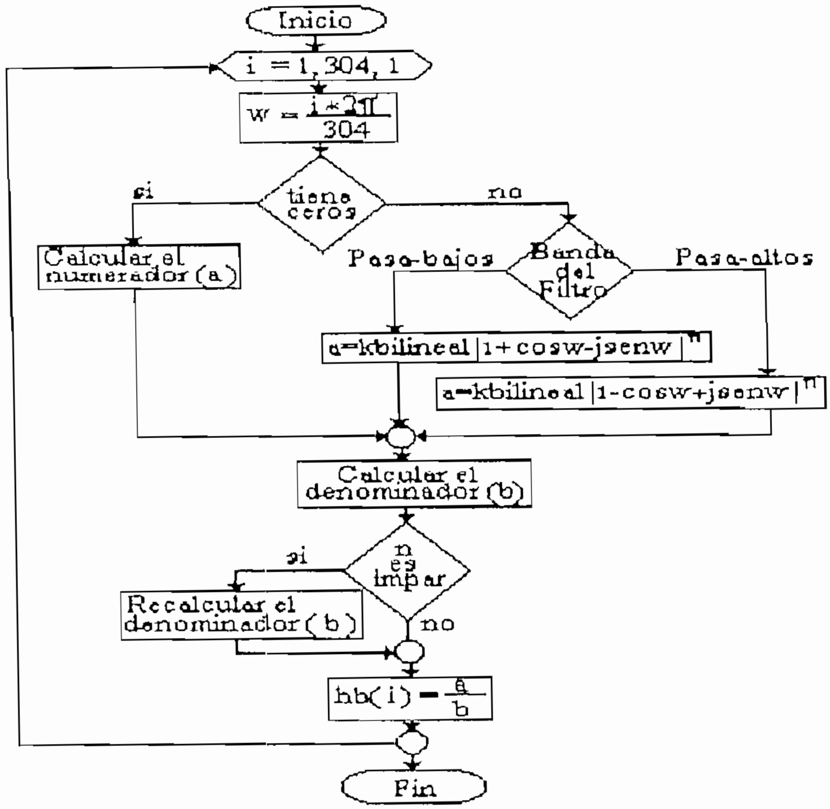


Diagrama 4.12: Subrutina "frecuenciabilineal"

4.3.2.13 Subrutina transformacionbilineal

Esta subrutina realiza el cálculo de la función de transferencia del filtro digital con transformación de banda.

Para esto se debe tener presente que la transformación de la banda puede hacerse por el método analógico-analógico o por el método digital-digital.

Hacemos un SELECT CASE de la variable trans, y tenemos:

CASO 1: Con transformación analógica-analógica.

En esta parte se determinarán las funciones de transferencia de los filtros pasa-altos, pasa-banda y elimina-banda. El desarrollo implica, primero el reemplazo de la variable s por la función de transformación de banda adecuada y luego el cambio de variable de $\bar{s}$ por z. Los resultados que se obtienen son los siguientes:

Para el filtro pasa-altos:

- Cuando la función inicial no tiene ceros:

$$H(z) = kbilineal \prod_{i=1}^{sec} \frac{1 - 2z^{-1} + z^{-2}}{1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2}}$$

- Cuando la función inicial tiene ceros:

$$H(z) = kbilineal \prod_{i=1}^{sec} \frac{1 + \beta_{0,i} z^{-1} + \beta_{1,i} z^{-2}}{1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2}}$$

Para los filtros pasa-banda cuya función analógica no tiene ceros:

$$H(z) = kbilineal \frac{[1 - z^{-2}]^{2sec}}{\prod_{i=1}^{sec} [1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2} + \alpha_{3,i} z^{-3} + \alpha_{4,i} z^{-4}]}$$

Para los filtros elimina-banda cuya función analógica no tiene ceros:

$$H(z) = kbilineal \frac{1 + \beta_{0,1} z^{-1} + \beta_{1,1} z^{-2} + \beta_{2,1} z^{-3} + \beta_{3,1} z^{-4}}{\prod_{i=1}^{sec} [1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2} + \alpha_{3,i} z^{-3} + \alpha_{4,i} z^{-4}]}$$

Para los filtros pasa-banda, y para los filtros elimina-banda cuya función original analógica tiene ceros:

$$H(z) = kbilineal \prod_{i=1}^{sec} \frac{1 + \beta_{0,i} z^{-1} + \beta_{1,i} z^{-2} + \beta_{2,i} z^{-3} + \beta_{3,i} z^{-4}}{1 + \alpha_{1,i} z^{-1} + \alpha_{2,i} z^{-2} + \alpha_{3,i} z^{-3} + \alpha_{4,i} z^{-4}}$$

A estas funciones se les debe agregar las secciones del polo real, que dependiendo del filtro son iguales a:

Para el filtro pasa-altos:

$$\frac{1 - z^{-1}}{1 + \alpha_{1,sec+1} z^{-1}}$$

Para el filtro pasa-banda :

$$\frac{1 - z^{-2}}{1 + \alpha_{1,sec+1} z^{-1} + \alpha_{2,sec+1} z^{-2}}$$

Para el filtro elimina-banda :

$$\frac{1 + \beta_{0,sec+1} z^{-1} + z^{-2}}{1 + \alpha_{1,sec+1} z^{-1} + \alpha_{2,sec+1} z^{-2}}$$

Ahora queda por indicar como se obtienen los coeficientes $\underline{\alpha}$ y $\underline{\beta}$, y la constante <u>kbilineal</u> .

a) Para los filtros pasa-altos:

$$\alpha_{1,i} = \frac{2u^2 - 8\,[s(i)_{re}^2 + s(i)_{im}]}{u^3 - 4u\,[s(i)_{re}] + 4\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

$$\alpha_{2,i} = \frac{u^2 + 4u[s(i)_{re}] + 4\,[s(i)_{re}^2 + s(i)_{im}^2]}{u^2 - 4u\,[s(i)_{re}] + 4\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

$$\alpha_{1,sec+1} = \frac{u + 2\,[s(sec+1)_{re}]}{u - 2\,[s(sec+1)_{re}]}$$

Para el caso en que existan ceros:

$$\beta_{0,i} = \frac{2u^2 - 8c(i)}{u^2 + 4c(i)}$$

$$\beta_{1,i} = 1$$

La constante kbilineal es igual a:

$$kbilineal = \frac{kanaloga\,2^n}{\prod_{i=1}^{sec} (u^2 - 4u\, s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}])}$$

si n es impar se recalcula este valor así:

kbilineal = kbilineal / [u - 2 s(sec+1)$_{re}$]

cuando la función tiene ceros, se recalcula así:

para n par:

$$kbilineal = kbilineal\,\frac{\prod_{i=1}^{sec} [u^2 + 4c(i)]}{2^n}$$

para n impar:

$$kbilineal = kbilineal\,\frac{\prod_{i=1}^{sec} [u^2 + 4c(i)]}{2^{n-1}}$$

En el programa, la constante <u>kbilineal</u> no se la calcula independientemente, sino que se incluye en los lazos del cálculo de α y β .

Para finalizar esta parte se llama a la subrutina "frecuenciabilineal", para calcular la respuesta de frecuencia.

b) Para los filtros pasa-banda:

Determinadas las constantes u y l en la subrutina "frecuencianat" podemos directamente hacer uso de ellas. Los coeficientes $\underline{\alpha}$ son iguales a:

$$\alpha_{1,i} = \frac{(-8+2ul)\,[(4+ul)\,2-4\,(u-l)\,s(i)_{re}]}{(4+ul)^2 - (4+ul)\,(u-l)\,4\,s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}]\,(u-l)^2}$$

$$\alpha_{2,i} = \frac{(4+ul)^2 - 8\,(u-l)^2\,[s(i)^2_{re}+s(i)^2_{im}] + (4+ul)^2 + (-8+2ul)^2}{(4+ul)^2 - (4+ul)\,(u-l)\,4\,s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}]\,(u-l)^2}$$

$$\alpha_{3,i} = \frac{(-8+2ul)\,[(4+ul)\,2 + 4\,(u-l)\,s(i)_{re}]}{(4+ul)^2 - (4+ul)\,(u-l)\,4\,s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}]\,(u-l)^2}$$

$$\alpha_{4,i} = \frac{(4+ul)^2 + (4+ul)\,(u-l)\,4\,s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}]\,(u-l)^2}{(4+ul)^2 - (4+ul)\,(u-l)\,4\,s(i)_{re} + 4\,[s(i)^2_{re} + s(i)^2_{im}]\,(u-l)^2}$$

Si n es impar:

$$alfa_{1,sec+1} = \frac{-8+2ul}{4-2\,[s(sec+1)_{re}]\,(u-l) + ul}$$

$$alfa_{2,sec+1} = \frac{4+2\,[s(sec+1)_{re}]\,(u-l) + ul}{4-2\,[s(sec+1)_{re}]\,(u-l) + ul}$$

Si la función tiene ceros, entonces:

$$\beta_{0,i} = \frac{-64 + 4\,(ul)^2}{16 + 4\,[2ul+c(i)\,(u-l)^2] + (ul)^2}$$

$$\beta_{1,i} = \frac{96 - 8\,[2ul+c(i)\,(u-l)^2] + 6\,(ul)^2}{16 + 4\,[2ul+c(i)\,(u-l)^2] + (ul)^2}$$

$$\beta_{2,i} = \beta_{0,i}$$

$$\beta_{3,i} = 1$$

La constante kbilineal se la determina así:

$$kbilineal = \frac{kanaloga\,[2\,(u-l)\,]^{2}}{\prod_{i=1}^{sec}\left[(4+ul)^{2}-4\,(4+ul)\,(u-l)\,s(i)_{re}+4\,[s(i)_{re}^{2}+s(i)_{im}^{2}]\,(u-l)^{2}\right]}$$

si n es impar se debe recalcular de la siguiente manera:

$$kbilineal = \frac{kbilineal}{4\,-\,2\,s(\sec+1)_{re}(u-l)\,+\,ul}$$

si tiene ceros la función se recalcula, así:

para n par:

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}[16+4\,[2ul+c(i)\,(u-l)^{2}]+(ul)^{2}]}{[2\,(u-l)\,]^{n}}$$

para n impar:

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}[16+4\,[2ul+c(i)\,(u-l)^{2}]+(ul)^{2}]}{[2\,(u-l)\,]^{n-1}}$$

c) Para los filtros elimina-banda:

Determinadas las constantes u y l en la subrutina "frecuencianat" podemos directamente hacer uso de ellas. Los coeficientes α son iguales a:

$$\alpha_{1,i} = \frac{-4\,[16+(ul)^2]\,[s(i)_{re}^2+s(i)_{im}^2] + 8\,s(i)_{re}(u-l)\,(4-ul)}{4\,(u-l)^2-4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

$$\alpha_{2,i} = \frac{-8\,(u-l)^2 + [2\,(4+ul)^2 + 4\,(4-ul)^2]\,[s(i)_{re}^2+s(i)_{im}^2]}{4\,(u-l)^2-4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

$$\alpha_{3,i} = \frac{-4\,[16+(ul)^2]\,[s(i)_{re}^2+s(i)_{im}^2] - 8\,s(i)_{re}(u-l)\,(4-ul)}{4\,(u-l)^2-4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

$$\alpha_{4,i} = \frac{4\,(u-l)^2 + 4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]}{4\,(u-l)^2 - 4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]}$$

si n es impar:

$$alfa_{1,sec+1} = \frac{2\,s(\mathrm{sec}+1)_{re}(4-ul)}{-s(\mathrm{sec}+1)_{re}(4+ul) + 2\,(u-l)}$$

$$alfa_{2,sec+1} = \frac{-s(\mathrm{sec}+1)_{re}(4+ul) - 2\,(u-l)}{-s(\mathrm{sec}+1)_{re}(4+ul) + 2\,(u-l)}$$

Los coeficientes ß son iguales a:

si la función original no tiene ceros:

$$\beta_{0,1} = \frac{-64 + 4\,(ul)^2}{(4+ul)^2}$$

$$\beta_{1,1} = \frac{6\,[16+(ul)^2] - 16\,ul}{(4+ul)^2}$$

si la función tiene ceros, entonces:

$$\beta_{0,i} = \frac{-4\,c(i)\,[16 - (ul)^2}{4\,[(u-l)^2+2\,c(i)\,ul] + c(i)\,[16+(ul)^2]}$$

$$\beta_{1,i} = \frac{-8\,[2ul\,c(i) + (u-l)^2] + 6\,c(i)\,[16+(ul)^2}{4\,[(u-l)^2+2\,c(i)\,ul] + c(i)\,[16+(ul)^2]}$$

y en ambos casos:

$$\beta_{2,i} = \beta_{0,i}$$
$$\beta_{3,i} = 1$$

si n es impar:

$$\beta_{0,sec+1} = \frac{-8 + 2ul}{4+ul}$$

La constante <u>kbilineal</u> se la determina así:

$$\alpha_{1,i} = \frac{kanaloga\,(4+ul)^{2sec}}{\prod_{i=1}^{sec} [4\,(u-l)^2 - 4\,(u-l)\,(4+ul)\,s(i)_{re} + (4+ul)^2\,[s(i)_{re}^2 + s(i)_{im}^2]]}$$

si n es impar se debe recalcular de la siguiente manera:

$$kbilineal = \frac{kbilineal\,(4+ul)}{-s(sec+1)_{re}(4+ul) + 2\,(u-l)}$$

si tiene ceros la función se recalcula, así:

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec} [c(i)\,[16+(ul)^2] + 4\,[(u-l)^2 + 2c(i)\,ul]}{(4+ul)^{2sec}}$$

CASO 2: Con transformación digital-digital.

En esta parte se utilizarán los resultados de la subrutina "bilineal", por lo tanto previamente se debe invocarla.

Estos resultados son los coeficientes de la función de transferencia digital del filtro pasa-bajos. Lo que se realiza propiamente en esta parte, es la transformación de banda de un filtro pasa-bajos ya digitalizado.

Para esto reemplazaremos la variable z por la función $G(\check{z}^{-1})$ encontrada en el apartado 3.2. Las funciones de transferencia así obtenidas tienen la misma forma que se encontró en los filtros digitales utilizando la transformación analógica-analógica.

Los coeficientes α y ß, y la constante kbilineal que se determinaron en "bilineal", son recalculados. Por eso en todas las expresiones que se verán a continuación, los términos α , ß y kbilineal que están en el miembro derecho, pertenecen a los calculados en "bilineal". Las constantes alfasola , alfaso y kso se determinaron en la subrutina "waps".

a) Para los filtros pasa-bajos:

Esta parte es exclusiva de los filtros elípticos, por lo dicho anteriormente.

$$\alpha_{1,i} = \frac{-2\,alfasola + \alpha_{1,i}(1 + alfasola^2) - 2\,alfasola\,\alpha_{2,i}}{1 - \alpha_{1,i}\,alfasola + \alpha_{2,i}\,alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2 - \alpha_{1,i}\,alfasola + \alpha_{2,i}}{1 - \alpha_{1,i}\,alfasola + \alpha_{2,i}\,alfasola^2}$$

Si existe la sección del polo real, entonces:

$$\alpha_{1,sec+1} = \frac{\alpha_{1,sec+1} - alfasola}{1 - \alpha_{1,sec+1}\,alfasola}$$

Cuando la función inicial tiene ceros se recalculan los ß:

$$\beta_{0,i} = \frac{\beta_{0,i}(1+alfasola^2) - 2alfasola\,(1 + \beta_{1,i})}{1 - \beta_{0,i}\,alfasola + \beta_{1,i}\,alfasola^2}$$

$$\beta_{1,i} = \frac{alfasola^2 - \beta_{0,i}alfasola + \beta_{1,i}}{1 - \beta_{0,i}alfasola + \beta_{1,i}alfasola^2}$$

Se recalcula la kbilineal, entonces:

$$kbilineal = \frac{kbilineal\,(1-alfasola)^n}{\prod_{i=1}^{sec}[1-\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2]}$$

si n es impar se vuelve a recalcular:

$$kbilineal = \frac{kbilineal}{1-\alpha_{1,sec+1}alfasola}$$

si tiene ceros se recalcula, así:

si n es par

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}[1-\beta_{0,i}alfasola+\beta_{1,i}alfasola^2]}{(1 - alfasola)^n}$$

si n es impar

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}[1-\beta_{0,i}alfasola+\beta_{1,i}alfasola^2]}{(1 - alfasola)^{n-1}}$$

Al final de estos cálculos se llama a la subrutina "frecuenciabilineal" para calcular la respuesta de frecuencia.

b) Para los filtros pasa-altos:

$$\alpha_{1,i} = \frac{-2\,alfasola-\alpha_{1,i}(1+alfasola^2)-2\,alfasola\,\alpha_{2,i}}{1+\alpha_{1,i}alfasola+\alpha_{2,i}alfasola^2}$$

$$\alpha_{2,i} = \frac{alfasola^2 + \alpha_{1,i} alfasola + \alpha_{2,i}}{1 + \alpha_{1,i} alfasola + \alpha_{2,i} alfasola^2}$$

Si existe la sección del polo real, entonces:

$$\alpha_{1,sec+1} = \frac{-\alpha_{1,sec+1} - alfasola}{1 + \alpha_{1,sec+1} alfasola}$$

Cuando la función inicial tiene ceros se recalculan los ß:

$$\beta_{0,i} = \frac{-\beta_{0,i}(1+alfasola^2) - 2alfasola(1 + \beta_{1,i})}{1 + \beta_{0,i} alfasola + \beta_{1,i} alfasola^2}$$

$$\beta_{1,i} = \frac{alfasola^2 + \beta_{0,i} alfasola + \beta_{1,i}}{1 + \beta_{0,i} alfasola + \beta_{1,i} alfasola^2}$$

Se recalcula la kbilineal, entonces:

$$kbilineal = \frac{kbilineal(1+alfasola)^n}{\prod_{i=1}^{sec} [1+\alpha_{1,i} alfasola+\alpha_{2,i} alfasola^2]}$$

si n es impar se vuelve a recalcular:

$$kbilineal = \frac{kbilineal}{1 + \alpha_{1,sec+1} alfasola}$$

si tiene ceros se recalcula, así:

si n es par

$$kbilineal = kbilineal \frac{\prod_{i=1}^{sec} [1 + \beta_{0,i} alfasola + \beta_{1,i} alfasola^2]}{(1 + alfasola)^n}$$

si n es impar

$$kbilineal = kbilineal \frac{\prod_{i=1}^{soc}[1+\beta_{0,i}alfasola+\beta_{1,i}alfasola^2]}{(1+alfasola)^{n-1}}$$

Al final de estos cálculos se llama a la subrutina "frecuenciabilineal" para calcular la respuesta de frecuencia.

c) Para los filtros pasa-banda:

Primero debemos calcular las constantes <u>v1</u> y <u>v2</u> , que intervendrán en las expresiones, entonces:

$$v1 = \frac{-2\,alfaso\,(kso)}{kso+1}$$

$$v2 = \frac{kso-1}{kso+1}$$

Los coeficientes <u>α</u> se los determina así:

$$\alpha_{1,i} = \frac{2v1-\alpha_{1,i}v1(1+v2)+2\,v1\,v2\,\alpha_{2,i}}{1-\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{2,i} = \frac{(2v2+v1^2)(1+\alpha_{2,i})-\alpha_{1,i}(1+v1^2+v2^2)}{1-\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{3,i} = \frac{2v1v2-\alpha_{1,i}v1(1+v2)+2\,v1\,\alpha_{2,i}}{1-\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{4,i} = \frac{v2^2-\alpha_{1,i}v2+\alpha_{2,i}}{1-\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

Si existe la sección del polo real, entonces:

$$\alpha_{1,soc+1} = \frac{v1\,(1-\alpha_{1,soc+1})}{1-\alpha_{1,soc+1}v2}$$

$$\alpha_{2,sec+1} = \frac{v2-\alpha_{1,sec+1}}{1-\alpha_{1,sec+1}v2}$$

Cuando la función inicial tiene ceros se recalculan los ß:

$$\beta_{0,i} = \frac{2v1-\beta_{0,i}v1(1+v2)+2v1v2\beta_{1,i}}{1-\beta_{0,i}v2+\beta_{1,i}v2^2}$$

$$\beta_{1,i} = \frac{(2v2+v1^2)(1+\beta_{1,i})-\beta_{0,i}(1+v1^2+v2^2)}{1-\beta_{0,i}v2+\beta_{1,i}v2^2}$$

$$\beta_{2,i} = \frac{2v1v2-\beta_{0,i}v1(1+v2)+2v1\beta_{1,i}}{1-\beta_{0,i}v2+\beta_{1,i}v2^2}$$

$$\beta_{3,i} = \frac{v2^2-\beta_{0,i}v2+\beta_{1,i}}{1-\beta_{0,i}v2+\beta_{1,i}v2^2}$$

Se recalcula la <u>kbilineal</u>, entonces:

$$kbilineal = \frac{kbilineal(1-v2)^n}{\prod_{i=1}^{sec}[1-\alpha_{1,i}v2+\alpha_{2,i}v2^2]}$$

si n es impar se vuelve a recalcular:

$$kbilineal = \frac{kbilineal}{1-\alpha_{1,sec+1}v2}$$

si tiene ceros se recalcula, así:

si n es par

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}[1-\beta_{0,i}v2+\beta_{1,i}v2^2]}{(1-v2)^n}$$

si n es impar

$$kbilineal = kbilineal\frac{\prod_{i=1}^{soc}[1-\beta_{0,i}v2+\beta_{1,i}v2^2]}{(1-v2)^{n-1}}$$

d) Para los filtros elimina-banda:

Primero debemos calcular las constantes v1 y v2 , que intervendrán en las expresiones, entonces:

$$v1 = \frac{-2\,alfaso}{kso+1}$$

$$v2 = \frac{1-kso}{kso+1}$$

Los coeficientes α se los determina así:

$$\alpha_{1,i} = \frac{2v1+\alpha_{1,i}v1(1+v2)+2v1v2\alpha_{2,i}}{1+\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{2,i} = \frac{(2v2+v1^2)(1+\alpha_{2,i})+\alpha_{1,i}(1+v1^2+v2^2)}{1+\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{3,i} = \frac{2v1v2+\alpha_{1,i}v1(1+v2)+2v1\alpha_{2,i}}{1+\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

$$\alpha_{4,i} = \frac{v2^2+\alpha_{1,i}v2+\alpha_{2,i}}{1+\alpha_{1,i}v2+\alpha_{2,i}v2^2}$$

Recalculando los ß, pero para este tipo de filtro depende si la función original tiene ceros o no:

No tiene ceros

$$\beta_{0,1} = \frac{4v1}{1+v2}$$

$$\beta_{1,1} = \frac{4v1^2+2(1+v2)^2}{(1+v2)^2}$$

$$\beta_{2,1} = \beta_{0,1}$$

$$\beta_{3,1} = 1$$

si tiene ceros

$$\beta_{0,i} = \frac{2v1+\beta_{0,i}v1(1+v2)+2v1v2\beta_{1,i}}{1+\beta_{0,i}v2+\beta_{1,i}v2^2}$$

$$\beta_{1,i} = \frac{(2v2+v1^2)(1+\beta_{1,i})+\beta_{0,i}(1+v1^2+v2^2)}{1+\beta_{0,i}v2+\beta_{1,i}v2^2}$$

$$\beta_{2,i} = \frac{2v1v2+\beta_{0,i}v1(1+v2)+2v1\beta_{1,i}}{1+\beta_{0,i}v2+\beta_{1,i}v2^2}$$

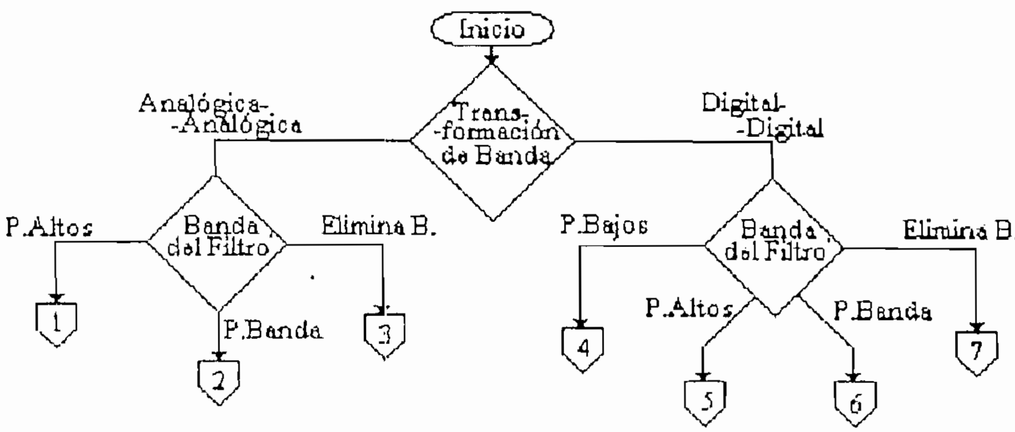
$$\beta_{3,i} = \frac{v2^2+\beta_{0,i}v2+\beta_{1,i}}{1+\beta_{0,i}v2+\beta_{1,i}v2^2}$$

Si existe la sección del polo real, entonces:

$$\alpha_{1,sec+1} = \frac{v1(1+\alpha_{1,sec+1})}{1+\alpha_{1,sec+1}v2}$$

$$\alpha_{2,sec+1} = \frac{v2+\alpha_{1,sec+1}}{1+\alpha_{1,sec+1}v2}$$

$$\beta_{0,sec+1} = \frac{2v1}{1+v2}$$



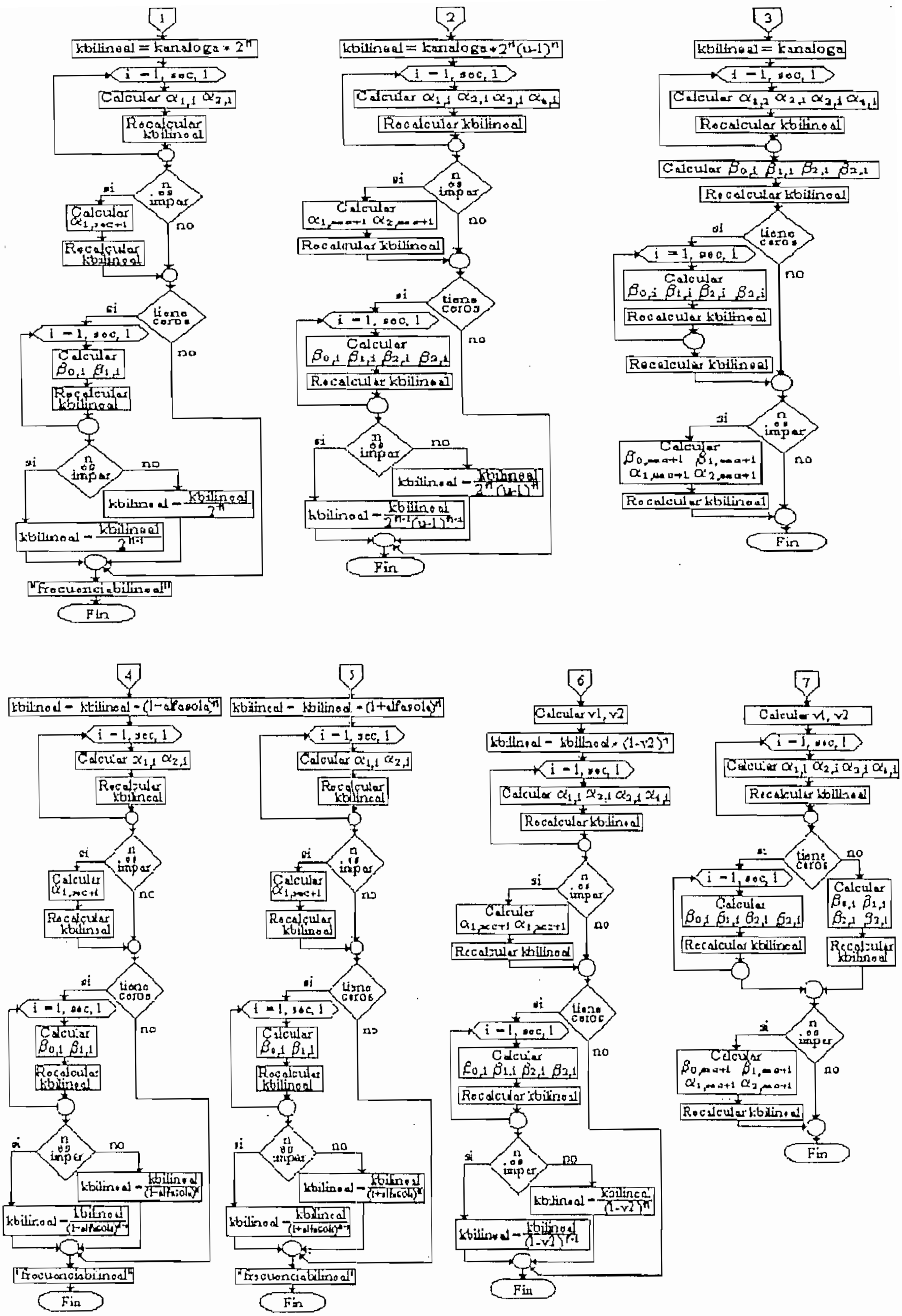


Diagrama 4.13: Subrutina "transformacionbilineal"

Se recalcula la kbilineal, entonces:

$$kbilineal = \frac{kbilineal\,(1+v2)^{2sec}}{\prod_{i=1}^{sec}\left[1+\alpha_{1,i}v2+\alpha_{2,i}v2^2\right]}$$

en cambio si tiene ceros se recalcula, así:

$$kbilineal = kbilineal\frac{\prod_{i=1}^{sec}\left[1+\beta_{0,i}v2+\beta_{1,i}v2^2\right]}{\prod_{i=1}^{sec}\left[1+\alpha_{1,i}v2+\alpha_{2,i}v2^2\right]}$$

si n es impar se vuelve a recalcular:

$$kbilineal = \frac{kbilineal\,(1+v2)}{1+\alpha_{1,sec+1}v2}$$

Para aclarar lo dicho de esta subrutina veamos el diagrama de flujo 4.13.

4.3.2.14 Subrutina transfrecuenciabilineal

Esta subrutina se encarga de calcular la respuesta de frecuencia digital de los filtros pasa-banda y elimina-banda, es decir hace lo mismo que hace la "frecuenciabilineal" con los filtros pasa-bajos y pasa-altos.

Esta subrutina no diferencia si la función de transferencia digital viene de una transformación analógica-analógica o de una transformación digital-digital, debido a que la forma de las funciones son iguáles. Cabe recalcar que los valores de los coeficientes son los diferentes.

De esta manera, para obtener la respuesta de frecuencia debemos reemplazar z^{-1} o $\check{z}^{-1}$ por [COS(w) -j SEN(w)] en la función de transferencia encontrada en "transformacionbilineal".

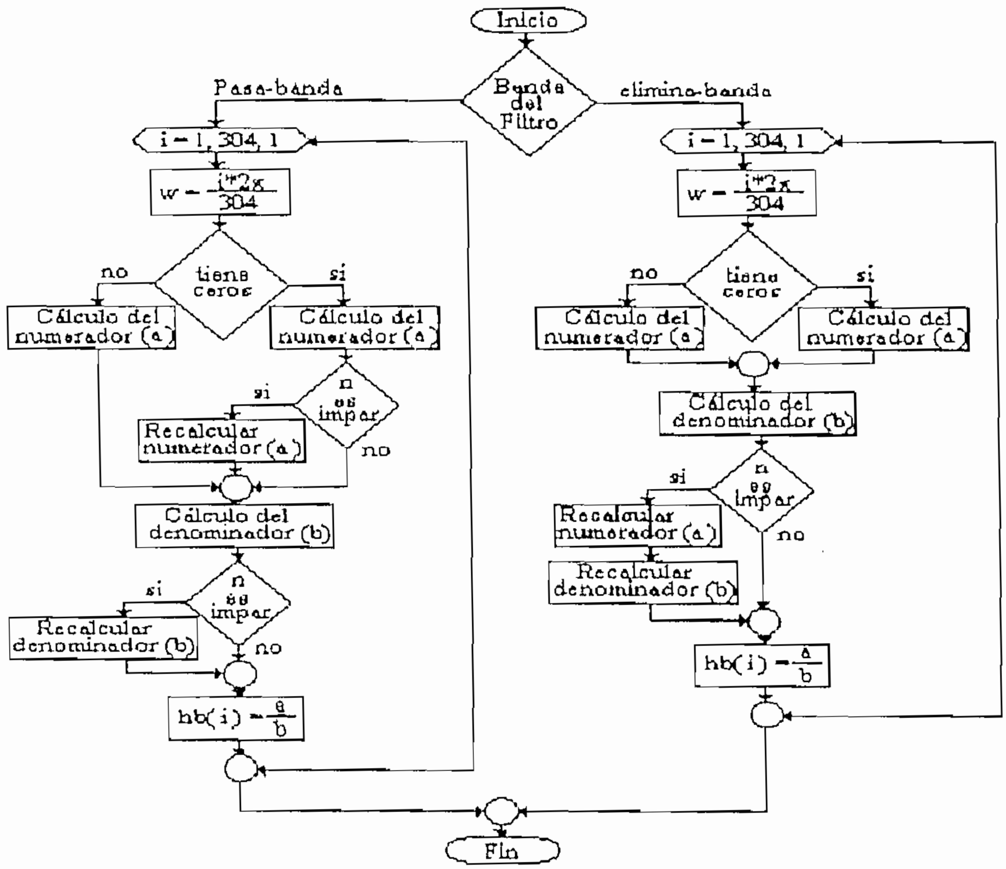


Diagrama 4.14: Subrutina "transfrecuenciabilineal"

Lo que aquí se hace es similar a lo realizado en otras subrutinas como la "transfrecuenciapulso", con la característica de que aquí no se suman las módulos de las secciones, sino que se multiplican.

Sería inútil describir el proceso, por lo que ya se ha dicho, entonces nos detendremos a observar el diagrama 4.14, donde se ilustra esta subrutina.

Para finalizar con el análisis de las subrutinas de cálculo, nos queda por revisar las siguientes subrutinas:

- SALIDATOTAL
- SALIDABILINEAL
- SALIDAPULSO

4.3.2.15 Subrutina salidatotal

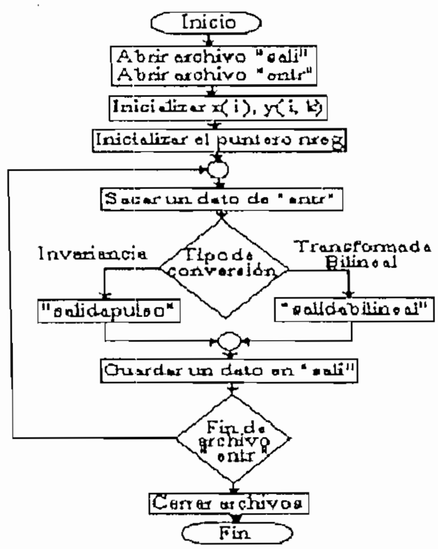
Esta subrutina se encarga de filtrar la señal de entrada. El proceso consiste en tomar un punto de la señal de entrada, llevarlo a la subrutina adecuada para filtrarlo, y luego ese resultado almacenarlo en un archivo.

Para esto empezamos abriendo dos archivos, el que contiene la señal de entrada y aquel donde se guardará la señal filtrada.

Luego inicializamos las variables de entrada x(i) y salida y(i,0), haciéndolos igual a cero. Estas variables son arreglos donde i varía de 1 a 5, esto es porque cuando se filtra una señal se utilizan los valores anteriores de la señal de entrada y de la filtrada, y de esta manera podemos simularlo. Así, el valor presente esta en i=5, el anterior en i=4, el anterior a éste en i=3, etc...

Posteriormente a esto, tomamos el primer valor de la señal y le asignamos a la variable x(5) , luego dependiendo del tipo de conversión de filtro que se haya utilizado llamamos a las subrutinas "salidapulso" o "salidabilineal" para que realicen el filtrado.
El resultado obtenido se encuentra en la variable y(5,i) , y éste lo almacenamos en el archivo correspondiente. A continuación regresamos a tomar el siguiente dato de la señal de entrada y repetimos el proceso. Esto se hace mediante un lazo DO...LOOP WHILE, que repetirá el proceso mientras no se acaben los datos de la señal.

Lo dicho lo aclararemos con el diagrama de flujo 4.15.
Esta subrutina es invocada en la subrutina "archivo".



Antes de introducirnos al análisis de las subrutinas siguientes, consideraremos algo que es común para las dos.

Conocida la función H(z) tenemos que:

$$H(z) = \frac{N(z)}{D(z)} = \frac{Y(z)}{X(z)}$$

entonces:

$$Y(z)\ D(z) = X(z)\ N(z)$$

ahora para una sección de orden r tenemos:

$$Y(z)\,[1+\alpha_{1,i}z^{-1}+\ldots+\alpha_{r,i}z^{-r}] = X(z)\,[1+\beta_{0,i}z^{-1}+\ldots+\beta_{r-1,i}z^{-r}]$$

sabemos además que :

$$y(n-k) = Z\ \{\ z^{-k}\ Y(z)\ \}$$

por lo tanto aplicando la transformada zeta inversa (Z{}) a toda la igualdad tenemos :

$$y(n)+\alpha_{1,i}y(n-1)+..+\alpha_{r,i}y(n-r)=x(n)+\beta(0,i)\,x(n-1)+..+\beta_{r-1,i}x(n-r)$$

despejando el término y(n), ya tenemos el resultado de la señal filtrada. En el programa la variable utilizada equivale a:

y(n)	en el programa es	y(5,0)
y(n-1)	en el programa es	y(4,0)
y(n-2)	en el programa es	y(3,0)
y(n-3)	en el programa es	y(2,0)
y(n-4)	en el programa es	y(1,0)

x(n)	en el programa es	x(5)
x(n-1)	en el programa es	x(4)
x(n-2)	en el programa es	x(3)
x(n-3)	en el programa es	x(2)
x(n-4)	en el programa es	x(1)

el otro parámetro k de la variable y(i,k), es utilizado para caracterizar las salidas parciales de cada sección. Entonces k representa al número de la sección.

4.3.2.16 Subrutina salidabilineal

Esta subrutina se encarga de filtrar la señal, cuando se ha utilizado la Transformada Bilineal.

Antes de describir debemos aclarar que, como las secciones de la función de transferencia se multiplican, se tienen una disposición en SERIE, llamada también RED en CASCADA.

El gráfico del flujo de la señal se lo indica en la figura 4.3

Observando el gráfico nos damos cuenta que la salida parcial de la (k-1)-ésima sección, es la entrada a la sección k-ésima. Esto quiere decir que si n es par el resultado lo obtendremos de y(5,sec) puesto que esta es la salida de la última sección, entonces es obvio que si n es impar el resultado estará en y(5,sec+1).

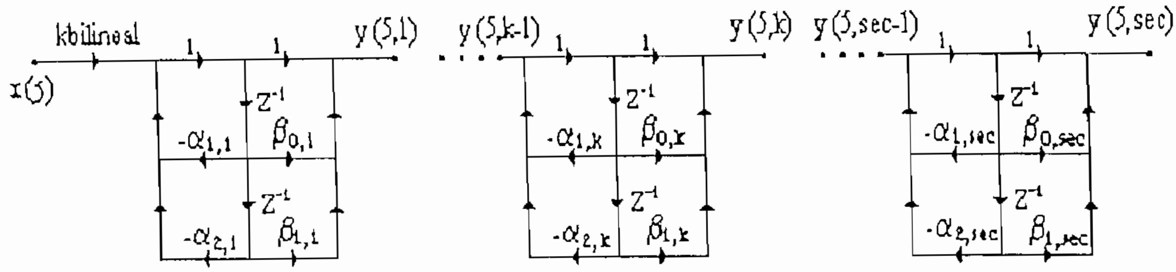


Figura 4.3: Red en Cascada

En el programa primero se evalúa la salida parcial y(5,1), por ejemplo para una sección cuadrática se hace de la siguiente manera:

$$y(5,1)=-\alpha_{1,1}y(4,1)-\alpha_{2,1}y(3,1)+kbilineal[x(5)+\beta_{0,1}x(4)+\beta_{1,1}x(3)]$$

podemos ver que sólo en la primera sección se introduce el valor de la constante kbilineal , tal como se indica en la figura 4.3.

Luego se ingresa en un lazo FOR ...NEXT de la variable k, desde k=2 hasta k=sec. En cada iteración se calcula:

$$y(5,k) = y(5,k-1)+\beta_{0,k}y(4,k-1)+\beta_{1,k}y(3,k-1)-\alpha_{1,k}y(4,k)-\alpha_{2,k}y(3,k)$$

En el desarrollo de esta subrutina se realizará el mismo procedimiento de calcular la señal filtrada, para todos los casos que se tiene. La única diferencia es que para los filtros pasa-banda o elimina-banda las secciones son de orden cuarto, y además se debe cuidar si aparece o no la sección del polo real.

Una sección de cuarto grado puede representarse como en la figura 4.4

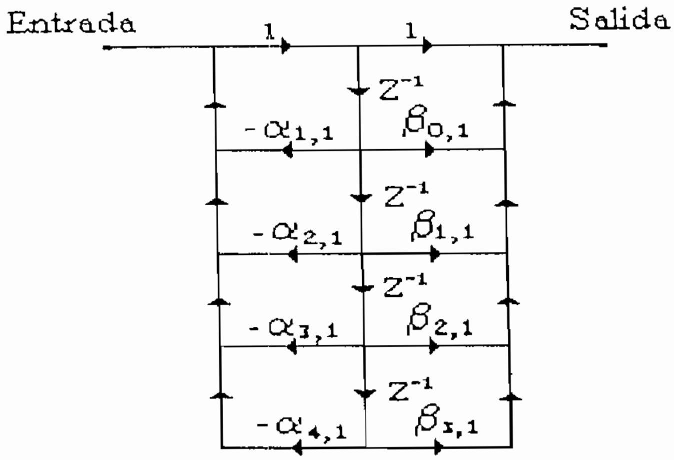


Figura 4.4: Sección de cuarto grado

La sección del polo real se ve representada en la figura 4.5

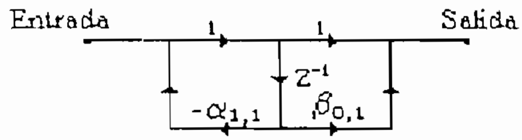


Figura 4.5: Sección del polo real

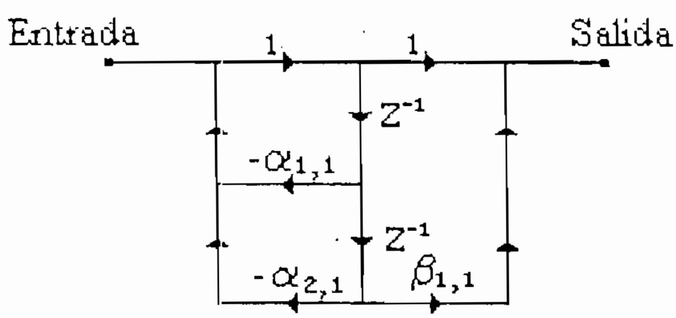


Figura 4.6: Sección del polo real para filtros pasa-banda sin ceros

Para el caso de un filtro pasa-banda, proveniente de un función analógica sin ceros, la sección del polo real tiene la forma de la figura 4.6.

En el diagrama 4.16 describimos esta subrutina.

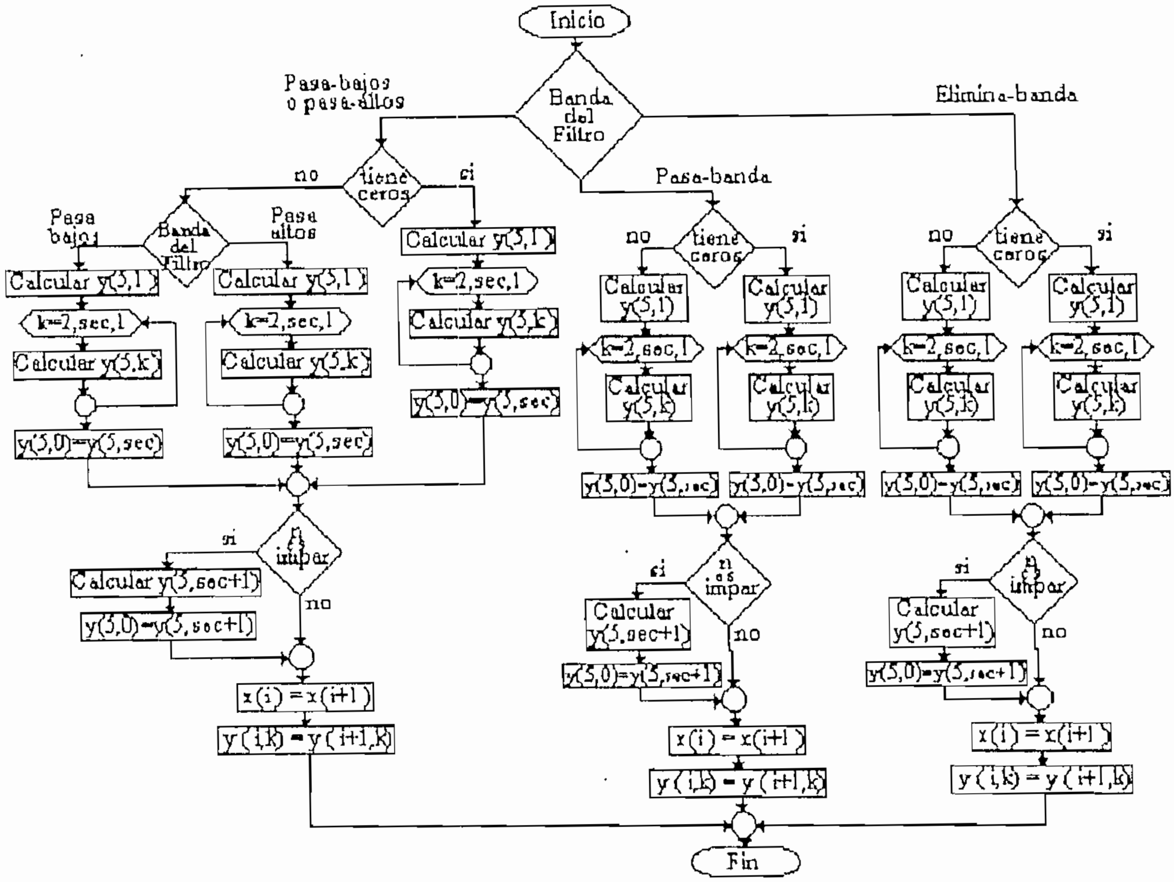


Diagrama 4.16: Subrutina "salidabilineal"

Antes de abandonar la subrutina, se actualiza los valores de <u>x(i)</u> y <u>y(i,k)</u> , que se utilizarán para filtrar el siguiente dato de la señal de entrada. Esto se lo hace así:

$$x(i) = x(i+1)$$
$$y(i,k) = y(i+1,k)$$

para i=1 hasta i=4 y k=1 hasta k=sec+1. De esta manera queda lista la variable <u>x(5)</u> , para reemplazarla en la siguiente iteración con el nuevo dato.

4.3.2.17 Subrutina salidapulso

Esta subrutina se encarga de filtrar la señal, cuando se ha utilizado la Invariancia de Impulso o la Invariancia de Pulso.

Antes de describir debemos aclarar que como las secciones de la función de transferencia se suman, se tienen una disposición en PARALELO llamada también RED en PARALELO.

El gráfico del flujo de la señal se lo indica en la figura 4.7

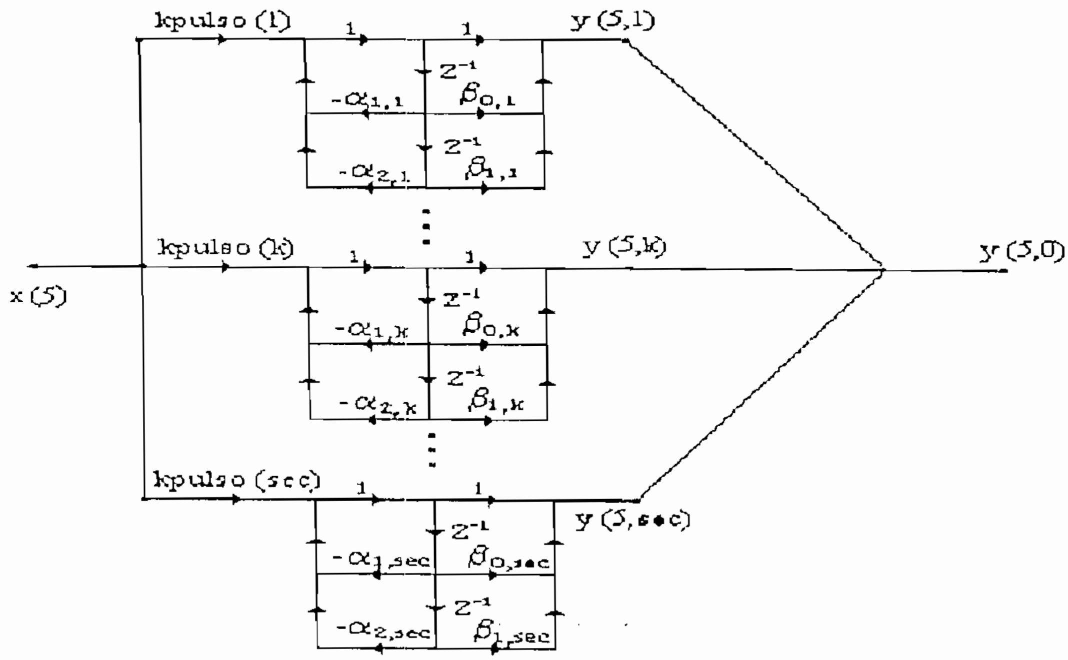


Figura 4.7: Red en Paralelo

Observando el gráfico nos damos cuenta que la señal de entrada es única para todas las secciones. De esta manera obtendremos directamente todas las salidas parciales, y luego al sumarlas tendremos el resultado.

En el programa se evalúa las salidas parciales y(5,k), por medio de un lazo FOR ...NEXT de la variable k que va desde k=1 hasta k=sec. Al final de cada iteración se agrega este valor a la variable y(5,0) .

Por ejemplo para un filtro de secciones cuadradas, en la iteración k-ésima tenemos:

$$y(5,k) = -\alpha_{1,k}y(4,k) - \alpha_{2,k}y(3,k) + kpulso[x(5) + \beta_{0,k}x(4) + \beta_{1,k}x(3)]$$

podemos ver que en todas las sección se introducirá el valor de la constante kpulso(k), tal como se indica en la figura 4.7.

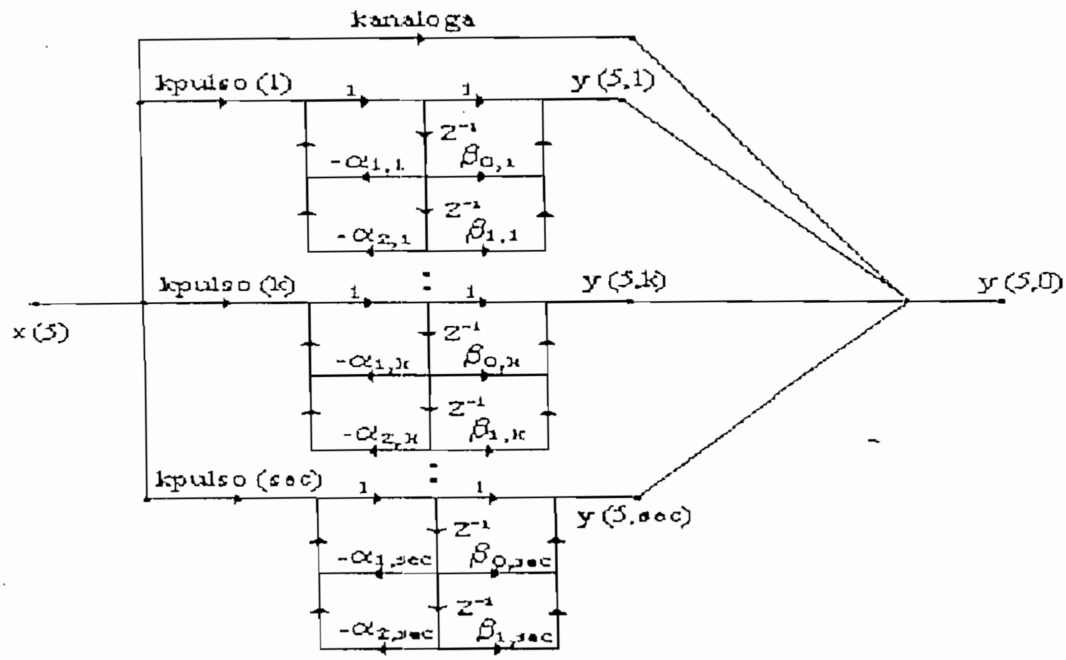


Figura 4.8: Red en Paralelo

Se debe tomar en cuenta que para los filtros pasa-banda y elimina-banda, las secciones son de cuarto grado; si el orden del filtro es impar, se debe añadir la sección del polo real; y cuando el filtro analógico inicial tiene ceros y además es de orden par, la forma del gráfico del flujo de la señal incluye la suma de la constante kanaloga. Ver la figura 4.8.

En el diagrama 4.17 describimos esta subrutina.

Antes de abandonar la subrutina, se actualiza los valores de x(i) y y(i,k), que se utilizarán con el siguiente dato tomado de la señal de entrada. Esto se lo hace de la misma forma que se hizo en "salidabilineal".

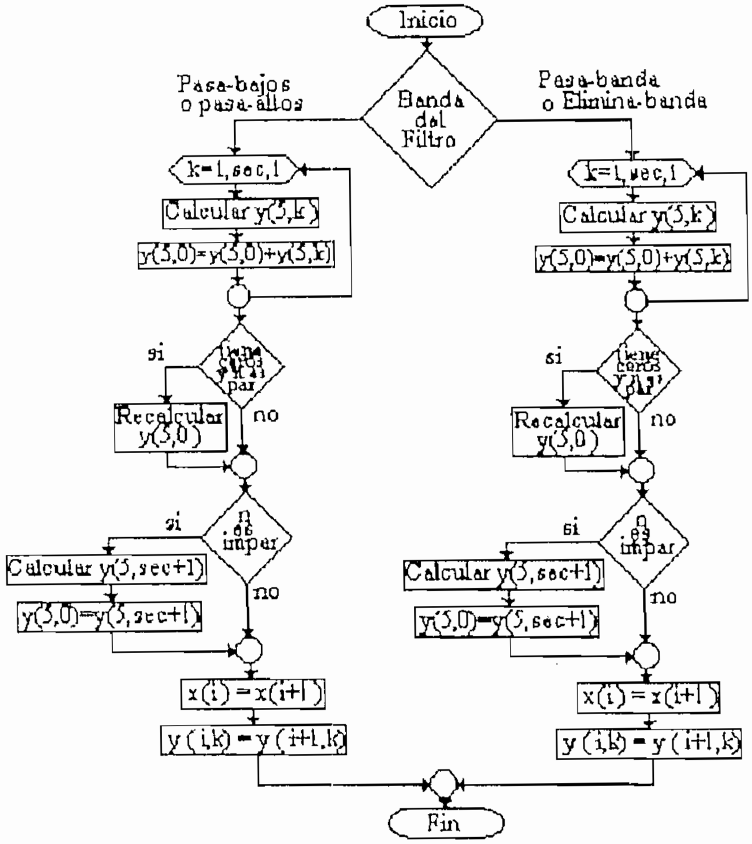


Diagrama 4.17: Subrutina "salidapulso"

Para el caso de la excepción (Filtro pasa-bajos no elíptico y con Invariancia de Pulso), no se incluyen los kpulso(k), sino que se asumen igual a 1, y además como el término independiente del numerador no aparece, se calcula la salida solo con los datos anteriores de la señal a filtrarse.

Con esta subrutina se termina las subrutinas de cálculo, que están organizadas en la subrutina "simulación". Indicaremos en el diagrama de flujo 4.19 el desarrollo de la subrutina "simulación".

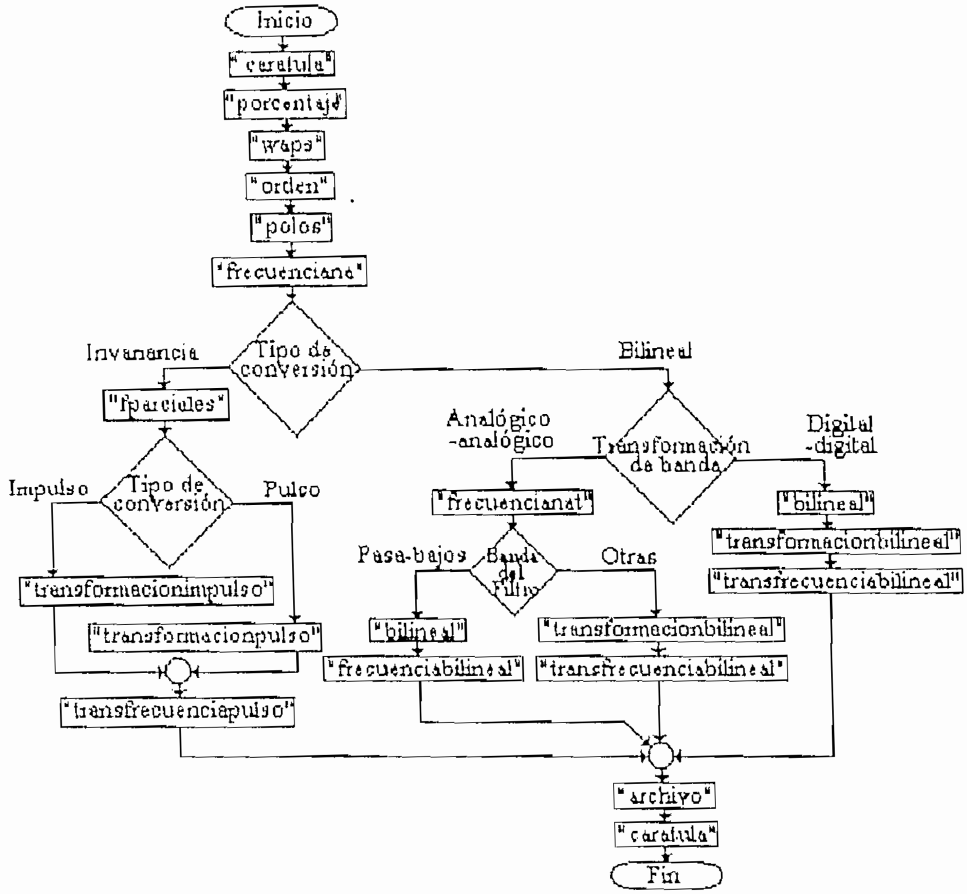


Diagrama 4.18: Subrutina "simulacion"

4.3.3 SUBRUTINAS DE PRESENTACION

Las subrutinas que se utilizan para que el programa se haga fácil de manejarlo son :

- AYUDA
- CARATULA
- CARATULAINICIAL
- DIBUJO
- FILTRAJE
- GRAFICOSMENU
- INPU
- INSTANTANEO
- PORCENTAJE

De éstas, las subrutinas "dibujo" y "filtraje", son de importancia porque presentan en la pantalla los resultados.

4.3.3.1 Subrutina dibujo

Esta subrutina se encarga de preparar la pantalla para presentar los resultados.

Dibuja las coordenadas sobre las cuales se situarán las respuestas de frecuencia, analógica y digital, la señal de entrada y la señal filtrada. Específicamente para las respuestas de frecuencia se dibuja una cuadrícula cada 0.25 unidades en el eje de las ordenadas y cada 0.2π [rad] en el eje de las abscisas.

Luego grafica las respuestas de frecuencia punto por punto, mediante un lazo de 304 puntos, correspondientes a los 304 valores almacenados en los respectivos arreglos.

Para dibujar la señal de entrada y la señal filtrada llama a la subrutina "filtraje".

4.3.3.2 Subrutina filtraje

Exclusivamente para dibujar la señal de entrada y la señal filtrada, se desarrolla esta subrutina.

El caso es que independientemente del número de muestras que tenga una señal, se debe tener la posibilidad de observar toda la señal. Esto se logra dibujando las señales por tramos. El tramo de visualización es de 600 puntos, pero se puede correr la señal de 300 en 300 puntos.

Esta subrutina abre los dos archivos, el de la señal de entrada y el de la señal de salida. Coloca un puntero, en el valor del primer punto a dibujarse, y grafica los 599 puntos siguientes. Para mover el tramo de visualización desplaza el puntero en 300 puntos en la dirección deseada.

Actualiza la barra inferior que indica la posición del tramo visto con respecto a toda la señal, cada vez que se mueve el puntero.

4.3.3.3 Subrutina ayuda

Esta subrutina es de información. Presenta en dos pantallas, todo el programa en forma resumida, incluidas sobretodo indicaciones de operación.

La subrutina tiene almacenado el texto que se visualiza para indicar las funciones que realiza cada opción del menú.

Es una ayuda inmediata que se da al usuario para que pueda utilizar el sistema.

4.3.3.4 Subrutina caratula

Esta subrutina dibuja la carátula de trabajo, que consiste en: un recuadro de toda la pantalla, en la parte superior el título " FILTRO DIGITAL ", y en la parte inferior indicaciones del uso de las teclas.

Esta subrutina se llama siempre que se empieza y se termina una operación, de esta manera, siempre la pantalla tendrá dibujada la carátula.

Se añade un sonido, para indicar que el trabajo se terminó.

4.3.3.5 Subrutina caratulainicial

Esta subrutina se encarga de dibujar la carátula de presentación del sistema. Esta subrutina se llama solo cuando se inicializa el sistema

4.3.3.6 Subrutina graficosmenu

Esta subrutina desarrolla el sub-menú de la operación gráficos, llevando la información en la variabe <u>comp</u>. La operación gráficos fue analizada en la subrutina "graficos", que pertenece a las subrutinas de datos.

4.3.3.7 <u>Subrutina inpu</u>

Esta subrutina realiza el trabajo de la sentencia INPUT pero de una manera mejorada.

Una de las ventajas es que, ingresa un número limitado de caracteres. De esta manera se evita que se dañe la pantalla de presentación por el exceso innecesario de caracteres.

Esta subrutina realiza un proceso inteligente. Cuando se desea ingresar solo números, detecta los caracteres que no lo son y los desecha.

Tiene seis parámetros de entrada, que se utilizan para:

- longitud% : determina el número de caracteres que permite ingresar.
- numero : almacena el valor numérico ingresado.
- palabra$: almacena los caracteres ingresados.
- nol% : bandera que especifica el ingreso de solo números.

  nol% = 0 → solo números

  nol% = 1 → cualquier caracter
- fila% : el número de la fila donde se colocará el cursor
- columna% : el número de la columna donde empieza la cadena de caracteres.

4.3.3.8 <u>Subrutina instantaneo</u>

Esta subrutina dibuja el recuadro con las características del filtro, que aparece al presionar <F2>.

4.3.3.9 <u>Subrutina porcentaje</u>

Esta subrutina dibuja la barra de espera cuando se está haciendo algún cálculo. Indica en forma relativa el porcentaje realizado y el que falta de realizar de cierta operación.

Ciertamente esta subrutina solo dibuja el recuadro donde se indicará el porcentaje, porque el trabajo de ir pintando a la barra lo hacen las mismas subrutinas de cálculo. El hecho es que por cada iteración de trabajo, que realicen las subrutinas de cálculo, se ordenará que se pinte una parte de la barra.

Específicamente esta barra de espera se dibuja solo en ciertas subrutinas de cálculo que son largas.

4.4 VARIABLES Y PARAMETROS DEL PROGRAMA

A lo largo de todo el programa se utilizan algunas variables y parámetros, que son nombrados de manera diferente y con nombres apropiados de acuerdo a la función que desempeñan.

Estos pueden ser de tipo GLOBAL, es decir que pueden ser utilizados, tanto en programa principal como en las subrutinas que se encuentren en el módulo principal, manteniendo el valor asignado. Además el valor podrá ser modificado desde cualquiera de estos sitios.

Las variables y los parámetros también pueden ser de tipo LOCAL, a estos se los puede manejar solamente dentro de una subrutina o dentro del programa principal. La limitación que tienen es que fuera de la subrutina en que se asignó esta variable el valor que presenta es cero.

4.4.1 VARIABLES

La utilización de variables se centra exclusivamente en las subrutinas de cálculo. Cuando se explicó el desarrollo de estas subrutinas, se fue indicando conjuntamente con el desarrollo matemático el nombre que adquieren las variables dentro del programa. A continuación presentaremos lo más relevante de las mismas.

Puesto que en nuestros cálculos se incluyen operaciones con complejos, procedimos a definir variables de tipo complejo. El tipo de la variable dentro del programa se llamará "complejo", y constará de dos partes; la parte (.real) y la parte (.imag) que serán números de doble precisión. Por ejemplo, si la variable a esta definida como un complejo, tendrá sus dos partes:

a.real representa la parte real del complejo
a.imag representa la parte imaginaria del complejo

4.4.1.1 Variables Globales

Las variables globales se definen en el programa principal con la sentencia:

DIM SHARED

De todas estas se definen como arreglos, las siguientes variables:

a) aa(1 a 200):

Este arreglo es de tipo "complejo" y su valor es asignado en la subrutina "fparciales". Representa los coeficientes de las fracciones parciales.

b) alfa(1 a 4,1 a 100),
beta(0 a 3,1 a 100),
gama(1 a 2,1 a 100):

Estos arreglos son de tipo SINGLE, es decir reales de simple precisión. Representan los coeficientes de las funciones de transferencia. Estos arreglos tienen la forma de una matriz, dado que depende de dos variables; la primera indicará la posición del coeficiente dentro de la sección, y la segunda el número de la sección.

c) c(1 a 200):

Este arreglo es un número de simple precisión y su valor es asignado en la subrutina "polos". Representa el valor del cuadrado de un cero.

d) s(1 a 200):

Este arreglo es de tipo "complejo" y su valor es asignado en la subrutina "polos". Representa a un polo de la función de transferencia analógica.

e) ha(1 a 305),
hat(1 a 305),
hb(1 a 305),
hi(1 a 305),
hp(1 a 305):

Estos arreglos son de números de simple precisión, y almacenan la respuesta de frecuencia de un filtro.

f) kpulso(1 a 200):

Este arreglo es un número de simple precisión, y representa a la constante de una sección en paralelo.

g) x(0 a 5),
y(0 a 5,0 a 100):

Estos arreglos son números de simple precisión, y se los utiliza para filtrar la señal.

Se entiende que por ser variables globales, estos arreglos se utilizan por lo menos en dos subrutinas.

Las variables que no son arreglos pero que si son globales son las siguientes:

a) alfaso, alfasola, kso, ksola, Bo, u, l:

Representan los valores de las constantes de las funciones de transformación de banda. Estas variables son calculadas por la subrutina "waps" y llevan la información hasta las subrutinas que realizan la transformación.

b) alf:

Representa al valor de la constante α, calculada en el desarrollo de los filtros de Chevishev. La información es llevada únicamente hasta la subrutina "polos".

c) attp, atts, fl, fu, fup, fm:

Estas variables sirven para llevar las especificaciones del filtro, tomadas en la subrutina "datos", hasta las subrutinas que diseñan el filtro analógico.

d) kanaloga, kbilineal:

Representan a las constantes de ciertas funciones de transferencia. Se utilizan en varias subrutinas.

e) max:

Representa el valor de amplitud máximo en la señal de entrada.

f) maximo:

Representa el número total de muestras que tiene una señal de entrada.

g) n, sec, p:

Estas variables son de tipo entero. La variable n almacena el valor del orden del filtro. La variable sec almacena el número de secciones que tiene el filtro, sin incluir la sección del polo real. La variable p puede ser considerada como parámetro, porque es un indicativo para realizar operaciones; p indica la paridad del orden del filtro, que conlleva incluir o no la sección del polo real.

h) wap, was, wac:

Representan las frecuencias de corte normalizadas. Estas son determinadas en la subrutina "waps". Cuando se usa la transformada bilineal son recalculadas en la subrutina "orden". Esta información es utilizada en la subrutina "polos".

i) nn, nreg:

Son variables utilizadas cuando se usan archivos. La variable nn es el puntero del archivo cuando estamos dibujando la señal de entrada y salida por tramos. La variable nreg indica la posición de la localidad de memoria de la cual se extrae el dato de la señal.

j) q:

Esta variable es determinada en la subrutina "orden" para efectuar un cálculo previo al cálculo del orden de los filtros elípticos. Pero esta información se necesita también en la subrutina "polos", por eso la declaramos global.

k) anal, arch1, arch2, digi, entr, sali:

Estas variables son de tipo STRING. Una variable de tipo STRING almacena caracteres de tipo ASCII. Se utilizan para guardar el nombre de archivos.

4.4.1.2 Variables Locales

Las variables locales son muchas, las mas importantes las mencionaremos a continuación:

a) i,k:

Por lo general estas variables son usadas para efectuar cálculos dentro de un lazo. La variable k le utilizamos solo cuando la variables i esta ocupada, para indicar el número de la sección que se esté tratando. La variable i se la utiliza además para especificar el punto de la respuesta de frecuencia que se esté calculando.

b) den(1 a 200),
in(1 a 200),
ini(1 a 200),
numin(1 a 200),
numini(1 a 200),
num(1 a 200):

Estos arreglos son definidos en la subrutina "fparciales", y se los utiliza para almacenar información que se va a utilizar dentro del proceso de hallar las fracciones parciales. El resultado se pone en el arreglo aa() , de modo que para liberar un poco de memoria debemos borrar estos arreglos antes de salir de la subrutina.

c) cuadro1(2000),
cuadro2(2000),
cuadro3(2000),
cuadro4(2000):

Se define estos arreglos en la subrutina "tipo" y sirven para guardar en la memoria recuadros de la pantalla. Al finalizar la subrutina borramos los arreglos.

d) cuadroa(1100),
   cuadrob(1100):

Se definen en la subrutina "filtraje". Guardan lo mismo que los arreglos anteriores, y son usados para invertir el color de las letras, estando en modo gráfico.

En algunas subrutinas hemos asignado al nombre "pi" el valor de la constante matemática π, pero esta no es una variable, sino que ha sido definida como CONSTANTE.

[4.1] En el QuickBASIC 4.50 las variables pueden ser definidas de acuerdo, a la extensión del número que se desee guardar o al tipo del mismo; de esta manera tenemos:

			RANGO	
a. Numeros enteros	(INTEGER)% :	-32768	á	32767
b. Números de simple precisión	(SINGLE)! :	1.4E-45	á	3.4E+38
c. Números de doble precisión	(DOUBLE)# :	4.9E-324	á	1.8E+308
d. Variables de tipo	(STRING)$:	0	á	32767 caracteres

4.4.2 PARAMETROS

Los parámetros son aquellos variables que se dedican a indicar que se hagan ciertas operaciones. En el programa todos los parámetros son variables de tipo entero.

La mayoría de parámetros son variables globales, con la excepción de algunos que se usan como banderas.

a) comp:

Este parámetro indica cual de las operaciones se seleccionó, al desplegar el sub-menú GRAFICOS. Los valores que adquiere éste de acuerdo a la selección son:

Selección	Valor de "comp"
RESULTADO GLOBAL	5
ARCHIVO 1	1
ARCHIVO 2	2
COMPARAR	3

b) men:

Este parámetro indica la selección tomada en el menú principal. Los valores que toma, dependiendo de la elección hecha son:

Selección	Valor de "men"
TIPO	1
ANALOGICO	2
CONVERSION	3
TRANSFORMACION	4
SIMULACION	5
ARCHIVAR	6
GRAFICOS	7
AYUDA	8
SALIR	9

c) tip:

Este parámetro indica que tipo de banda tiene el filtro que vamos a diseñar. Los valores que tiene son:

Selección	Valor de "tip"
PASA-BAJOS	1
PASA-ALTOS	2
PASA-BANDA	4
ELIMINA-BANDA	3

d) tipana:

Este parámetro indica el tipo de filtro analógico, a partir del cual se desarrolla el filtro digital. Los valores que adopta de acuerdo a la selección son:

Selección	Valor de "tipana"
BUTTERWORTH	1
CHEVISHEV TIPO I	2
CHEVISHEV TIPO II	3
ELIPTICOS	4

e) tipconv:

Este parámetro guarda la selección hecha para el tipo de conversión que digitalizará al filtro analógico. Los valores que adquiere son:

Selección	Valor de "tipconv"
INVARIANCIA DE IMPULSO	1
TRANSFORMACION BILINEAL	2
INVARIANCIA DE PULSO	3

f) trans:

Este parámetro indica si la transformación de banda será analógica-analógica o digital-digital. Los valores que toma son:

Selección	Valor de "trans"
ANALOGICA-ANALOGICA	1
DIGITAL-DIGITAL	2

Los parámetros y variables que son comunes entre dos subrutinas que se encuentran en diferentes módulos, se especifican en la declaración de estas subrutinas en el módulo principal.

REFERENCIA:

[4.1] STEVEN NAMEROFF, QuickBASIC:Manual de referencia , McGraw-Hill Interamericana de España, 1989.

CAPITULO 5

RESULTADOS

5.1 ANALISIS DE RESULTADOS

En este numeral analizaremos los resultados obtenidos al simular el filtro digital, de una manera general. El análisis en forma detallada de los resultados será mas conveniente hacerlo en el próximo numeral.

Para hacer mas comprensible la redacción de esta parte, vamos a emplear las siguientes siglas:

T: Para indicar el tipo de banda del filtro.
A: Para indicar el tipo de filtro analógico a partir del cual se ha realizado el diseño.
C: Para indicar el tipo de conversión que se ha utilizado para digitalizar al filtro
TB: Para indicar el tipo de transformación de banda utilizado.

5.1.1 FORMAS OBTENIDAS DE LAS RESPUESTAS DE FRECUENCIA DIGITAL DE LOS FILTROS SIMULADOS.

5.1.1.1 <u>Dependiendo de la banda escogida</u>

Para esto vamos a simular el filtro, con las siguientes características:

A: Butterworth
C: Invariancia de Impulso
TB: Digital-Digital

a) Forma de un filtro Pasa-bajos:

Figura 5.1: Filtro pasa-bajos

La forma que tiene es propia de un filtro pasa-bajos. Una banda de paso, sin atenuación para las frecuencias bajas, posteriormente se observa la banda de transición que cae paulatinamente, y luego la banda de supresión en las altas frecuencias.

Como el rango de frecuencia visualizado es de 2π, se ve un reflejo en la parte de π á 2π, de la parte de 0 a π. No debe confundirnos este fenómeno, ya que una consecuencia del teorema de muestreo es la repetición periódica de la respuesta de frecuencia. En todos los casos veremos este reflejo.

b) Forma de un filtro Pasa-altos:

Figura 5.2: Filtro pasa-altos

La forma indicada, pertenece a un filtro pasa-altos. Primero aparece la banda de supresión en las frecuencias bajas y luego la banda de paso en las frecuencias altas. La banda de transición asciende en forma continua hasta alcanzar la banda de paso.

c) Forma de un filtro Pasa-banda:

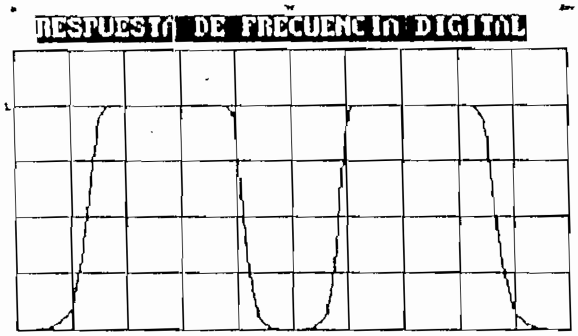


Figura 5.3: Filtro pasa-banda

Este filtro como se nota tiene dos bandas de transición, dos bandas de supresión y una de paso. La banda de paso se encuentra entre las frecuencias medias, justamente para dejar pasar todas las componentes de frecuencia que están dentro de la banda.

d) Forma de un filtro Elimina-banda:

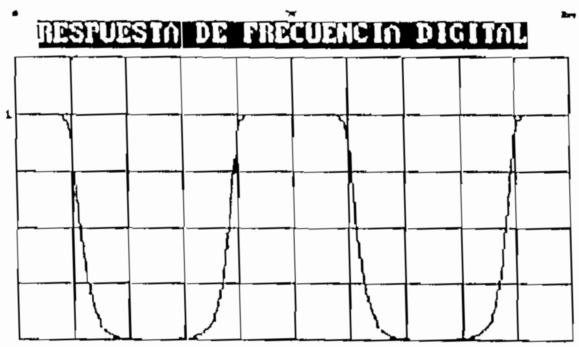


Figura 5.4:Filtro elimina-banda

De manera recíproca se tiene la forma del filtro elimina-banda, con dos bandas de transición, dos bandas de paso y una de supresión. La banda de supresión está entre las frecuencias medias, y se encarga de eliminar las componentes de frecuencia que se sitúan dentro de esta banda.

Si nos detenemos a mirar el filtro pasa-banda y el filtro elimina-banda, nos daremos cuenta que no tienen una perfecta simetría. Esto no contradice lo dicho en el capítulo 3, pues ahí se asume que el filtro es simétrico para fundamentar el desarrollo matemático, y el gráfico de este filtro sí es simétrico pero no con una alta precisión.

5.1.1.2 Dependiendo del filtro analógico que se toma como base

Vamos a simular un filtro con las siguientes características:

T : Pasa-bajos
C : Transformada Bilineal
TB : Digital-Digital

Entonces tenemos:

a) Los de Butterworth:

Como ya hemos visto la forma de éstos en el numeral anterior, podemos decir que la respuesta obtenida es una respuesta plana en todas las bandas. Además que la banda de transición se produce suavemente.

b) Los de Chevishev tipo I:

Figura 5.5: F. Chevishev tipo I

Como se ve, esta respuesta tiene el equirrizado en la banda de paso, y la caída a la banda de supresión es monótona. El filtro simulado tiene un orden N=4. Esto se puede comprobar, contando el número de picos que tiene la banda de paso y su repetición periódica.

c) Los de Chevishev tipo II:

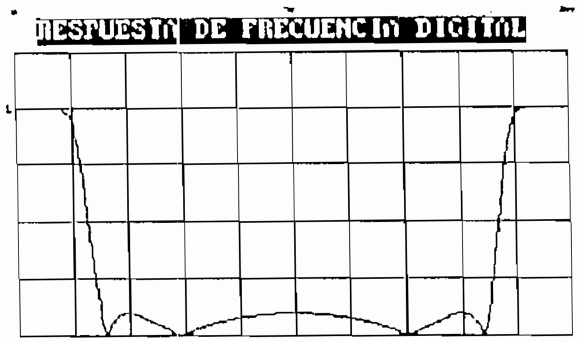


Figura 5.6: F.Chevishev tipo II

Es notorio que para este caso el equirrizado se encuentra en la banda de supresión, y que para llegar a ella la respuesta cae monótonamente desde la banda de paso. El orden del filtro simulado es N=4, que corresponde al número de picos de la banda de supresión y su repetición periódica.

d) Los Elípticos:

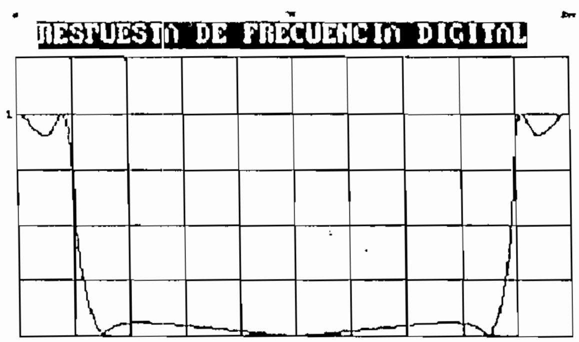


Figura 5.7: Filtro Elíptico

Esta forma tiene equirrizado en la banda de paso y en la banda de supresión. La caída de la banda de transición es rápida. El orden de este filtro es N=3. Podemos comprobar cuantificando ,ó el número de picos de la banda de paso y su repetición periódica, ó el número de picos de la banda de supresión y su repetición periódica. Si revisamos la figura 1.18, nos daremos cuenta que, la forma de esta respuesta de frecuencia es la esperada.

5.1.1.3 Dependiendo de las especificaciones del filtro

a) De la atenuación de la banda de paso:

Esta condición es mas notoria en los filtros que tienen equirrizado en la banda de paso. Consideremos un filtro que tiene las siguientes características:

T: Pasa-bajos
A: Chevishev tipo I
C: Transformada Bilineal
TB: Digital-digital

Si la atenuación es de 0.01 [dB] corresponderá a una relación adimensional de 0.9988, porque:

$$A = -20 \log(r)$$

$$r = 10^{-\frac{A}{20}}$$

donde "A" es el valor de atenuación en decibelios, y "r" es el valor de la relación adimensional correspondiente. Entonces para este caso se puede ver en la figura 5.8, que el rizado ya no se aprecia.

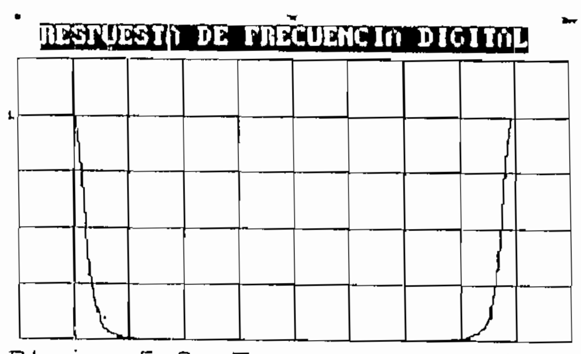


Figura 5.8: F.con atenuación de paso de 0.01[dB]

Si la atenuación es de 0.2 [dB] corresponde a una relación adimensional de 0.977. En la figura 5.9 se observa un rizado muy pequeño.

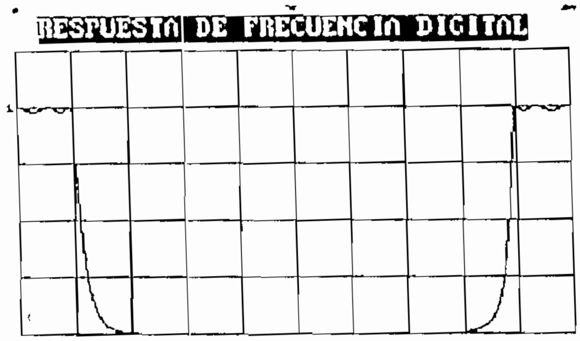


Figura 5.9: F.con atenuación de paso de 0.2 [dB]

Si la atenuación es de 1 [dB] corresponde a una relación adimensional de 0.89. En la respuesta de frecuencia de este caso, se aprecia muy bien el rizado. Vea la figura 5.5.

b) De la atenuación de la banda de supresión:

Esta condición es más notoria en los filtros que tienen equirrizado en la banda de supresión. Consideremos un filtro

Figura 5.10: F. con atenuación de supresión de 30 [dB]

que tiene las siguientes características:

T: Pasa-bajos

A: Chevishev tipo II

C: Transformada Bilineal

TB: Digital-digital

Si la atenuación es de 20 [dB], la relación adimensional es de 0.1. Esto lo podemos verificar en la figura 5.6.

Si la atenuación es de 30 [dB], la relación adimensional es de 0.031. Esto lo podemos ver en la figura 5.10.

Para atenuaciones mayores, en la respuesta de frecuencia ya no se aprecia el equirrizado.

c) De las frecuencias de corte :

Consideremos un filtro que tiene las siguientes características:

T: Pasa-bajos
A: Butterworth
C: Transformada Bilineal
TB: Digital-digital

Si tomamos un filtro de banda estrecha, con frecuencia de paso de 1200 [Hz] y frecuencia de supresión de 1800 [Hz], que normalizadas con una frecuencia de muestreo de 12000 [Hz], corresponde a 0.2π y 0.3π [rad]; la forma que tiene este filtro se presenta en la figura 5.1.

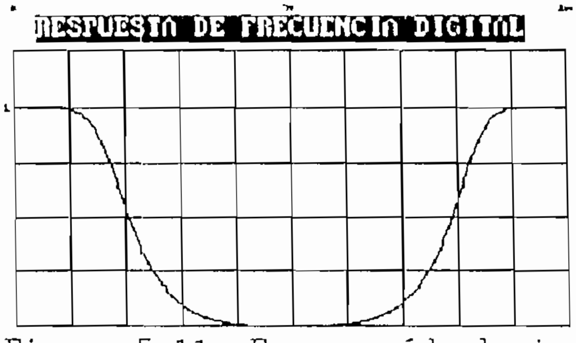


Figura 5.11: F.con caída lenta

Si movemos la frecuencia de supresión podremos darle distintas caídas a la banda de transición, así, para una frecuencia de

3600 [Hz] (ó 0.6π [rad]) la banda de transición cae muy lentamente. Vea la figura 5.11.

Pero si la frecuencia de supresión es de 1300 [Hz], la banda de transición cae rápidamente, casi en forma perpendicular. Vea la figura 5.12.

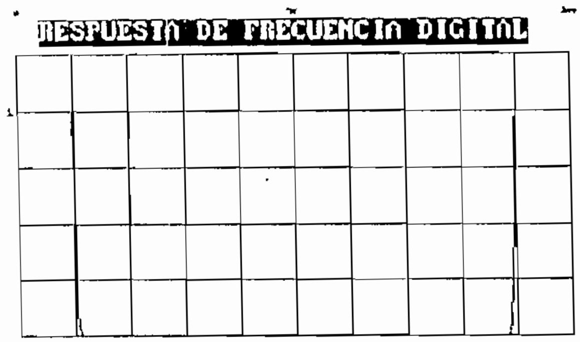


Figura 5.12: F. con caída rápida

Un filtro de banda ancha lo podemos obtener aumentando la frecuencia de paso. Por ejemplo sea esta frecuencia igual a 4200 [Hz], con una frecuencia de supresión de 4800, la respuesta de frecuencia la tenemos en la figura 5.13.

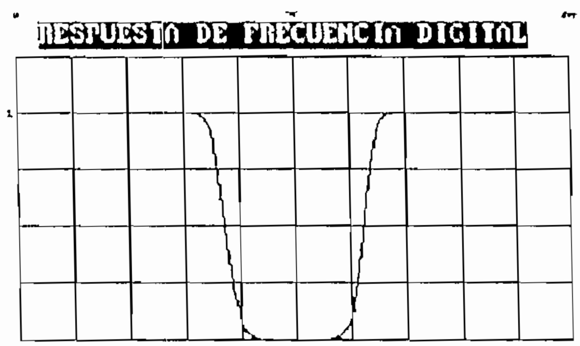


Figura 5.13: F. de banda ancha

De esta manera, nos podemos dar cuenta que podemos obtener el filtro que se nos antoje, solo con especificar las frecuencias de corte y las atenuaciones de las bandas.

5.1.2 FORMAS OBTENIDAS DE LAS RESPUESTAS DE FRECUENCIA ANALOGICA DE LOS FILTROS SIMULADOS.

En una respuesta de frecuencia analógica lo mas relevante que podemos ver es:

a) El espectro de frecuencia no es periódico.

Vea la figura 5.14. donde se representa la respuesta de frecuencia de un filtro de Chevishev tipo II, y esta no se repite.

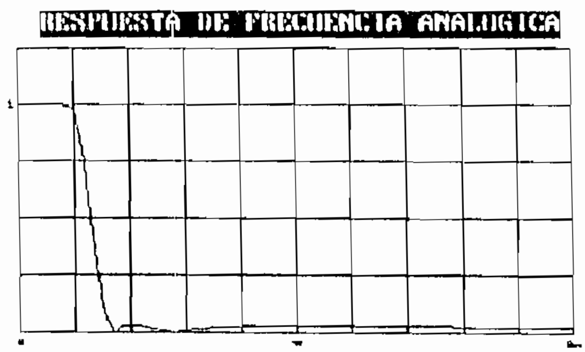


Figura 5.14: F.de Chevishev tipo II

b) La banda de transición de los filtros elípticos siempre contiene a la frecuencia de 1 [rad/seg].

En la figura 5.15. podemos observar que la frecuencia de 0.318π [rad/seg] (ó 1[rad/seg]), efectivamente corresponde a la banda de transición.

Figura 5.15: F. Elíptico

Este resultado se sustenta por lo dicho en el numeral 1.4.2.

c) Cuando la transformación de banda es digital-digital, la respuesta de frecuencia analógica siempre corresponde a un filtro pasa-bajos.

Para esto simularemos un filtro pasa-banda, y observaremos las respuestas de frecuencia analógica y digital en la figura 5.16

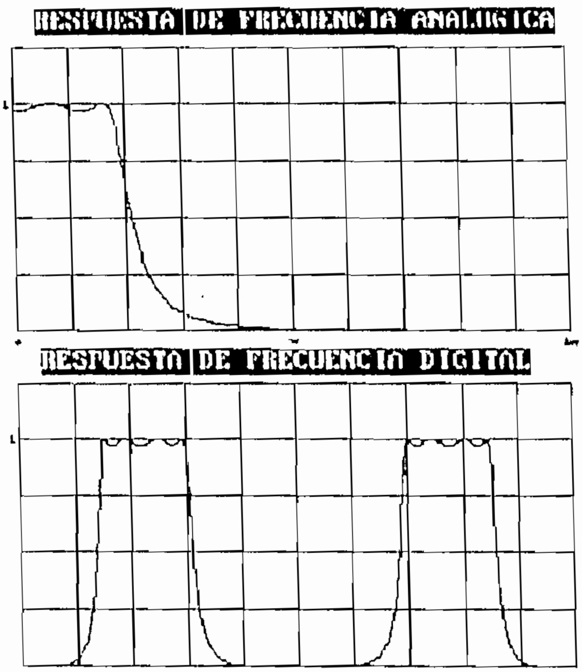


Figura 5.16: Transformación Digital-Digital

d) Cuando la transformación de banda es analógica-analógica, la respuesta de frecuencia analógica corresponde al tipo de banda del filtro especificado.

Simularemos un filtro elimina-banda, y veremos este resultado en la figura 5.17.

En este gráfico en la parte superior aparecen superpuestas las dos respuestas de frecuencia analógicas, antes y después de la transformación de banda.

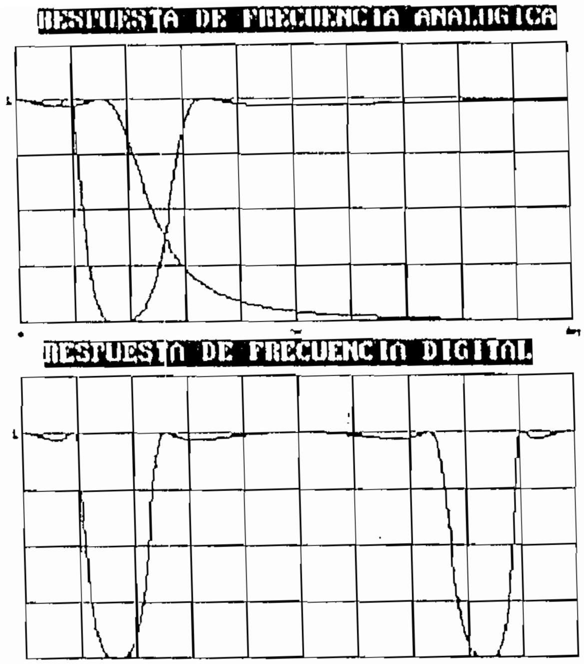


Figura 5.17: Transformación Analógica-Analógica

5.1.3 FORMAS OBTENIDAS DE LA SEÑAL FILTRADA.

Para poder mostrar estos resultados hemos escogido una señal de entrada adecuada, vea la figura 5.18. Esta señal corresponde al sonido de la vocal "i", y como se observa esta señal tiene componentes de baja y alta frecuencia. La alta frecuencia se ve claramente por la presencia de picos a lo largo de la señal, y la baja frecuencia por la forma de sinusoide no perfecta de mayor amplitud.

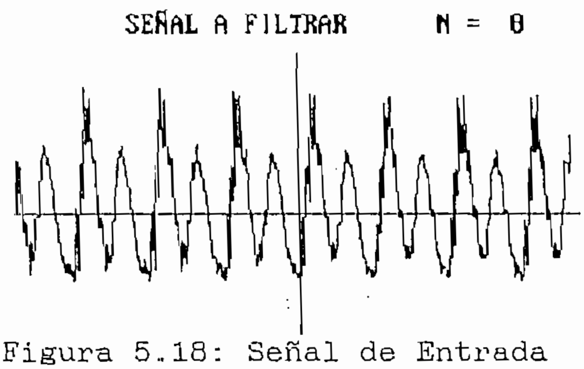


Figura 5.18: Señal de Entrada

Al filtrar esta señal, obtendremos los siguientes resultados:

a) Por un filtro pasa-bajos:

En la figura 5.19. vemos que toda la alta frecuencia se ha eliminado, quedándonos una señal de baja frecuencia. Esta señal es la sinusoide no perfecta de amplitud mayor, que no tiene picos.

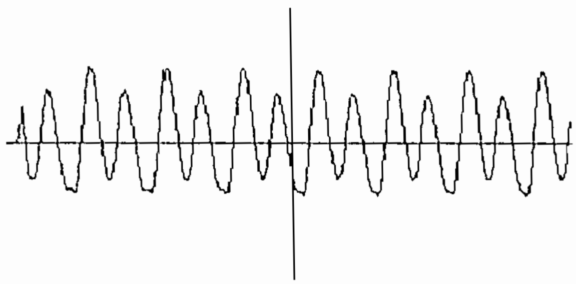


Figura 5.19: Señal filtrada por un filtro pasa-bajos

Las especificaciones de este filtro son:

- Atenuación de la banda de paso = 1 [dB]
- Atenuación de la banda de supresión = 20 [dB]
- Frecuencia de paso = 1200 [Hz]
- Frecuencia de supresión = 1800 [Hz]

b) Por un filtro pasa-altos:

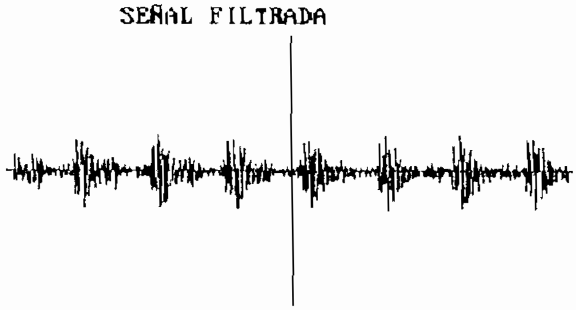


Figura 5.20: Señal filtrada por un filtro pasa-altos

En este caso se elimina la sinusoide no perfecta, que corresponde a las componentes de baja frecuencia. Nos queda una señal periódica de mediana amplitud, pero de alta frecuencia. Vea la figura 5.20

Las especificaciones de este filtro son:

- Atenuación de la banda de paso = 1 [dB]
- Atenuación de la banda de supresión = 20 [dB]
- Frecuencia de paso = 1800 [Hz]
- Frecuencia de supresión = 1200 [Hz]

c) Por un filtro pasa-banda:

Para poder notar el efecto de este filtro, filtraremos la señal con un filtro pasa-banda tan selectivo que dentro de lo posible pase una sola componente de frecuencia. Se puede observar, en la figura 5.21, que el resultado corresponde a una sinusoide casi perfecta, de frecuencia igual a la seleccionada en la banda de paso del filtro.

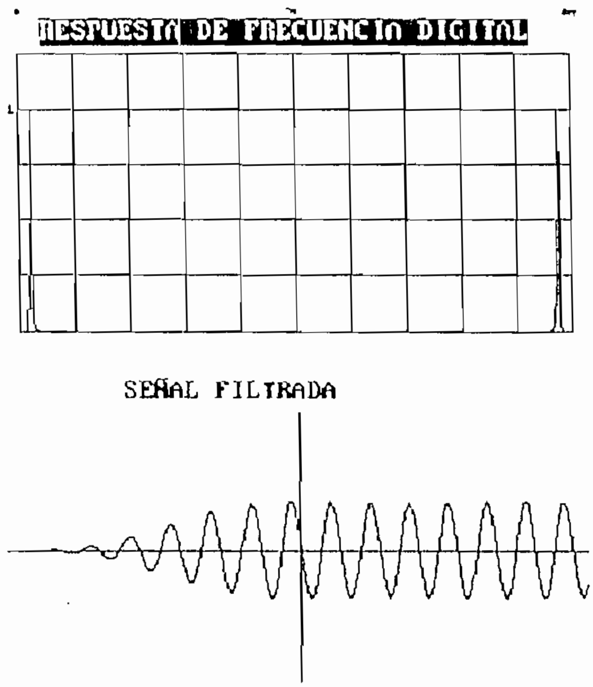


Figura 5.21: Filtrada por un filtro pasa-banda

Las especificaciones de este filtro son:

- Atenuación de la banda de paso = 0,3 [dB]

- Atenuación de la banda de supresión = 20 [dB]
- Frecuencia de paso de la banda de transición inferior = 250 [Hz]
- Frecuencia de supresión de la banda de transición inferior = 200 [Hz]
- Frecuencia de paso de la banda de transición superior = 300 [Hz]

En el gráfico de la respuesta de frecuencia, se nota la alta selectividad del filtro. La alta selectividad hace que la señal filtrada tenga un tiempo considerable de inicialización.

d) Por un filtro elimina-banda:

Para ver este efecto, eliminaremos la banda que en el filtro anterior fue de paso. Entonces se tendrá una señal deformada, debido a que se quitó una de sus componente de frecuencia. Vea la figura 5.22.

Las especificaciones de este filtro son:

- Atenuación de la banda de paso = 0.3 [dB]
- Atenuación de la banda de supresión = 99 [dB]
- Frecuencia de paso de la banda de transición inferior = 200 [Hz]
- Frecuencia de supresión de la banda de transición inferior = 250 [Hz]
- Frecuencia de paso de la banda de transición superior = 300 [Hz]

Por tener una alta selectividad se eligió una atenuación de supresión bastante alta.

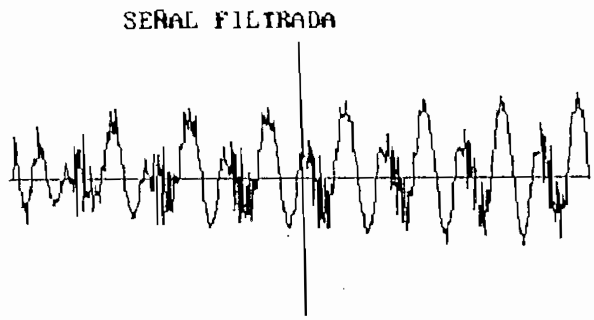


Figura 5.22: Filtrado por un filtro elimina-banda

La señal filtrada, tiene un período de inicialización grande; se debe a la alta selectividad.

5.2 ESTUDIOS COMPARATIVOS

De los resultados obtenidos al simular el filtro, los estudios comparativos que podemos realizar son:

5.2.1 RESPECTO DE LA RESPUESTA DE FRECUENCIA DIGITAL

a) Al simular un filtro pasa-bajos de Butterworth, nos damos cuenta que para las mismas especificaciones, la respuesta de frecuencia digital obtenida al digitalizar con la Invariancia de Impulso es mas estrecha que al digitalizar con la Transformada Bilineal; observe la figura 5.23.

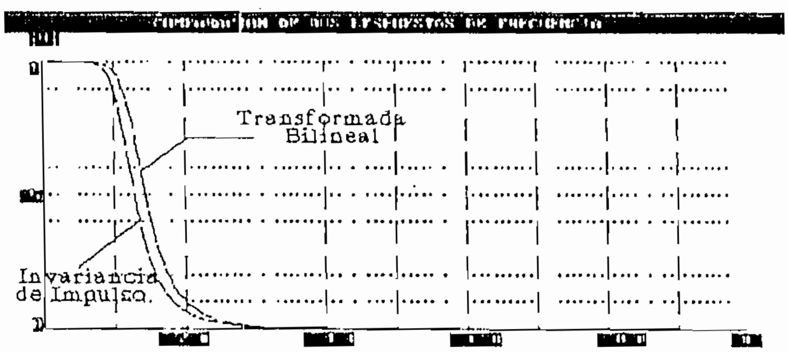


Figura 5.23: Filtro de Butterworth

Este resultado concuerda con la figura 1.5., puesto que en la subrutina "orden" se estableció el cálculo de Ωc de acuerdo al método de digitalización que utilicemos.

b) En el diseño de los filtros, utilizando la Transformada Bilineal, se incluye el prealabeo de frecuencia para el filtro analógico, de tal manera que el filtro digital cumpla con los requerimientos. Este prealabeo es mayor para los filtros de banda mas ancha. Observemos las figuras 5.24 y 5.25, donde se grafican la respuesta digital vs. la respuesta analógica. En ambos casos las respuesta analógica es la mas ancha.

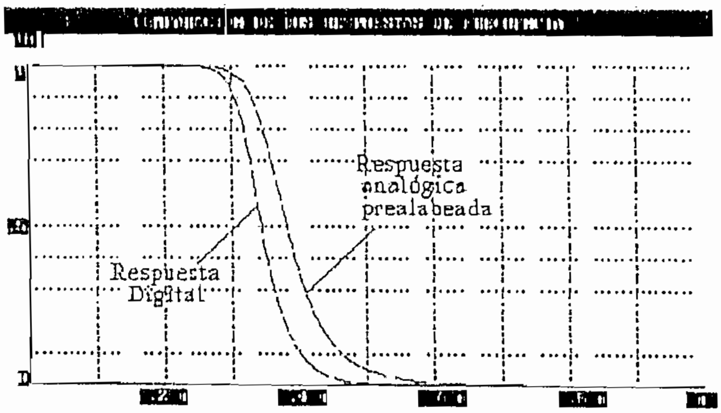


Figura 5.24: F. de banda estrecha

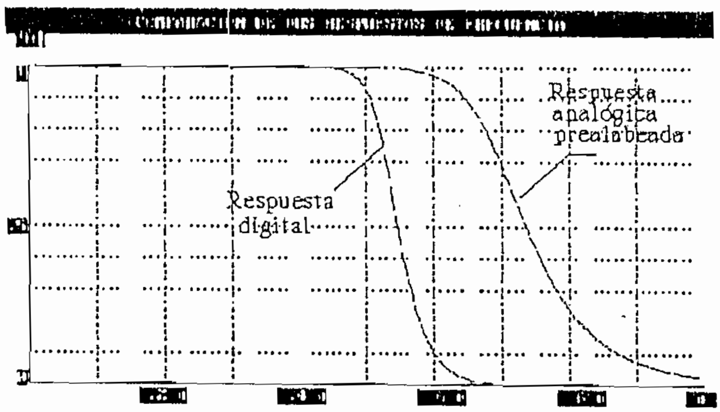


Figura 5.25: F. de banda ancha

c) Hemos detectado cruce de espectros en los filtros que tienen equirrizado en la banda de supresión, estos son los filtros de Chevishev tipo II y los filtros Elípticos.
Estos cruces de espectro se localizan en los filtros digitalizados por Invariancia, siendo mas notorio en la Invariancia de Impulso.

Tomemos un filtro pasa-bajos de Chevishev tipo II con atenuación de supresión igual a 20 [dB], las respuestas de frecuencia para los distintos métodos de conversión, analógica en digital, son:

Figura 5.26: Con Invariancia de Impulso

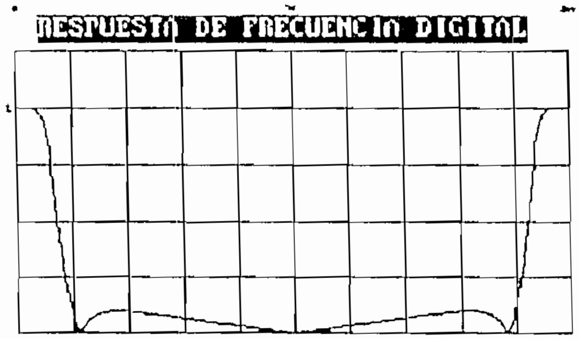


Figura 5.27: Con Transformada Bilineal

Figura 5.28: Con Invariancia de Pulso

En la figura 5.26 se tiene una respuesta de frecuencia cuyo módulo máximo excede el valor adimensional de 1. Este resultado es debido al cruce de espectros, y lo podemos comprobar aumentando la atenuación de la banda de supresión, de tal manera que se vaya eliminando el equirrizado y el filtro analógico ya no tenga la respuesta de frecuencia de banda infinita. Para una atenuación de supresión de 30 [dB], tenemos: (Vea la figura 5.29)

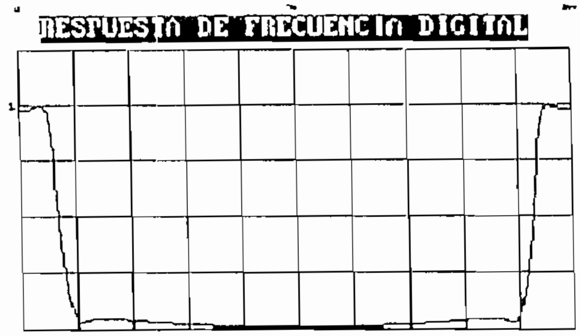


Figura 5.29: Con atenuación 30[dB]

Vemos que el efecto producido tiene otra forma, pero se acerca mas a la respuesta de frecuencia esperada.
Para una atenuación de supresión de 50 (figura 5.30), el efecto casi desaparece, lo que significa que, para atenuaciones mayores el efecto ya no se verá.

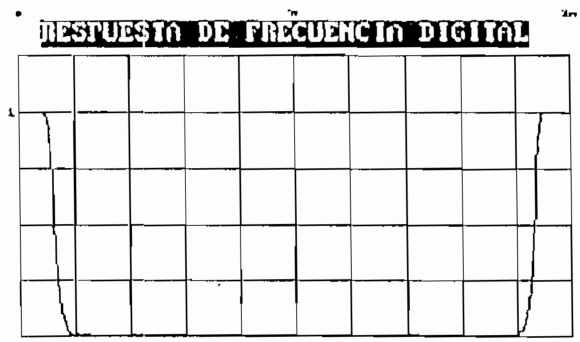


Figura 5.30: Con atenuación de 50 [dB]

A continuación veremos que para los filtros elípticos el efecto es similar. Simularemos un filtro pasa-bajos elíptico con los diferentes métodos de conversión, y con una atenuación de supresión de 20 [dB]. Los resultados obtenidos se observan en las figuras 5.31, 5.32 y 5.33.

Figura 5.31: Con Invariancia de Impulso

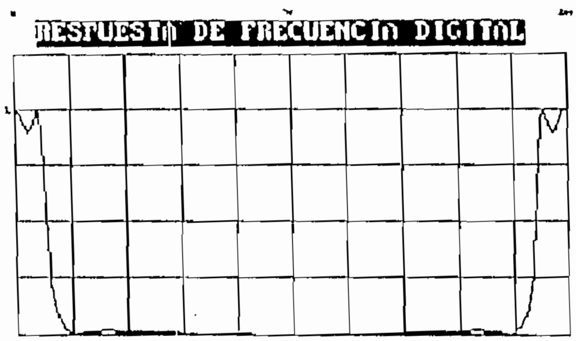


Figura 5.32: Con Transformada Bilineal

Figura 5.33: Con Invariancia de Pulso

Hasta aquí podemos ver la severa ventaja al utilizar Invariancia de Pulso respecto de utilizar la Invariancia de Impulso.

La Invariancia de Pulso produce cruce de espectros pero en estos filtros no se ha notado. En los filtros pasa-altos de Butterworth el efecto se nota como una pequeña caída al comenzar la banda de paso, pero que no se sale de la especificación establecida. Esto lo vemos en las figuras 5.34, 5.35, y 5.36

Figura 5.34: Con Invariancia de Impulso

Figura 5.35: Con Transformada Bilineal

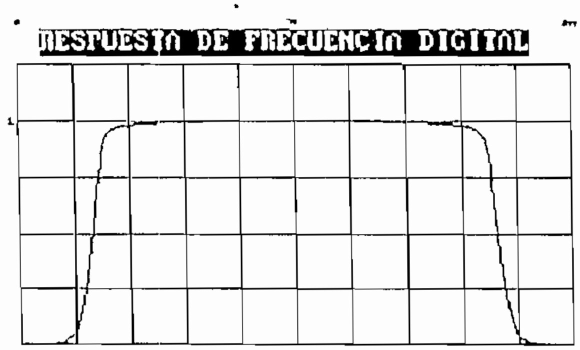


Figura 5.36: Con Invariancia de Pulso

Los efectos producidos debido al cruce de espectros son mas acentuados en los filtros pasa-banda y elimina-banda.

d) Cuando simulamos un filtro utilizando transformación de banda analógica-analógica y digital-digital, los filtros son idénticos. A continuación presentaremos estos resultados en las figuras 5.37, 5.38, y 5.39.

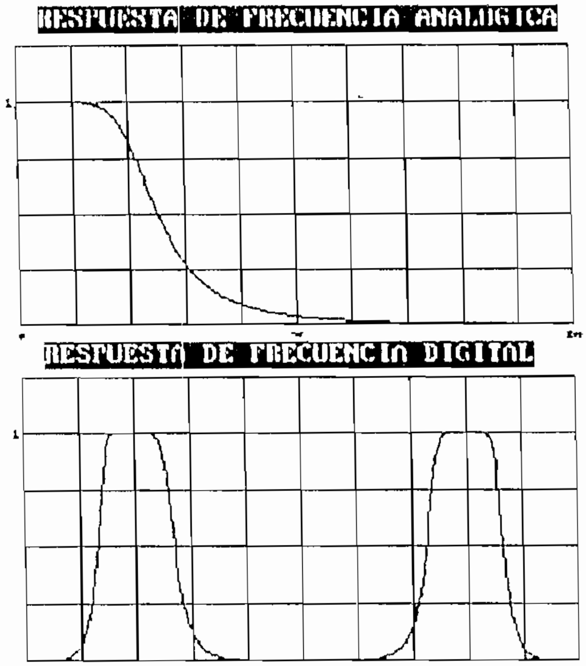


Figura 5.37:Con transformación
Digital-Digital

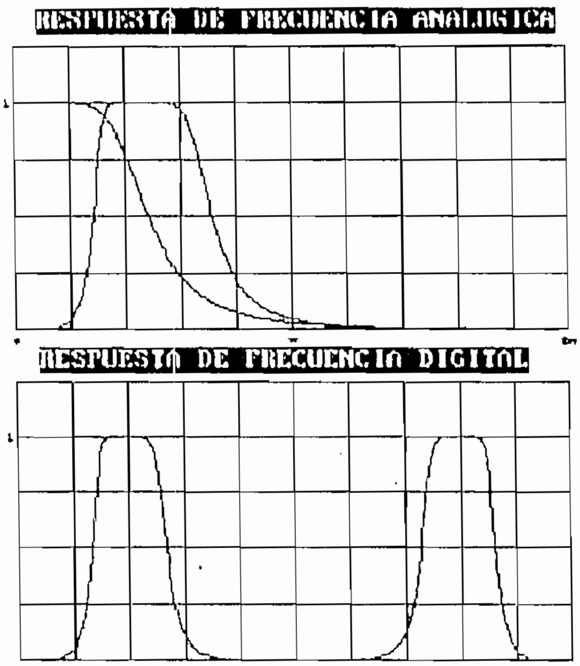


Figura 5.38:Con Transformación
Analógica-Analógica

En las figuras 5.37 y 5.38 se presenta la respuesta digital de un filtro pasa-banda, conjuntamente con la respuesta analógica. En la figura 5.38 también se puede ver que la respuesta analógica esta prealabeada, porque se utiliza la

Transformada Bilineal. En la figura 5.39 se ha dibuja las dos respuestas de frecuencia digital de este análisis. Como se ve estas se han superpuesto, ratificando lo que se dijo previamente.

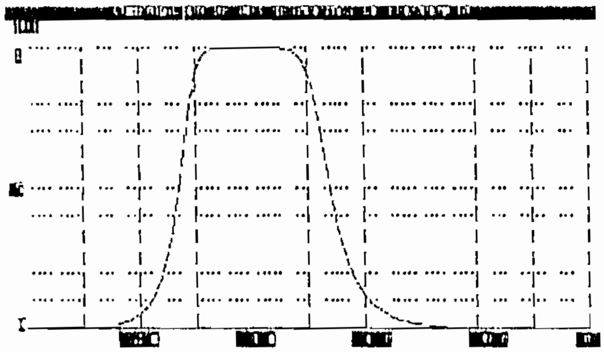


Figura 5.39: Dig-Dig. vs. Anal-Anal.

El filtro con transformación de banda analógico-analógico se demora 1 minuto con 11 segundos en simularse, mientras que con la transformación digital-digital se demora 51 segundos. Esto es porque se calcula una respuesta de frecuencia adicional, perteneciente al filtro analógico transformado en banda.

5.2.2 RESPECTO DEL ORDEN DE LOS FILTROS

Simularemos un filtro de Butterworth, uno de Chevishev y un Elíptico, para las mismas especificaciones, y determinaremos el orden que estos adquieren.

Para un filtro de especificaciones:

- Atenuación de paso = 1 [dB]
- Atenuación de supresión = 20 [dB]
- Frecuencia de paso = 1200 [Hz]
- Frecuencia de supresión = 1800 [Hz]

Los resultados obtenidos son:

Butterworth	Chevishev	Elíptico
6	4	3

Si aumentamos la atenuación de supresión a 40 [dB], tenemos:

Butterworth	Chevishev	Elíptico
10	6	5

Si aumentamos la atenuación de supresión a 60 [dB], tenemos:

Butterworth	Chevishev	Elíptico
14	8	6

Si aumentamos la atenuación de supresión a 80 [dB], tenemos:

Butterworth	Chevishev	Elíptico
18	10	7

Para aumentar el orden hacemos mas estrecha la banda de transición, cambiando la frecuencia de paso a 1500 [Hz]. Los resultados así obtenidos son:

Butterworth	Chevishev	Elíptico
29	12	9

Ahora cambiando la atenuación de paso a 0.1 [dB], tenemos:

Butterworth	Chevishev	Elíptico
32	14	10

Es notorio que para los filtros de Butterworth el orden necesitado es bastante mayor a los otros filtros.

En cambio el filtro elíptico tiene la ventaja de cumplir los requerimientos con un filtro de orden pequeño.

Los filtros de Chevishev tipo I y tipo II, tienen el mismo orden. Estos filtros necesitan de un orden un poco mayor a los elípticos.

En los filtros pasa-banda y elimina-banda, el orden del filtro se va reduciendo mientras mayor sea el acercamiento de las bandas de transición.

5.2.3 RESPECTO DEL RETARDO DE LA SEÑAL FILTRADA

Utilizando la Invariancia de Impulso, filtraremos una señal con un filtro de orden n=4 (figura 5.40) y con un filtro de orden n=25 (figura 5.41).

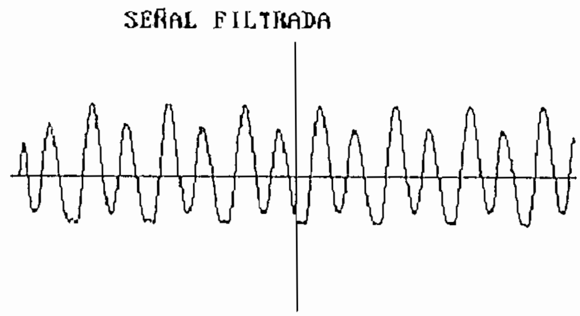


Figura 5.40: Con n=4 e Invariancia de Impulso

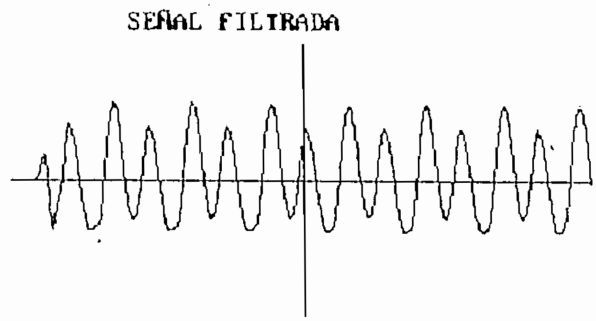


Figura 5.41: Con n=25 e Invariancia de Impulso

Observando la señal en la posición central (donde se encuentra un eje de referencia vertical), nos podemos dar cuenta claramente que la señal se encuentra retardada mayormente para el filtro de orden mayor.

Al utilizar la Transformada Bilineal en un filtro de orden n=25, la señal tiene el mismo retardo, que se observó en la figura 5.41. Esto lo podemos comprobar observando la figura 5.42.

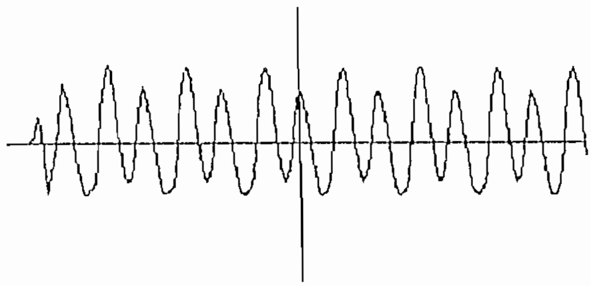


Figura 5.42: Con n=25 y Transformación Bilineal

Cuando la señal se filtra en tiempo real, el retardo es mayor al utilizar la Transformación Bilineal, puesto que las secciones se colocan en serie. Pero al simular estos filtro en tiempo no real, no vemos tal consecuencia.

Lógicamente un filtro con mayor número de secciones, emplea mas tiempo para simularse.

5.3 COMENTARIOS Y CONCLUSIONES

- El poder simular un filtro, es de gran utilidad, sobre todo por la versatilidad que se tiene en cambiar las especificaciones de este, sin que esto influya un costo adicional.

- El programa desarrollado filtra señales digitales, previamente almacenadas en un archivo de datos. Los archivos tratados tienen datos que pueden ocupar 2 bytes. En general, los archivos de datos pueden tener diferentes formatos. Uno de los formatos más utilizados en la actualidad son los archivos " *.wav " que se los obtiene al digitalizar una señal con la tarjeta SOUND BLASTER del kit de multimedia para WINDOWS en computadores compatibles con IBM. Este programa reconoce el formato *.wav, de esta manera este programa tiene un gran campo de acción.

- Este filtro puede utilizarse para editar señales. Podemos recortar o dejar ciertas componentes de frecuencias que sean de interés. En señales de audio con este filtro podremos eliminar ruido o podremos escuchar algún sonido que sea de interés, conociendo la frecuencia de este.

- En la simulación del filtro, es de mucha importancia la intervención de las funciones de transferencia en forma de secciones, puesto que de esta manera se optimiza el trabajo; se puede simular una sección y utilizarla recursivamente para cualquier número de secciones.

- Cualquier inexactitud pequeña que se tenga en los resultados de las respuestas de frecuencia, puede asumirse que es por causa de aproximar el valor del orden n. Este factor que produce imprecisión es imposible de eliminar, porque en la realidad no se podrán crear secciones de orden fraccionario.

- Cuando digitalizamos un filtro mediante Invariancia de Impulso o de Pulso, se recomienda tener cuidado no tanto en el ancho de la banda de paso, sino mas bien, en la banda de supresión ya que el fenómeno del aliasing parece depender de la amplitud del rizado de la banda de supresión.

- La ventaja de digitalizar un filtro con la Invariancia de Pulso se ve sobre todo en que el efecto de los cruces de espectro es menor que en la Invariancia de Impulso. En cuanto a cualquier imperfección que se tenga en la señal filtrada, cuando esta no es tipo pulso (señales no cuadradas), debe ser porque el análisis de esta invariancia se partió de señales que son tipo pulso.

- La Invariancia de Pulso ya es una herramienta usada en la simulación de filtros analógicos, en un enlace de comunicaciones [5.1].

- La Transformada Bilineal no presenta complicaciones ni restricciones de banda, por lo que se recomienda su utilización.

- Una ventaja que se considera para utilizar la transformación de banda digital-digital es el tiempo de simulación que resulta inferior al de la transformación de banda analógica-analógica.

REFERENCIA:

[5.1] FLOYD M. GARDNER, "A Transformation for Digital Simulation of Analog Filters", IEEE Transaction on Comunications , Vol COM-34, NO.7, July 1986.

ANEXO A

MANUAL DEL USUARIO

ANEXO A

MANUAL DEL USUARIO

1. REQUERIMIENTOS DEL COMPUTADOR

Sobre todo se recomienda utilizar un computador que realice rápidamente los cálculos matemáticos, para no esperar mucho por los resultados.

De los entornos, no es necesario poseer "mouse", ni tarjeta digitalizadora, ni impresora.

Se recomienda utilizar un monitor a colores, o por lo menos un monitor VGA con 16 tonalidades de gris, puesto que el programa está realizado en modo gráfico; inclusive para comparar respuestas de frecuencia se han superpuesto funciones gráficas en varios colores.

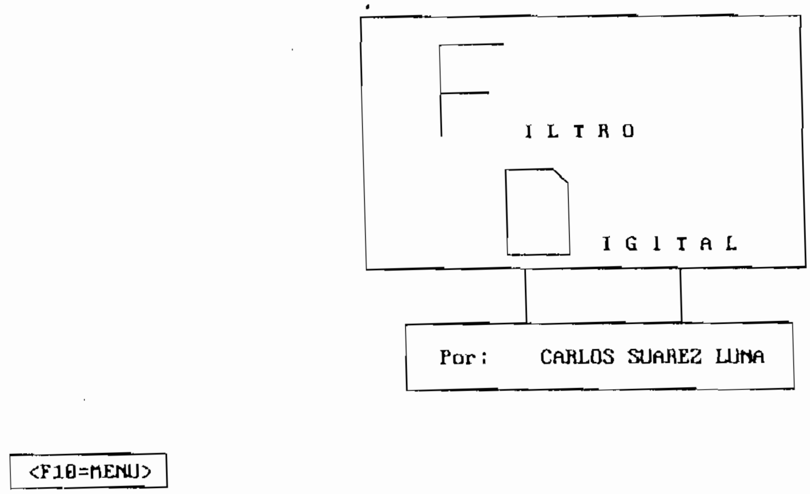
2. INSTALACION

Se le proporciona al usuario los archivos, FDIGITAL.BAS, FDIGITA2.BAS, FDIGITA3.BAS y FDIGITAL.EXE. Para arrancar el sistema, podemos directamente utilizar el archivo FDIGITAL.EXE bajo DOS. Otra alternativa es introduciéndonos al entorno del QuickBASIC y desde allí: llamar al archivo FDIGITAL.BAS como módulo principal y luego, cargar a los archivos FDIGITA2.BAS y FDIGITA3.BAS como módulos agregados.

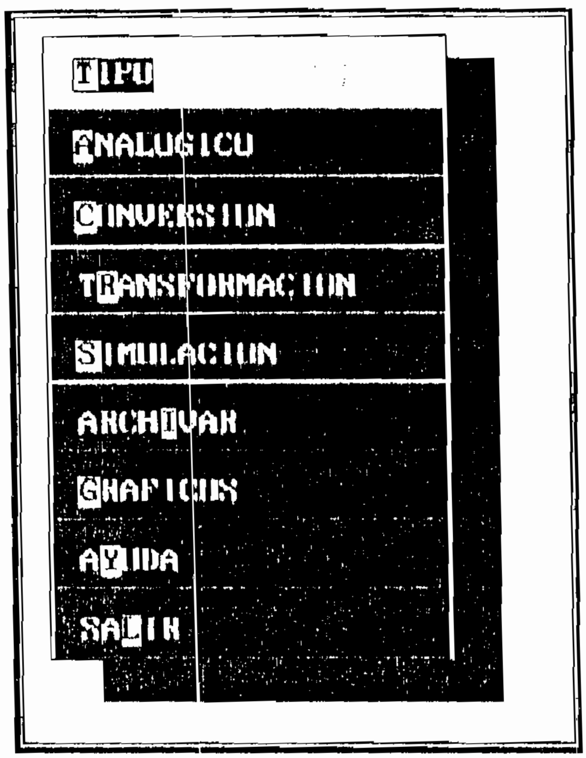
3. OPERACION

Al instante, en que se arranca el sistema aparece una pantalla de presentación invitándonos a activar el menú con la tecla

<F10>.



El menú que se despliega es de la siguiente forma:



en éste mediante las flechas podemos movernos hasta localizar la opción seleccionada. Para activar cualquier opción, presionaremos la tecla <ENTER>. Otra manera de utilizar el menú es presionando la letra resaltada; entonces se selecciona y se activa a la vez la opción.

Este menú aparece dentro de una carátula, que, en la parte superior tiene el título de "FILTRO DIGITAL" y en la inferior indicación de las teclas para el manejo.

Existen cuatro indicaciones de teclas, dispuestas de la siguiente manera:

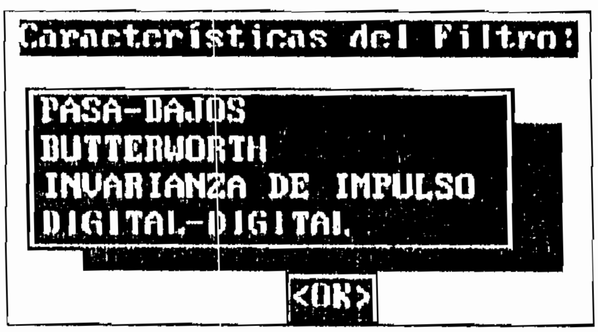
<FLECHAS=Seleccionar> <ENTER=Aceptar> <F2=Características> <F10=Menú>

Cada una de estas aparece, siempre y cuando esta tecla esté disponible.

Con la tecla <ENTER> podemos ejecutar tareas o confirmar el ingreso de datos.

Al presionar la tecla <F2>, aparece en la pantalla un pequeño recuadro en la parte superior derecha, donde se especifican las características del filtro a simular. Esta tecla siempre se puede accionar. Estas características corresponden respectivamente al: tipo de banda del filtro, tipo de filtro analógico en base al cual se hará el diseño, método de conversión del filtro analógico en digital que se emplea, y al método de transformación de banda utilizado.

Como ejemplo, se ven las características del filtro con el que se inicializa el programa:



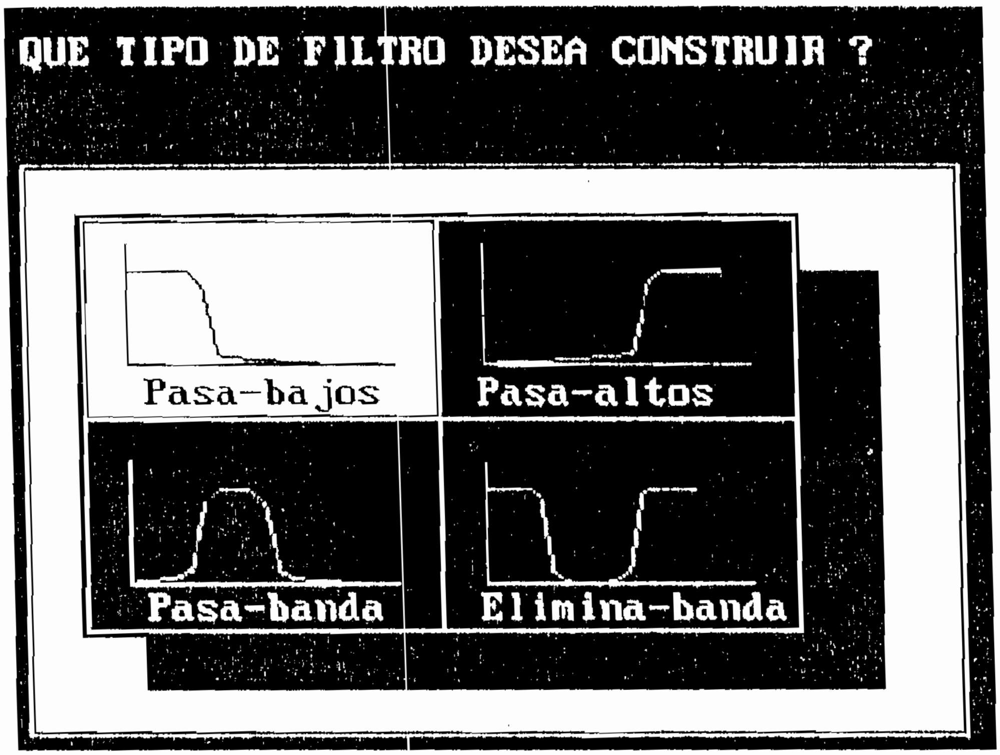
Este recuadro desaparece al presionar <ENTER>.

La tecla <F10>, despliega al menú principal. Esta tecla está activa siempre, excepto cuando el programa está dibujando las respuestas de frecuencia o las señales en tratamiento. Esta tecla en ocasiones, es la única alternativa para finalizar una tarea en ejecución.

Al inicializar el sistema hay dos opciones que no se las puede ejecutar, éstas son: TRANSFORMACION y ARCHIVAR. El motivo de esto lo veremos después.

3.1 Opción TIPO

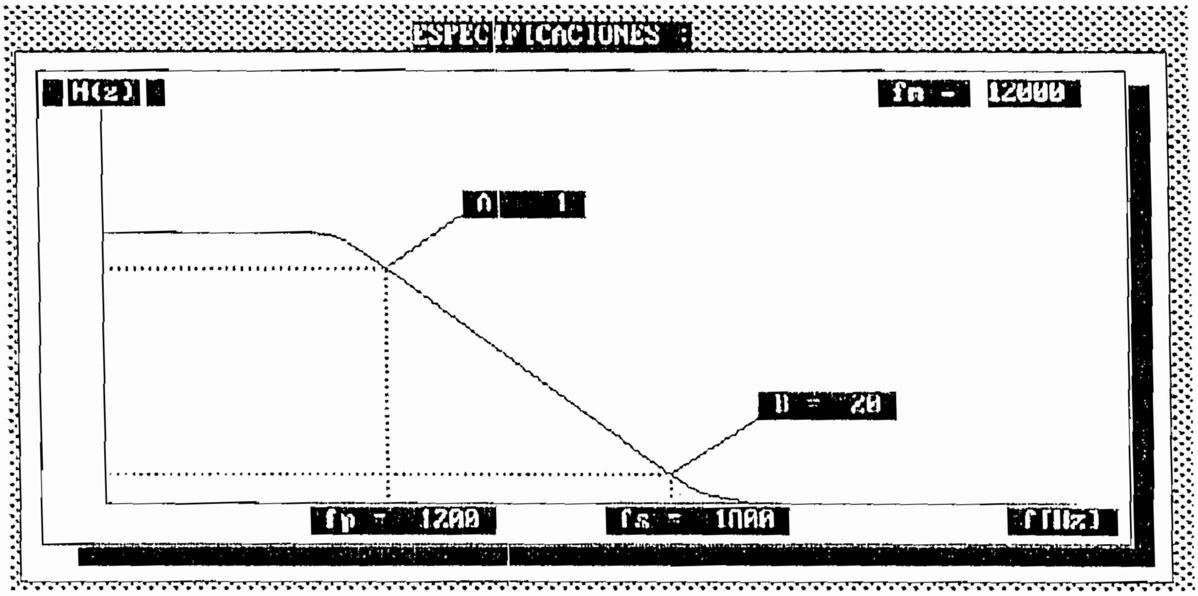
Al tomar esta opción aparece la siguiente pantalla :



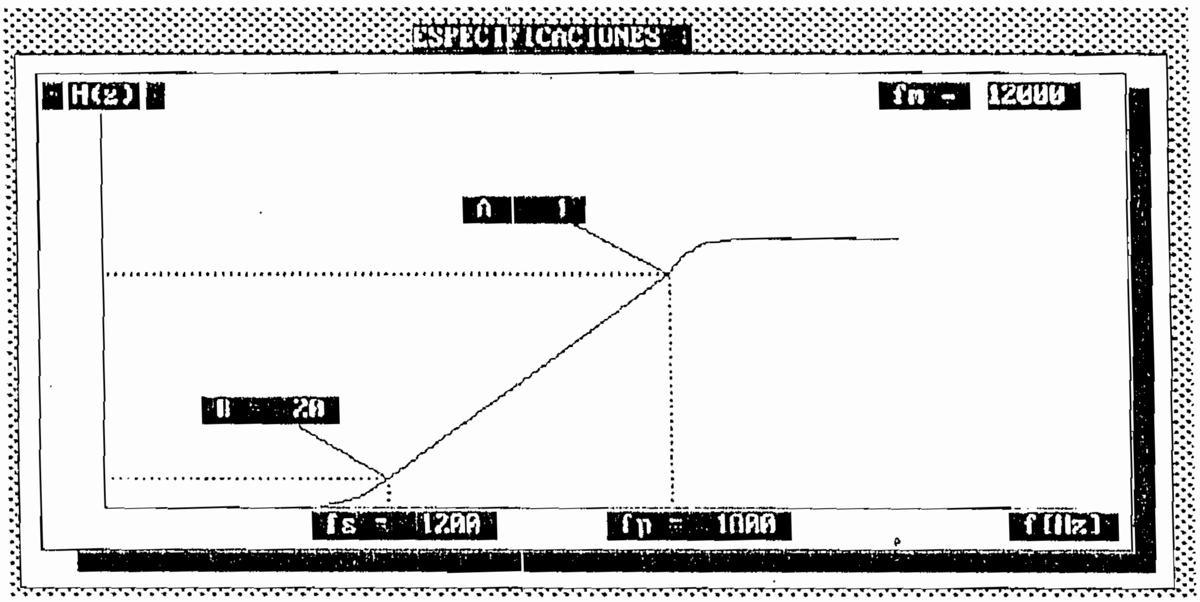
sobre la cual podemos movernos con las flechas, y cuando se haya elegido el tipo de banda para el filtro, presionaremos <ENTER>. A continuación aparecerá una pantalla para ingresar las especificaciones de la respuesta de frecuencia.

Dependiendo del caso ésta será:

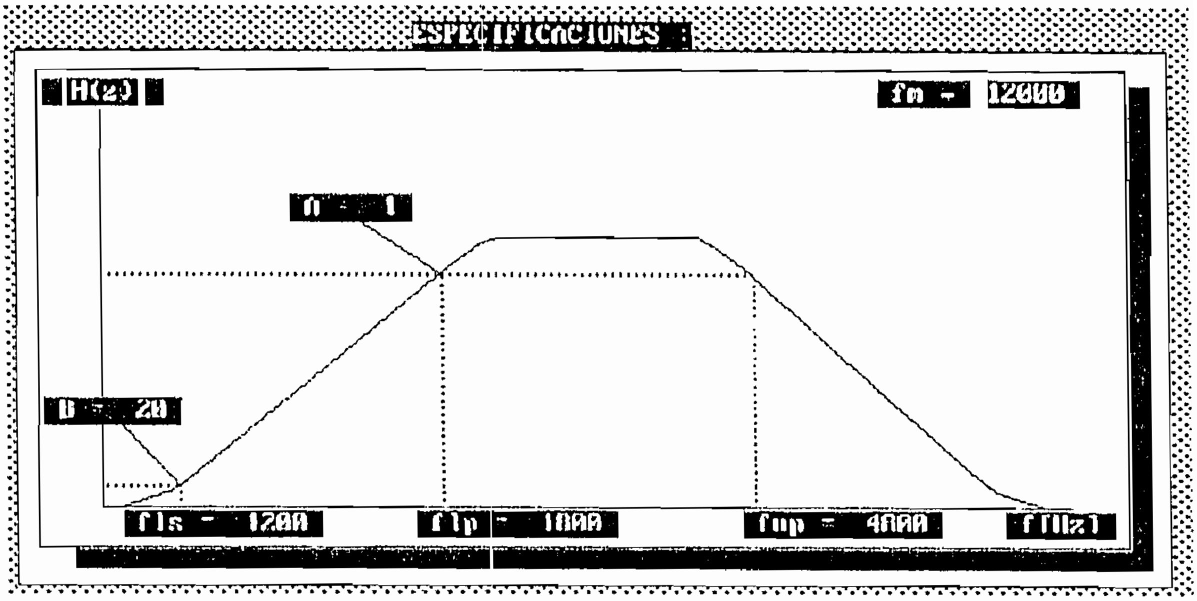
a) Para los filtros pasa-bajos:



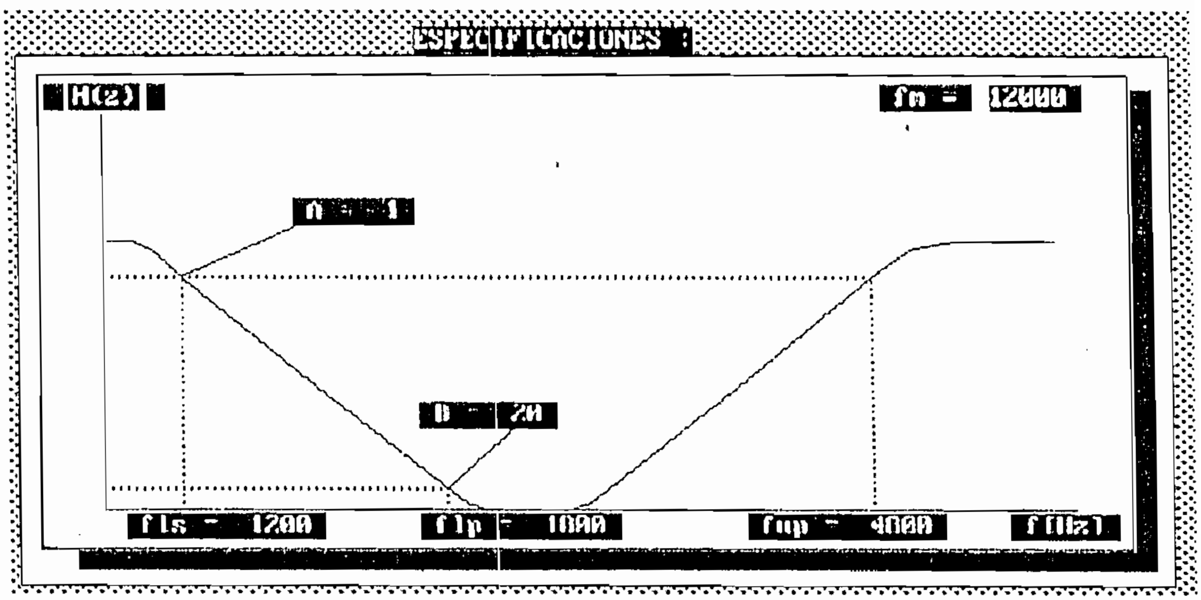
b) Para los filtros pasa-altos



c) Para los filtros pasa-banda



d) Para los filtros elimina-banda

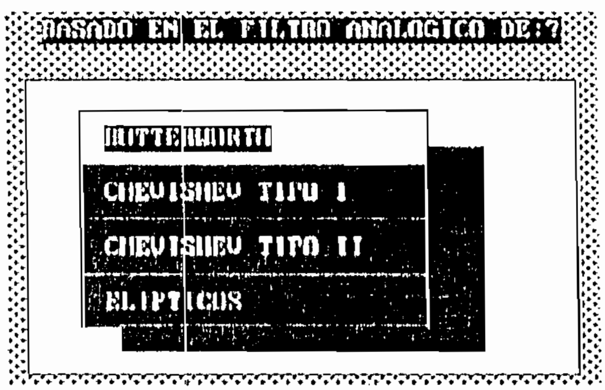


Inicialmente el cursor se encuentra en el lugar de la frecuencia de muestreo y está listo para ingresar el valor, si se desea. El cursor salta sobre cada uno de los valores que se indican, cada vez que se presiona <ENTER>. El ciclo es repetitivo, y la única manera de salir, es presionando <F10>. El recorrido del cursor inicia en la frecuencia de muestreo, sigue con las atenuaciones, y finaliza con las frecuencias de corte.

Para ingresar un valor se debe asegurar que el cursor se encuentre en la especificación adecuada, luego cuando se ingrese el primer número se borrará el valor anterior y se podrá ingresar el resto de números. Cuando se ha ingresado todo el valor se debe presionar <ENTER> para verificar si el dato ingresado es correcto. Si el valor es correcto este aparece en la pantalla, pero si no es correcto, en lugar de este, aparece el valor anterior.

3.2 <u>Opción ANALOGICO</u>

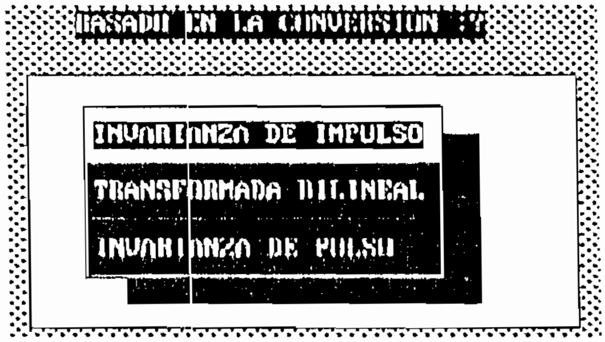
Esta opción se utiliza para especificar el tipo de filtro analógico a partir del cual se realizará el diseño del filtro digital. Al activar ésta, se despliega el siguiente sub-menú:



Por medio de las flechas podemos elegir cualquiera de esas cuatro opciones, y la escogida la aceptaremos con presionar <ENTER>. Inmediatamente aparecerá el menú principal.

3.3 Opción CONVERSION

Al activarse esta opción, se despliega el siguiente sub-menú:

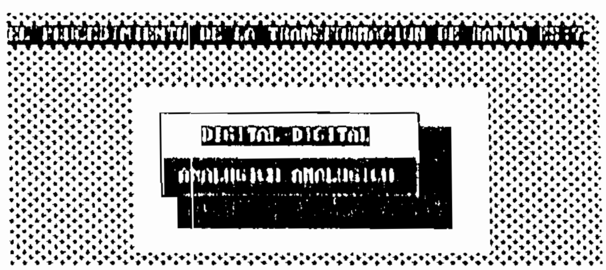


en este podemos seleccionar con las flechas, el método de conversión analógico a digital, que se empleará en el filtro a simularse. De la misma manera que el caso anterior al aceptar la selección realizada con <ENTER> aparecerá el menú principal.

3.4 Opción TRANSFORMACION

Esta opción no se podrá activar cuando el método de conversión analógico a digital, sea distinto de la Transformada Bilineal. En ese caso se pasa a la siguiente opción.

Cuando ya se active esta opción, aparece el siguiente submenú:

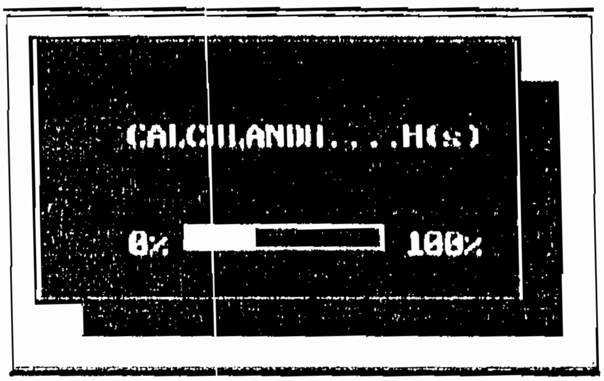


se escogerá con las flechas uno de los dos tipos de transformación de banda. Al aceptar presionando la tecla <ENTER>, se activará el menú principal. Si la selección fue por la transformación ANALOGICA-ANALOGICA, la opción CONVERSION queda deshabilitada; por lo tanto para seleccionar otro tipo de conversión que no sea la Transformación Bilineal, deberemos escoger la transformación DIGITAL-DIGITAL.

Inicialmente la opción TRANSFORMACION, no se puede activar, porque el programa inicializa con la Invariancia de Impulso como método de conversión.

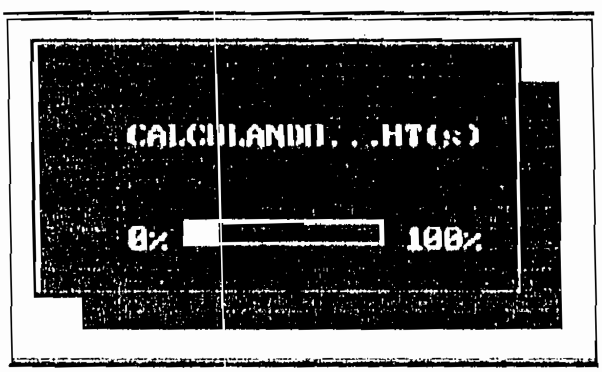
3.5 Opción SIMULACION

Esta opción como su nombre lo indica, es para simular el filtro. Inmediatamente se realizan los cálculos, empezando por la función de transferencia analógica. Para poder tener una idea de cuanto falta para terminar este cálculo, se presenta en la pantalla lo siguiente:

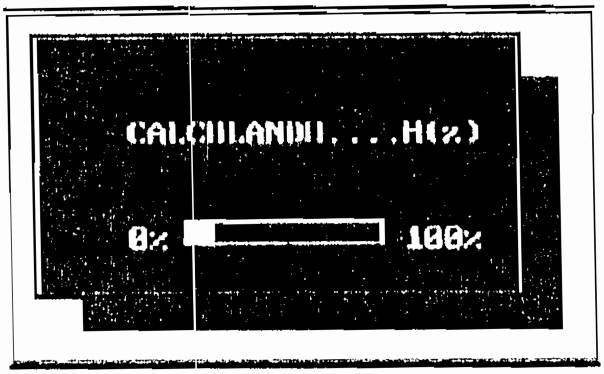


cuando la barra se llene totalmente, los cálculos terminarán.

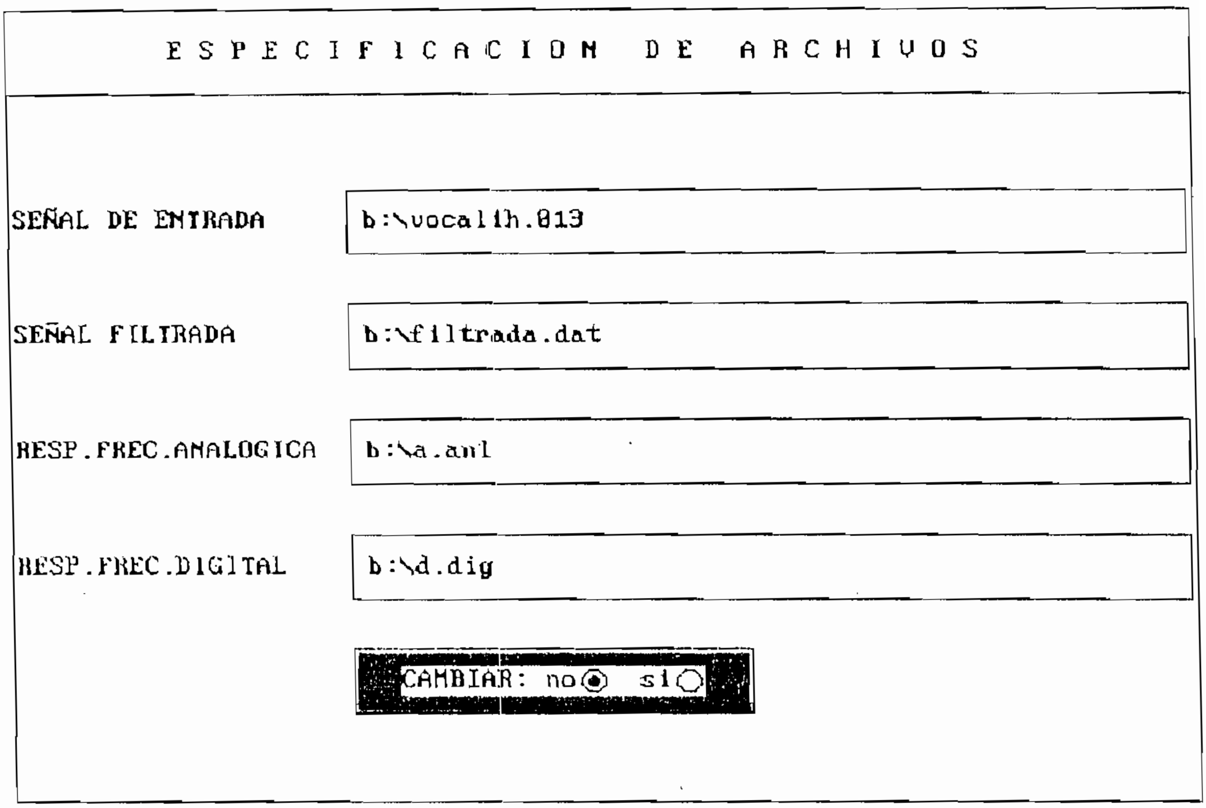
A continuación realiza el cálculo de la función de transferencia analógica con transformación en banda, si es que la transformación elegida es la analógica-analógica. En ese caso aparecerá:



Luego se calculará la función de transferencia digital, que se indica así:



En este momento ya tenemos el filtro simulado, y antes de efectuar el filtraje de la señal, se debe especificar la localización de archivos. Para esto aparece la siguiente pantalla:



los archivos allí presentados, son los que se utilizaron cuando se probó el sistema.

En la parte inferior hay un recuadro que pregunta si deseamos

CAMBIAR el nombre de los archivos indicados en pantalla. Con las <FLECHAS> elegiremos entre SI ó NO, y para aceptar deberemos presionar <ENTER>.

Cuando elegimos SI, el cursor se coloca al inicio del path del primer archivo, y esta listo para efectuar algún cambio en este. El programa reconoce archivos de entrada, cuyos datos sean representados por 12 bits, o que sean del formato *.wav.

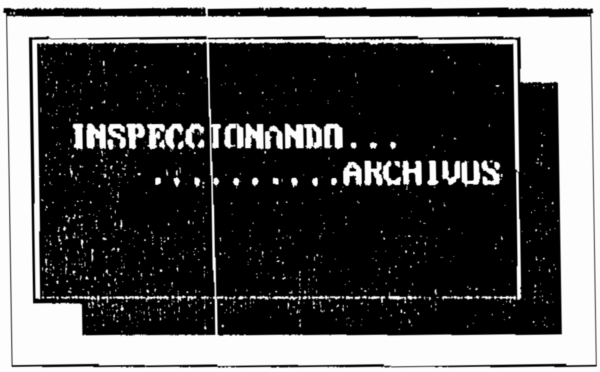
Todo archivo debe incluir su path.

En cualquier momento el cursor puede desplazarse al inicio de los otros archivos, con presionar <ENTER>. Se recomienda que se pase al lugar del siguiente archivo, sólo cuando se este seguro de haber ingresado el nombre correcto del archivo en curso.

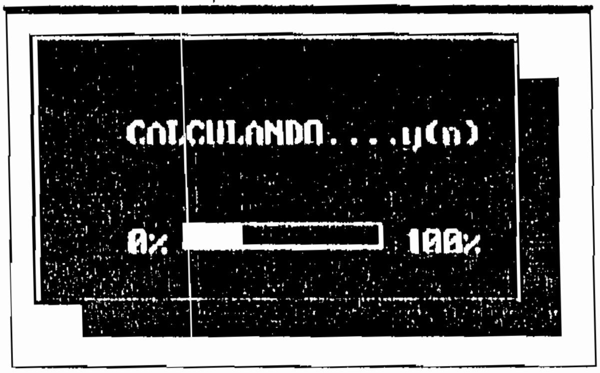
Al finalizar el ingreso de los archivos, aparecerá de nuevo el recuadro que pregunta si se desea CAMBIAR. Esto es con la finalidad de poder corregir alguna equivocación.

Si elegimos que NO, se procede a filtrar la señal. Para que pueda filtrarse una señal, cada dato de esta debe estar representado por 12 bits. Entonces :

- Si el archivo de entrada es del formato de 12 bits, el programa inspecciona el archivo, mientras que en la pantalla se muestra la siguiente indicación:



La inspección del archivo consiste en determinar el número de datos, y la amplitud máxima de la señal. Luego se procede a filtrar la señal con los datos del filtro actual simulado, mientras en la pantalla aparece la siguiente indicación :

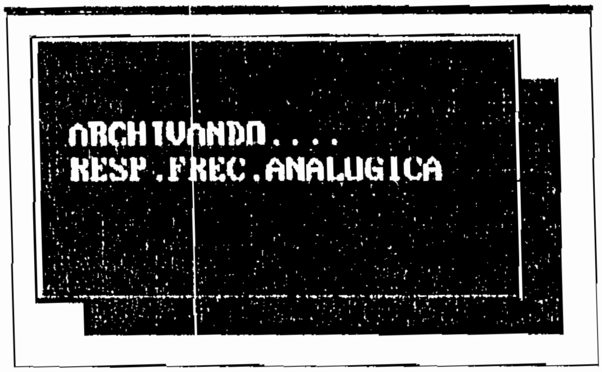


Los datos de la señal filtrada se almacenan en el respectivo archivo, y con el formato de 12 bits.

- Si el formato del archivo es *.wav, el programa debe convertir este archivo al formato de 12 bits para poder filtrar la señal. Para esto crea un nuevo archivo con el mismo nombre del *.wav, pero con la extensión "chs", donde se almacena la señal de entrada con el formato de 12 bits. Mientras se hace la conversión de formatos aparece en la pantalla lo siguiente:

Como la señal filtrada se almacena en el archivo con el formato de 12 bits, al finalizar el filtrado se convierte este archivo al formato *.wav. Para esto se crea un archivo, con el nombre del archivo de la señal filtrada y con extensión *.wav. De esta manera los resultados podrán ser correctamente utilizados.

- Cuando se ha especificado un archivo para la respuesta de frecuencia analógica, se almacenan los datos en este mientras en la pantalla se lee:



- Cuando se ha especificado un archivo para la respuesta de frecuencia analógica, se almacenan los datos en este mientras en la pantalla aparece la leyenda:

Finalmente aparecerá de nuevo el menú principal.

3.6 <u>Opción ARCHIVAR</u>

Al activar esta opción aparece la pantalla de especificación de archivos, mostrada en la opción anterior.

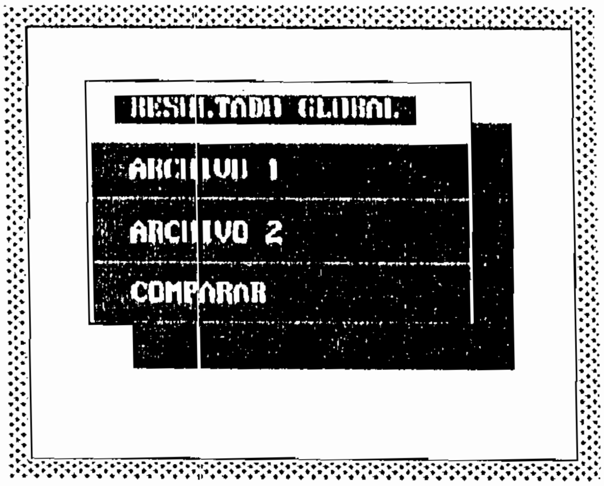
El propósito de esta opción es el de utilizar las características ya fijadas para el filtro, para filtrar cualquier señal adicional. Sólo cuando se requiera otro filtro, debemos utilizar la opción SIMULACION. Por esta razón esta opción no se la puede activar mientras no se haya simulado un filtro. Este caso se tiene al inicializar el sistema.

Para filtrar una señal, o para almacenar una respuesta de frecuencia, el programa debe detectar algún cambio en los archivos que las especifican.

El proceso de análisis de los archivos antes de filtrar la señal es el mismo que el utilizado en la opción anterior.

3.7 Opción GRAFICOS

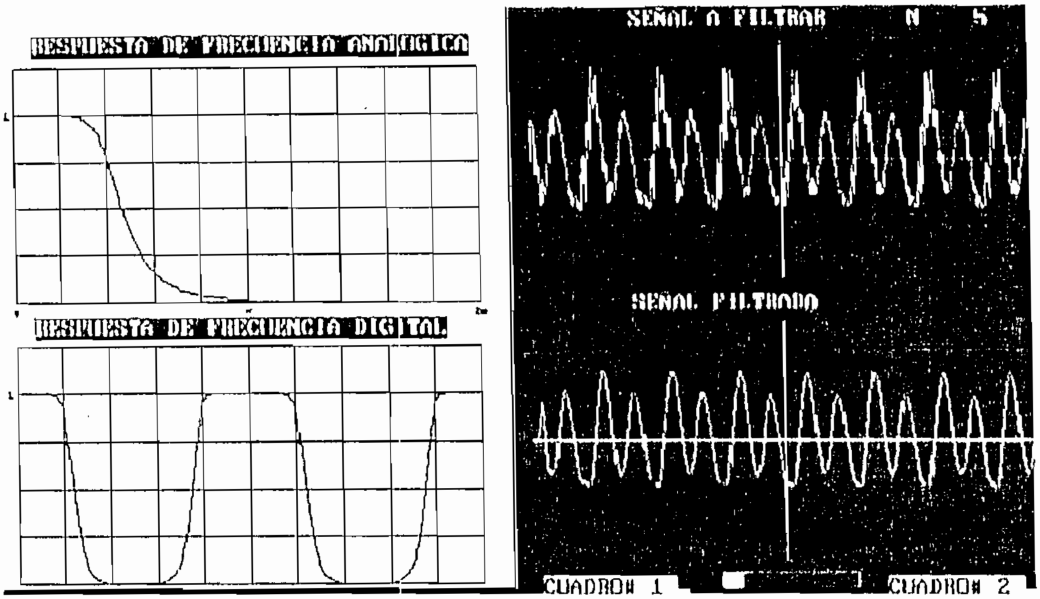
Al activar esta opción vamos a poder visualizar los resultados en la pantalla del computador. Primero se despliega el siguiente sub-menú:



Con éste podemos seleccionar la forma en que se mostrarán los resultados.

a) Al elegir RESULTADO GLOBAL, se podrá observar la respuesta de frecuencia analógica en color azul-brillante con fondo blanco, la respuesta de frecuencia digital en color rojo-brillante con fondo blanco, la señal antes y después de ser filtrada en color blanco con fondo negro.

Por ejemplo para un filtro dado, el RESULTADO GLOBAL es:

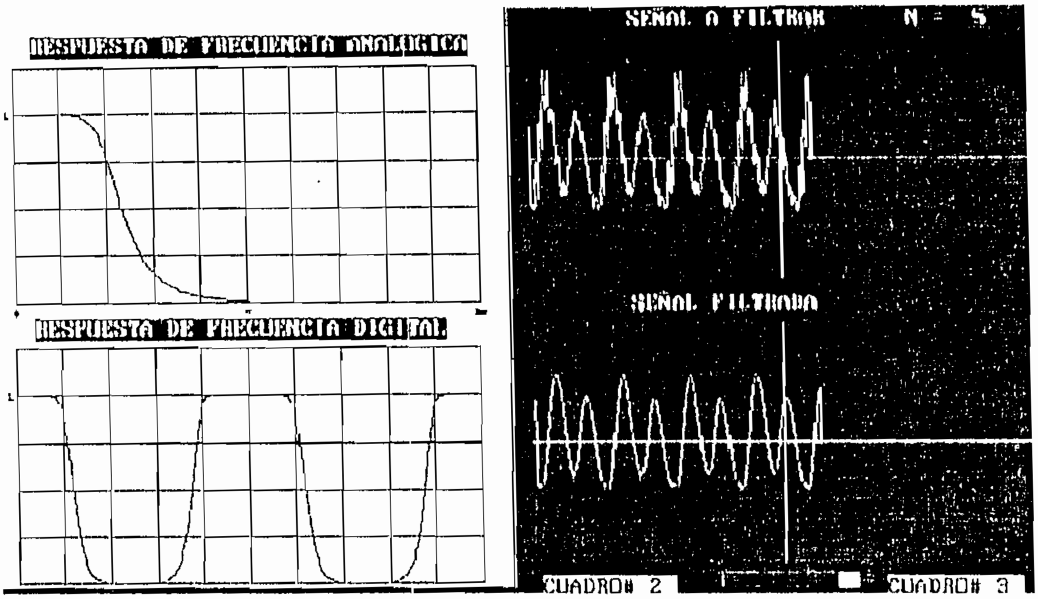


Cuando la transformación de banda es analógica-analógica, la respuesta de frecuencia del filtro analógico transformado aparece, en el mismo recuadro de la respuesta de frecuencia analógica, en color verde brillante.

En caso de que la señal sea de larga duración y no pueda presentarse en su totalidad en un solo recuadro, para observar otro tramo de la señales antes y después del filtrado, utilizamos las flechas. En la parte inferior de la señal filtrada, se divisa un rectángulo con un cuadrado brillante dentro. El rectángulo simboliza a toda la señal, y el cuadrado a la parte de la señal que se ve en este instante en pantalla. De esa manera podremos darnos cuenta, relativamente, en que posición se encuentra, la parte de la señal que se observa respecto de toda la señal.

Los recuadros donde se dibujan las señales, tienen un eje vertical de referencia que separa los dos cuadros.

A continuación vamos a observar el siguiente tramo de la señal utilizando la <FLECHA DERECHA>.



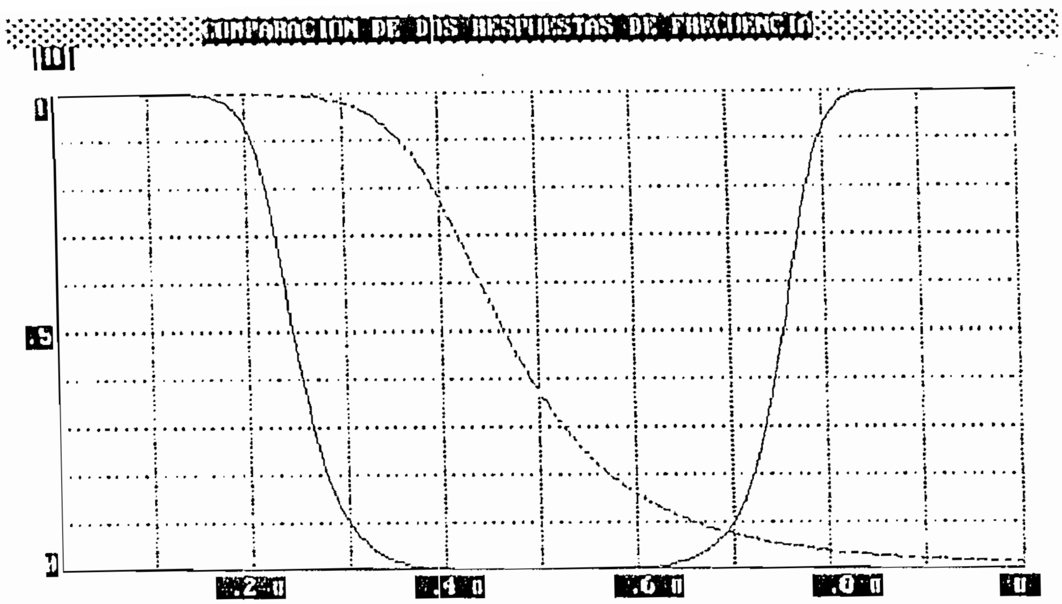
En el gráfico podemos ver que el CUADRO#1 ha desaparecido; el CUADRO#2 que estaba a la derecha del eje de referencia, se ha desplazado al lado izquierdo; y ha aparecido el CUADRO#3.

Las respuestas de frecuencia se encuentran dentro de un recuadro dividido en cuadrículas, donde cada una representa: 0.2π [rad] de frecuencia, y 0.25 del valor adimensional de la atenuación.

En la parte superior derecha, de la pantalla, se localiza el orden del filtro simulado.

Si presionamos <ENTER> aparecerá de nuevo el sub-menú de GRAFICOS, y si presionamos <F10> aparecerá el menú principal.

b) Al escoger la opción COMPARAR, se despliega una pantalla donde se encuentran las respuestas de frecuencia analógica y digital en forma más amplia y bajo el mismo recuadro, para facilitar su comparación. Un ejemplo de este tipo de pantalla se da a continuación:



Cada cuadrícula, representa 0.1π [rad], y 0.1 del valor adimensional de la atenuación.

El rango de visualización de esta pantalla es de π [rad] en el eje horizontal, y de una unidad del valor adimensional de la atenuación.

Si presionamos <ENTER> aparecerá de nuevo el sub-menú de GRAFICOS, y si presionamos <F10> aparecerá el menú principal.

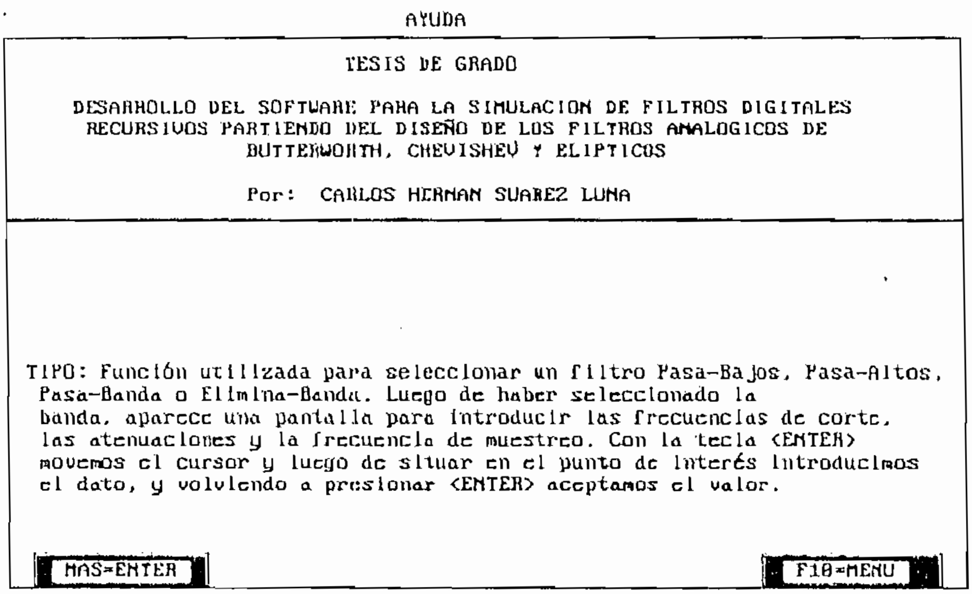
c) Al elegir ARCHIVO 1 o ARCHIVO 2, en la parte inferior del sub-menú se localiza el cursor, para poder ingresar el nombre de un archivo, correspondiente a una respuesta de frecuencia:

el objetivo es poder comparar dos respuestas de frecuencia, cualquiera, ingresando los nombres de los archivos que los contienen. Este archivo debe ser especificado, incluyendo el path. Cuando ya se acepten los archivos 1 y 2, activaremos la opción COMPARAR, y automáticamente aparecerán estas dos respuesta de frecuencia, en el mismo recuadro.

El eje horizontal de las respuestas de frecuencia representa [rad] si se trata de una respuesta digital y [rad/seg] si se trata de una respuesta analógica.

3.8 Opción AYUDA

Al seleccionar esta opción, se presenta una pantalla de información, en forma resumida, acerca del manejo elemental del programa. La pantalla donde se informa es:



Por el uso de colores, en la pantalla aquí vista no se han impreso algunas frases.

Si presionamos <ENTER> saldrá la segunda pantalla, y si presionamos <F10> estaremos de nuevo en el menú principal.

La segunda pantalla es:

AYUDA

CONVERSION: Para seleccionar el tipo de conversión de filtros analogicos en digitales.

SIMULACION: Esta opción efectúa la simulación del filtro, que consiste en calcular las funciones de transferencia analógica H(s) y digital H(z), y la señal filtrada y(n). Además se calculará la función analógica transformada HT(s), solo cuando la transformación sea Analógica-Analógica.

GRAFICOS: Permite observar los resultados.
RESULTADO GLOBAL, se ven las respuestas de frecuencias analógica y digital y la señal antes y después de filtrarla. En estas podemos elegir un tramo de observación, utilizando las flechas.
COMPARACION, para comparar dos respuestas de frecuencia que pueden ser especificadas en ARCHIVO 1 y ARCHIVO 2.

SALIR: Se usa para terminar la sesión de trabajo.

MAS=ENTER   F10=MENU

3.9 Opción SALIR

Esta opción sirve para abandonar el programa. Al final sale la siguiente pantalla de despedida.

FIN DE LA SESION

En general, cada vez que se ejecuta una opción, o se termina un trabajo, se produce un sonido para asegurar que la tarea fue ejecutada.

Cuando se ingresa un nombre de archivo incorrecto, el programa detecta el error, y hace aparecer la siguiente indicación:

Luego al presionar <ENTER> salimos al menú principal. Cuando el error se detecta en ARCHIVO 1 ó ARCHIVO 2 en la opción GRAFICOS, luego de presionar <ENTER> podemos solucionar el problema; si no deseamos corregir el error, presionando <F10> saldremos al menú.

Con las indicaciones dadas aquí creemos que es suficiente para que el usuario pueda operar el programa. Además en cada paso del programa se proporciona la información suficiente para manejarlo.

ANEXO B

LISTADO DEL PROGRAMA

ANEXO B

LISTADO DEL PROGRAMA

1. MODULOS

```
'MODULO PRINCIPAL:

'Declaración de un complejo

     TYPE complejo
          real AS DOUBLE
          imag AS DOUBLE
     END TYPE

' Dimensionamiento de variables
DIM SHARED aa(1 TO 200) AS complejo
DIM SHARED alf AS SINGLE
DIM SHARED alfa(1 TO 4, 1 TO 100)
DIM SHARED alfaso
DIM SHARED alfasola
DIM SHARED attp AS SINGLE
DIM SHARED atts AS SINGLE
DIM SHARED beta(0 TO 3, 1 TO 100)
DIM SHARED Bo
DIM SHARED c(1 TO 200)
DIM SHARED comp AS INTEGER
DIM SHARED errorabrir%
DIM SHARED fl, fm, fu, fup
DIM SHARED gama(1 TO 2, 1 TO 100)
DIM SHARED ha(1 TO 305)
DIM SHARED hat(1 TO 305)
DIM SHARED hb(1 TO 305)
DIM SHARED hi(1 TO 305)
DIM SHARED hp(1 TO 305)
DIM SHARED kanaloga, kbilineal
DIM SHARED kpulso(1 TO 200)
DIM SHARED kso, ksola
DIM SHARED l
DIM SHARED max, maximo
DIM SHARED men AS INTEGER
DIM SHARED n AS INTEGER
DIM SHARED nn AS INTEGER
DIM SHARED nreg, q
DIM SHARED p AS INTEGER
DIM SHARED s(1 TO 200) AS complejo
DIM SHARED sec AS INTEGER
DIM SHARED tip AS INTEGER
DIM SHARED tipana AS INTEGER
DIM SHARED tipconv AS INTEGER
DIM SHARED trans AS INTEGER
DIM SHARED u, wap, wac, was
```

```
DIM SHARED x(0 TO 5)
DIM SHARED y(0 TO 5, 0 TO 100)
DIM SHARED anal AS STRING
DIM SHARED digi AS STRING
DIM SHARED entr AS STRING
DIM SHARED sali AS STRING

'Declaración de subrutinas
DECLARE SUB archivo (ha(), hat(), hb(), hi(), hp(), nn AS
INTEGER, tipconv AS INTEGER, trans AS INTEGER, anal AS STRING,
digi AS STRING, entr AS STRING, sali AS STRING, men AS
INTEGER, errorabrir%, max, maximo, be%, bs%, ba%, bd%)
DECLARE SUB ayuda ()
DECLARE SUB bilineal ()
DECLARE SUB caratula (tip AS INTEGER, tipana AS INTEGER,
tipconv AS INTEGER, trans AS INTEGER)
DECLARE SUB caratulainicial ()
DECLARE SUB datos ()
DECLARE SUB dibujo ()
DECLARE SUB filtraje (max, maximo, nn AS INTEGER, entr AS
STRING, sali AS STRING, x(), y())
DECLARE SUB fparciales (aa() AS complejo, c(), kanaloga, n AS
INTEGER, s() AS complejo, sec AS INTEGER, tipana AS INTEGER)
DECLARE SUB frecuenciabilineal ()
DECLARE SUB frecuenciana (c(), gama(), ha(), kanaloga, p AS
INTEGER, sec AS INTEGER, tipana AS INTEGER)
DECLARE SUB frecuencianat ()
DECLARE SUB graficos ()
DECLARE SUB graficosmenu (comp AS INTEGER)
DECLARE SUB inpu (longitudt%, numero, palabra$, nol%, fila%,
                                                  columna%)
DECLARE SUB inspeccion (max, maximo, entr AS STRING)
DECLARE SUB instantaneo (tip AS INTEGER, tipana AS INTEGER,
                        tipconv AS INTEGER, trans AS INTEGER)
DECLARE SUB menu (men AS INTEGER)
DECLARE SUB orden ()
DECLARE SUB polos ()
DECLARE SUB porcentaje ()
DECLARE SUB salidabilineal ()
DECLARE SUB salidapulso ()
DECLARE SUB salidatotal ()
DECLARE SUB simulacion ()
DECLARE SUB tipo ()
DECLARE SUB tipoanalogico ()
DECLARE SUB tipoconversion ()
DECLARE SUB transformacion ()
DECLARE SUB transformacionbilineal (alfa(), alfaso, alfasola,
      beta(), c(), kanaloga, kbilineal, kso, l, n AS INTEGER,
      p AS INTEGER, s() AS complejo, sec AS INTEGER,
      tip AS INTEGER, tipana AS INTEGER, trans AS INTEGER, u)
DECLARE SUB transformacionimpulso (aa() AS complejo, alfa(),
      alfaso, alfasola, beta(), kpulso(), kso, p AS INTEGER,
             s() AS complejo, sec AS INTEGER, tip AS INTEGER)
DECLARE SUB transformacionpulso (aa() AS complejo, alfa(),
      alfaso, alfasola, beta(), kpulso(), kso, p AS INTEGER,
             s() AS complejo, sec AS INTEGER, tip AS INTEGER)
DECLARE SUB transfrecuenciabilineal (alfa(), beta(), hb(),
```

```
     kbilineal, p AS INTEGER, sec AS INTEGER,  tip AS INTEGER,
                                                tipana AS INTEGER)
DECLARE SUB transfrecuenciapulso (alfa(),  beta(), hi(), hp(),
         kanaloga, kpulso(), p AS INTEGER, sec AS INTEGER,
       tip AS INTEGER, tipana AS INTEGER, tipconv AS INTEGER)
DECLARE SUB valoresiniciales ()
DECLARE SUB  waps (alfaso, alfasola, Bo, fl, fm, fu, fup, kso,
                     ksola, tip AS  INTEGER, tipana AS INTEGER,
                                trans AS INTEGER, u, wap, was)

CONST pi = 3.141592654#
SCREEN 12
CLEAR , , 32769#
KEY(2) ON                               'Activación de la tecla F10
ON KEY(2) GOSUB instan                  'Utilización de la tecla F10
KEY(10) ON                              'Activación de la tecla F10
ON KEY(10) GOSUB menu                   'Utilización de la tecla F10
ON ERROR GOTO archivoerror

          CALL valoresiniciales
          CALL caratulainicial


          DO
          LOOP
DO
menu1:
KEY(10) ON
     SELECT CASE men
          CASE 1:
               CALL tipo
               CALL datos
          CASE 2:
               CALL tipoanalogico
          CASE 3:
               CALL tipoconversion
          CASE 4:
               CALL transformacion
          CASE 5:
               CALL simulacion
          CASE 6:
               CALL archivo(ha(), hat(), hb(), hi(),  hp(), nn,
                             tipconv, trans, anal,  digi, entr,
                                  sali,  men, errorabrir%, max,
                                   maximo, be%, bs%, ba%, bd%)
          CASE 7:
               CALL graficos
          CASE 8:
               CALL ayuda
          CASE 9:
               CLS
               LINE (151, 151)-(489, 329), , B
               LOCATE 15, 33
               PRINT "FIN DE LA SESION"
               RANDOMIZE TIMER
               FOR i = 1 TO 2400
                    PSET  (RND * 620 + 10, RND * 460 + 10),
```

```
                                                        INT(RND * 3) + 1
                    NEXT
                    END
          END SELECT
          CALL menu(men)
LOOP
menu:
          CLOSE (1)
          CLOSE (2)
          IF empezar% = 0 THEN
               CALL caratula(tip, tipana, tipconv, trans)
               empezar% = 1
          END IF
          CALL menu(men)
          KEY(10) OFF
          RETURN menu1

instan:
          CALL instantaneo(tip, tipana, tipconv, trans)
          RETURN
archivoerror:
          REDIM mensajerror(8000)
          GET (310, 160)-(550, 320), mensajerror
          CALL porcentaje
          LOCATE 14, 43
          PRINT "ERROR AL ABRIR EL "
          LOCATE 15, 43
          PRINT "ARCHIVO ..."
          LINE (360, 255)-(510, 275), 0, BF
          LOCATE 17, 52
          PRINT "< OK >"
          errorabrir% = -1
          DO
           v$ = INKEY$
          LOOP WHILE v$ <> CHR$(13)
          PUT (310, 160), mensajerror, PSET
          ERASE mensajerror
          RESUME NEXT



'SEGUNDO MODULO

'Declaración de Subrutinas

DECLARE SUB inspeccion (max!, maximo!, entr AS STRING)
DECLARE SUB porcentaje ()
DECLARE SUB salidatotal ()
DECLARE SUB inpu (longitudt%, numero!, palabra$, nol%, fila%,
                                                         columna%)
DECLARE SUB frecuenciabilineal ()

'Declaración de un complejo
     TYPE complejo
          real AS DOUBLE
          imag AS DOUBLE
     END TYPE
```

```
CONST pi = 3.141592654#


'TERCER MODULO

'Declaración de un complejo
     TYPE complejo
          real AS DOUBLE
          imag AS DOUBLE
     END TYPE

CONST pi = 3.141592654#
```

2. SUBRUTINAS

'Por facilitar la ubicación de una subrutina, las
'presentaremos de acuerdo a la clasificación realizada en el
'numeral 4.3.

a) SUBRUTINAS DE INGRESO DE DATOS

```
'Subrutina "tipo"


SUB tipo
 REDIM cuadro1(2000)
 REDIM cuadro2(2000)
 REDIM cuadro3(2000)
 REDIM cuadro4(2000)
 CALL caratula(tip, tipana, tipconv, trans)
     LINE (175, 130)-(490, 344), , BF
     LINE (176, 131)-(489, 343), 0, B
     LINE (193, 148)-(437, 306), 0, BF
     LINE (195, 150)-(435, 304), , B
     LINE (437, 170)-(463, 326), 0, BF
     LINE (215, 306)-(463, 326), 0, BF
     LINE (315, 150)-(315, 304)
     LINE (195, 225)-(435, 225)
     LOCATE 6, 23
     PRINT "QUE TIPO DE FILTRO DESEA CONSTRUIR ?"

  'Dibujo del filtro pasa-bajos
     LINE (210, 160)-(210, 205)
     LINE -(300, 205)
     LINE (210, 170)-(230, 170)
     LINE -(235, 175)
     LINE -(240, 201)
     LINE -(245, 202)
     LINE -(270, 204)
     LINE -(290, 205)
```

```
   LOCATE 14, 28
   PRINT "Pasa-bajos"

'Dibujo del filtro pasa-altos
   LINE (330, 160)-(330, 205)
   LINE -(420, 205)
   LINE (410, 170)-(390, 170)
   LINE -(385, 175)
   LINE -(380, 201)
   LINE -(375, 202)
   LINE -(350, 204)
   LINE -(330, 205)
   LOCATE 14, 42
   PRINT "Pasa-altos"

'Dibujo del filtro pasa-banda
   LINE (210, 241)-(210, 286)
   LINE -(300, 286)
   LINE (210, 286)-(225, 284)
   LINE -(230, 281)
   LINE -(235, 256)
   LINE -(240, 251)
   LINE -(250, 251)
   LINE -(255, 256)
   LINE -(260, 281)
   LINE -(265, 284)
   LINE -(290, 286)
   LOCATE 19, 28
   PRINT "Pasa-banda"

'Dibujo del filtro elimina-banda
   LINE (330, 241)-(330, 286)
   LINE -(420, 286)
   LINE (330, 251)-(345, 251)
   LINE -(348, 253)
   LINE -(352, 281)
   LINE -(355, 284)
   LINE -(360, 286)
   LINE -(370, 286)
   LINE -(375, 284)
   LINE -(378, 281)
   LINE -(382, 253)
   LINE -(385, 251)
   LINE -(400, 251)
   LOCATE 19, 42
   PRINT "Elimina-banda"

GET (197, 152)-(313, 223), cuadro1
GET (317, 152)-(433, 223), cuadro2
GET (197, 227)-(313, 302), cuadro3
GET (317, 227)-(433, 302), cuadro4

     tipi% = tip

 DO
 SELECT CASE tipi%
 CASE 1:
```

```
    IF anteriortip% = 2 THEN
        PUT (317, 152), cuadro2, PSET
    ELSEIF anteriortip% = 4 THEN
        PUT (197, 227), cuadro3, PSET
    END IF
    anteriortip% = 1
        PUT (197, 152), cuadro1, PRESET
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipi% = 4
    ELSEIF RIGHT$(v$, 1) = CHR$(77) THEN
        tipi% = 2
    END IF
CASE 2:
    IF anteriortip% = 1 THEN
        PUT (197, 152), cuadro1, PSET
    ELSEIF anteriortip% = 3 THEN
        PUT (317, 227), cuadro4, PSET
    END IF
    anteriortip% = 2
        PUT (317, 152), cuadro2, PRESET
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipi% = 3
    ELSEIF RIGHT$(v$, 1) = CHR$(75) THEN
        tipi% = 1
    END IF
CASE 3:
    IF anteriortip% = 2 THEN
        PUT (317, 152), cuadro2, PSET
    ELSEIF anteriortip% = 4 THEN
        PUT (197, 227), cuadro3, PSET
    END IF
    anteriortip% = 3
        PUT (317, 227), cuadro4, PRESET
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""
    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipi% = 2
    ELSEIF RIGHT$(v$, 1) = CHR$(75) THEN
        tipi% = 4
    END IF
CASE 4:
    IF anteriortip% = 1 THEN
        PUT (197, 152), cuadro1, PSET
    ELSEIF anteriortip% = 3 THEN
        PUT (317, 227), cuadro4, PSET
    END IF
    anteriortip% = 4
        PUT (197, 227), cuadro3, PRESET
    DO
    v$ = INKEY$
```

```
    LOOP WHILE v$ = ""
    IF RIGHT$(v$, 1) = CHR$(72) THEN
         tipi% = 1
    ELSEIF RIGHT$(v$, 1) = CHR$(77) THEN
         tipi% = 3
    END IF
  END SELECT
  LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
         tip = tipi%
  ERASE cuadro1
  ERASE cuadro2
  ERASE cuadro3
  ERASE cuadro4

END SUB


'Subrutina "datos"


SUB datos
CALL caratula(tip, tipana, tipconv, trans)
LINE (7, 448)-(200, 464), 0, BF
LOCATE 6, 30
PRINT "ESPECIFICACIONES :"
LINE (25, 95)-(620, 395), , BF
LINE (26, 96)-(619, 394), 0, B
LINE (55, 375)-(610, 385), 0, BF
LINE (600, 115)-(610, 385), 0, BF

LINE (37, 107)-(598, 373), 0, B
LINE (70, 130)-(70, 350), 0
LINE -(570, 350), 0
LOCATE 8, 6
PRINT " |H(z)| "
LOCATE 23, 68
PRINT " f[Hz] "

   SELECT CASE tip
      CASE 1:
              LINE (70, 200)-(180, 200), 0
              LINE -(188, 201), 0
              LINE -(195, 204), 0
              LINE -(210, 215), 0
              LINE -(350, 325), 0
              LINE -(365, 336), 0
              LINE -(372, 341), 0
              LINE -(380, 345), 0
              LINE -(405, 350), 0
         FOR i = 1 TO 36
              PSET (i * 4 + 70, 220), 0
         NEXT
         FOR i = 1 TO 32
              PSET (216, 350 - i * 4), 0
         NEXT
         FOR i = 1 TO 73
              PSET (i * 4 + 68, 334), 0
```

```
    NEXT
    FOR i = 1 TO 3
          PSET (362, 350 - i * 4), 0
    NEXT
        LOCATE 12, 33
        PRINT " A = "; attp
        LOCATE 19, 52
        PRINT " B = "; atts
        LINE (216, 220)-(255, 190), 0
        LINE (362, 334)-(408, 303), 0
        LOCATE 23, 23
        PRINT " fp = "; fl
        LOCATE 23, 42
        PRINT " fs = "; fu
CASE 2:
        LINE (180, 350)-(200, 345), 0
        LINE -(216, 334), 0
        LINE -(362, 220), 0
        LINE (361, 220)-(369, 210), 0
        LINE -(380, 203), 0
        LINE -(400, 200), 0
        LINE -(480, 200), 0
    FOR i = 1 TO 36
          PSET (i * 4 + 70, 334), 0
    NEXT
    FOR i = 1 TO 3
          PSET (216, 350 - i * 4), 0
    NEXT
    FOR i = 1 TO 73
          PSET (i * 4 + 68, 220), 0
    NEXT
    FOR i = 1 TO 32
          PSET (362, 350 - i * 4), 0
    NEXT
        LOCATE 12, 33
        PRINT " A = "; attp
        LOCATE 19, 16
        PRINT " B = "; atts
        LINE (216, 334)-(170, 305), 0
        LINE (362, 220)-(310, 190), 0
        LOCATE 23, 23
        PRINT " fs = "; fl
        LOCATE 23, 42
        PRINT " fp = "; fu
CASE 3:
        LINE (70, 200)-(85, 200), 0
        LINE -(95, 205), 0
        LINE -(110, 220), 0
        LINE -(256, 346), 0
        LINE -(266, 350), 0
        LINE (310, 350)-(320, 346), 0
        LINE -(466, 220), 0
        LINE -(486, 205), 0
        LINE -(510, 200), 0
        LINE -(560, 200), 0
    FOR i = 1 TO 99
          PSET (i * 4 + 70, 220), 0
```

```
    NEXT
    FOR i = 1 TO 32
            PSET (466, 350 - i * 4), 0
    NEXT
    FOR i = 1 TO 32
            PSET (110, 350 - i * 4), 0
    NEXT
    FOR i = 1 TO 44
            PSET (i * 4 + 70, 338), 0
    NEXT
    FOR i = 1 TO 3
            PSET (246, 350 - i * 4), 0
    NEXT
    IF fup <= (2 * fu - fl) THEN
            fup = 2 * fu - fl + 1
    END IF
         LOCATE 23, 11
          PRINT " fls = "; fl
         LOCATE 23, 30
          PRINT " flp = "; fu
         LOCATE 23, 51
          PRINT " fup = "; fup
         LOCATE 12, 22
          PRINT " A = "; attp
         LOCATE 19, 30
          PRINT " B = "; atts
         LINE (246, 338)-(280, 304), 0
         LINE (110, 220)-(170, 192), 0

CASE 4:
         LINE (280, 200)-(272, 201), 0
         LINE -(265, 204), 0
         LINE -(250, 215), 0
         LINE -(110, 338), 0
         LINE -(103, 342), 0
         LINE -(95, 345), 0
         LINE -(80, 350), 0
         LINE (280, 200)-(378, 200), 0
         LINE -(385, 204), 0
         LINE -(400, 215), 0
         LINE -(527, 338), 0
         LINE -(534, 342), 0
         LINE -(542, 345), 0
         LINE -(557, 350), 0
     FOR i = 1 TO 84
            PSET (i * 4 + 70, 220), 0
     NEXT
     FOR i = 1 TO 32
            PSET (407, 350 - i * 4), 0
     NEXT
     FOR i = 1 TO 32
            PSET (246, 350 - i * 4), 0
     NEXT
     FOR i = 1 TO 10
            PSET (i * 4 + 68, 338), 0
     NEXT
     FOR i = 1 TO 3
```

```
                PSET (110, 350 - i * 4), 0
        NEXT
            LOCATE 23, 11
             PRINT " fls = "; fl
            LOCATE 23, 30
             PRINT " flp = "; fu
            LOCATE 23, 51
             PRINT " fup = "; fup
            LOCATE 12, 22
             PRINT " A = "; attp
            LOCATE 19, 6
             PRINT " B = "; atts
            LINE (246, 220)-(205, 192), 0
            LINE (110, 338)-(80, 304), 0
END SELECT
        LOCATE 8, 60
         PRINT " fm = "; fm
DO
        CALL inpu(5, prefm, palabra$, 0, 8, 66)
        IF (0 < prefm AND prefm > fu * 2) THEN
             fm = prefm
        END IF
        LOCATE 8, 66
        PRINT fm

 SELECT CASE tip
 CASE 1, 2:
        CALL inpu(3, preattp, palabra$, 0, 12, 38)
        IF (0 < preattp AND preattp < atts) THEN
          attp = preattp
        END IF
        LOCATE 12, 38
        LINE (300, 191)-(400, 170), , BF
        PRINT attp

    IF tip = 1 THEN

        CALL inpu(3, preatts, palabra$, 0, 19, 57)
        IF (0 < preatts AND preatts > attp) THEN
             atts = preatts
        END IF
        LOCATE 19, 57
        LINE (460, 305)-(560, 270), , BF
        PRINT atts
    ELSE
        CALL inpu(3, preatts, palabra$, 0, 19, 21)
        IF (0 < preatts AND preatts > attp) THEN
             atts = preatts
        END IF
        LOCATE 19, 21
        LINE (170, 305)-(250, 285), , BF
        PRINT atts
    END IF

        CALL inpu(5, prefl, palabra$, 0, 23, 29)
        IF (0 < prefl AND prefl < fu) THEN
             fl = prefl
```

```
        END IF
        LOCATE 23, 29
        LINE (250, 351)-(330, 370), , BF
        PRINT fl

        CALL inpu(5, prefu, palabra$, 0, 23, 48)
        IF (fl < prefu AND 2 * prefu < fm) THEN
             fu = prefu
        END IF
        LOCATE 23, 48
        LINE (410, 351)-(500, 370), , BF
        PRINT fu

CASE 3, 4:

        CALL inpu(3, preattp, palabra$, 0, 12, 27)
        IF (0 < preattp AND preattp < atts) THEN
          attp = preattp
        END IF
        LOCATE 12, 27
        LINE (230, 191)-(340, 170), , BF
        PRINT attp

    IF tip = 4 THEN

        CALL inpu(3, preatts, palabra$, 0, 19, 11)
        IF (0 < preatts AND preatts > attp) THEN
             atts = preatts
        END IF
        LOCATE 19, 11
        LINE (98, 303)-(148, 286), , BF
        PRINT atts

        CALL inpu(5, prefl, palabra$, 0, 23, 18)
        IF (0 < prefl AND prefl < fu) THEN
             fl = prefl
        END IF
        LOCATE 23, 18
        LINE (153, 351)-(234, 370), , BF
        PRINT fl

        CALL inpu(5, prefu, palabra$, 0, 23, 37)
        IF (fl < prefu AND prefu < fup) THEN
             fu = prefu
        END IF
        LOCATE 23, 37
        LINE (307, 351)-(396, 370), , BF
        PRINT fu

        CALL inpu(5, prefup, palabra$, 0, 23, 58)
        IF (fu < prefup AND 2 * prefup < fm) THEN
             fup = prefup
        END IF
        LOCATE 23, 58
        LINE (472, 351)-(538, 370), , BF
        PRINT fup
    ELSE
```

```
            CALL inpu(3, preatts, palabra$, 0, 19, 35)
            IF (0 < preatts AND preatts > attp) THEN
                 atts = preatts
            END IF
            LOCATE 19, 35
            LINE (290, 303)-(360, 286), , BF
            PRINT atts

            CALL inpu(5, prefl, palabra$, 0, 23, 18)
            IF (0 < prefl AND prefl < fu AND
                                      fup > (2 * fu - prefl)) THEN
                 fl = prefl
            END IF
            LOCATE 23, 18
            LINE (153, 351)-(234, 370), , BF
            PRINT fl

            CALL inpu(5, prefu, palabra$, 0, 23, 37)
            IF (fl < prefu AND prefu < fup AND
                                      fup > (2 * prefu - fl)) THEN
                 fu = prefu
            END IF
            LOCATE 23, 37
            LINE (307, 351)-(396, 370), , BF
            PRINT fu

            CALL inpu(5, prefup, palabra$, 0, 23, 58)
            IF (fu < prefup AND 2 * prefup < fm AND
                                      prefup > (2 * fu - fl)) THEN
                 fup = prefup
            END IF
            LOCATE 23, 58
            LINE (472, 351)-(538, 370), , BF
            PRINT fup
        END IF
     END SELECT
  LOOP
END SUB


'Subrutina "tipoanalogico"


SUB tipoanalogico
     CALL caratula(tip, tipana, tipconv, trans)
          LINE (190, 180)-(480, 355), , BF
          LINE (191, 181)-(479, 354), 0, B
          LOCATE 10, 26
          PRINT "BASADO EN EL FILTRO ANALOGICO DE:?"
          LINE (219, 199)-(401, 326), 0, BF
          LINE (240, 220)-(430, 340), 0, BF
          LINE (220, 200)-(400, 230), , B
          LINE -(220, 263), , B
          LINE -(400, 295), , B
          LINE -(220, 325), , B
          LOCATE 14, 30
```

```
        PRINT "BUTTERWORTH"
        LOCATE 16, 30
        PRINT "CHEVISHEV TIPO I"
        LOCATE 18, 30
        PRINT "CHEVISHEV TIPO II"
        LOCATE 20, 30
        PRINT "ELIPTICOS"
tipanal% = tipana
DO
SELECT CASE tipanal%
CASE 1:
    IF anteriortipanal% = 4 THEN
         LINE (399, 296)-(221, 324), 0, BF
         LOCATE 20, 30
         PRINT "ELIPTICOS"
      END IF
    IF anteriortipanal% = 2 THEN
         LINE (399, 231)-(221, 262), 0, BF
         LOCATE 16, 30
         PRINT "CHEVISHEV TIPO I"
      END IF

    anteriortipanal% = 1
    LINE (220, 200)-(400, 230), , BF
    LOCATE 14, 30
    PRINT "BUTTERWORTH"
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipanal% = 4
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipanal% = 2
    END IF

 CASE 2:
    IF anteriortipanal% = 1 THEN
         LINE (221, 201)-(399, 229), 0, BF
         LOCATE 14, 30
         PRINT "BUTTERWORTH"
      END IF
    IF anteriortipanal% = 3 THEN
         LINE (221, 264)-(399, 294), 0, BF
         LOCATE 18, 30
         PRINT "CHEVISHEV TIPO II"
      END IF

    anteriortipanal% = 2

    LINE (400, 230)-(220, 263), , BF
    LOCATE 16, 30
    PRINT "CHEVISHEV TIPO I"
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""
```

```
    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipanal% = 1
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipanal% = 3
    END IF

CASE 3:
    IF anteriortipanal% = 4 THEN
         LINE (399, 296)-(221, 324), 0, BF
         LOCATE 20, 30
         PRINT "ELIPTICOS"
      END IF
    IF anteriortipanal% = 2 THEN
         LINE (399, 231)-(221, 262), 0, BF
         LOCATE 16, 30
         PRINT "CHEVISHEV TIPO I"
      END IF

    anteriortipanal% = 3
    LINE (220, 263)-(400, 295), , BF
    LOCATE 18, 30
    PRINT "CHEVISHEV TIPO II"
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipanal% = 2
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipanal% = 4
    END IF

CASE 4:
    IF anteriortipanal% = 1 THEN
         LINE (221, 201)-(399, 229), 0, BF
         LOCATE 14, 30
         PRINT "BUTTERWORTH"
      END IF
    IF anteriortipanal% = 3 THEN
         LINE (221, 264)-(399, 294), 0, BF
         LOCATE 18, 30
         PRINT "CHEVISHEV TIPO II"
      END IF

    anteriortipanal% = 4
    LINE (400, 295)-(220, 325), , BF
    LOCATE 20, 30
    PRINT "ELIPTICOS"
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipanal% = 3
```

```
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipanal% = 1
    END IF

 END SELECT

 LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
 tipana = tipanal%

END SUB



'Subrutina "tipoconversion"


SUB tipoconversion
 IF trans <> 1 THEN
     CALL caratula(tip, tipana, tipconv, trans)
          LINE (190, 180)-(470, 325), , BF
          LINE (191, 181)-(469, 324), 0, B
          LINE (219, 199)-(401, 296), 0, BF
          LINE (240, 215)-(420, 310), 0, BF
          LOCATE 10, 28
          PRINT "BASADO EN LA CONVERSION :?"
          LINE (220, 200)-(400, 230), , B
          LINE -(220, 263), , B
          LINE -(400, 295), , B
          LOCATE 14, 29
          PRINT "INVARIANZA DE IMPULSO"
          LOCATE 16, 29
          PRINT "TRANSFORMADA BILINEAL"
          LOCATE 18, 29
          PRINT "INVARIANZA DE PULSO"
  tipconve% = tipconv
  DO
   SELECT CASE tipconve%
   CASE 1:
      IF anteriortipconve% = 3 THEN
           LINE (221, 264)-(399, 294), 0, BF
           LOCATE 18, 29
           PRINT "INVARIANZA DE PULSO"
        END IF
      IF anteriortipconve% = 2 THEN
           LINE (399, 231)-(221, 262), 0, BF
           LOCATE 16, 29
           PRINT "TRANSFORMADA BILINEAL"
        END IF
      anteriortipconve% = 1
      LINE (220, 200)-(400, 230), , BF
      LOCATE 14, 29
      PRINT "INVARIANZA DE IMPULSO"
      DO
      v$ = INKEY$
      LOOP WHILE v$ = ""
```

```
   IF RIGHT$(v$, 1) = CHR$(72) THEN
       tipconve% = 3
   END IF
   IF RIGHT$(v$, 1) = CHR$(80) THEN
       tipconve% = 2
   END IF

CASE 2:

   IF anteriortipconve% = 1 THEN
        LINE (221, 201)-(399, 229), 0, BF
        LOCATE 14, 29
        PRINT "INVARIANZA DE IMPULSO"
     END IF
   IF anteriortipconve% = 3 THEN
        LINE (221, 264)-(399, 294), 0, BF
        LOCATE 18, 29
        PRINT "INVARIANZA DE PULSO"
     END IF

   anteriortipconve% = 2

   LINE (400, 230)-(220, 263), , BF
   LOCATE 16, 29
   PRINT "TRANSFORMADA BILINEAL"
   DO
   v$ = INKEY$
   LOOP WHILE v$ = ""

   IF RIGHT$(v$, 1) = CHR$(72) THEN
       tipconve% = 1
   END IF
   IF RIGHT$(v$, 1) = CHR$(80) THEN
       tipconve% = 3
   END IF

CASE 3:

   IF anteriortipconve% = 1 THEN
        LINE (221, 201)-(399, 229), 0, BF
        LOCATE 14, 29
        PRINT "INVARIANZA DE IMPULSO"
     END IF
   IF anteriortipconve% = 2 THEN
        LINE (399, 231)-(221, 262), 0, BF
        LOCATE 16, 29
        PRINT "TRANSFORMADA BILINEAL"
     END IF

   anteriortipconve% = 3
   LINE (220, 263)-(400, 295), , BF
   LOCATE 18, 29
   PRINT "INVARIANZA DE PULSO"
   DO
   v$ = INKEY$
   LOOP WHILE v$ = ""
```

```
    IF RIGHT$(v$, 1) = CHR$(72) THEN
        tipconve% = 2
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        tipconve% = 1
    END IF

  END SELECT

 LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
 tipconv = tipconve%
 END IF

END SUB



´Subrutina "transformacion"



SUB transformacion
 IF tipconv = 2 THEN
     CALL caratula(tip, tipana, tipconv, trans)
          LOCATE 11, 15
          PRINT "EL PROCEDIMIENTO DE LA TRANSFORMACION DE
                                                  BANDA ES:?"
          LINE (200, 210)-(450, 340), , BF
          LINE (219, 229)-(401, 296), 0, BF
          LINE (230, 240)-(425, 320), 0, BF
          LINE (400, 230)-(220, 263), , B
          LINE (220, 263)-(400, 295), , B
          LOCATE 16, 32
          PRINT "DIGITAL-DIGITAL"
          LOCATE 18, 30
          PRINT "ANALOGICO-ANALOGICO"
    transf% = trans
   DO
   SELECT CASE transf%
   CASE 1:
      LINE (399, 231)-(221, 262), 0, BF
      LOCATE 16, 32
      PRINT "DIGITAL-DIGITAL"

      LINE (220, 263)-(400, 295), , BF
      LOCATE 18, 30
      PRINT "ANALOGICO-ANALOGICO"

      DO
      v$ = INKEY$
      LOOP WHILE v$ = ""

      IF (RIGHT$(v$, 1) = CHR$(72) OR
                              RIGHT$(v$, 1) = CHR$(80)) THEN
          transf% = 2
      END IF
```

```
 CASE 2:
    LINE (221, 264)-(399, 294), 0, BF
    LOCATE 18, 30
    PRINT "ANALOGICO-ANALOGICO"

    LINE (400, 230)-(220, 263), , BF
    LOCATE 16, 32
    PRINT "DIGITAL-DIGITAL"

    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF (RIGHT$(v$, 1) = CHR$(72) OR
                                  RIGHT$(v$, 1) = CHR$(80)) THEN
        transf% = 1
    END IF
  END SELECT
  LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
  trans = transf%
 END IF
END SUB



´Subrutina "archivo"


SUB archivo (ha(), hat(), hb(), hi(), hp(), nn  AS INTEGER,
                    tipconv  AS INTEGER, trans AS INTEGER,
             anal AS STRING, digi AS STRING, entr AS STRING,
             sali  AS STRING, men AS INTEGER, errorabrir%, max,
                                maximo, be%, bs%, ba%, bd%)

 IF hi(303) <> 0 OR hb(303) <> 0 OR hp(303) <> 0 THEN

  DIM wa AS STRING * 2
  DIM prewa AS STRING * 1
  DIM ra AS STRING * 2
  DIM rd AS STRING * 2

CLS
LINE (0, 0)-(639, 479), , B
PAINT (1, 1), 7, 15
LINE (0, 48)-(639, 48)
LOCATE 2, 15
PRINT "E S P E C I F I C A C I O N   D E   A R C H I V O S"
FOR i = 1 TO 5
      LINE (0, 479)-(639, 480 - i * 4)
      SOUND 440 + 44 * i, 1
NEXT
LOCATE 29, 27
PRINT "<ENTER=Aceptar>";
LOCATE 29, 46
PRINT "<F2=Características>";
 IF men = 6 THEN
    LOCATE 29, 70
```

```
    PRINT "<F10=Menú>";
 END IF
FOR i = 0 TO 3
 LINE (200,  (6 + 4 *  i) * 16 +  6)-(620, (8 + 4  * i)  * 16 +
10), , B
NEXT
 blanquear$ = SPACE$(50)
 FOR i = 0 TO 3
   LOCATE 8 + 4 * i, 27
   PRINT blanquear$
 NEXT
LOCATE 8, 5
PRINT "SEÑAL DE ENTRADA"
LOCATE 8, 27
PRINT entr
LOCATE 12, 5
PRINT "SEÑAL FILTRADA"
LOCATE 12, 27
PRINT sali
LOCATE 16, 5
PRINT "RESP.FREC.ANALOGICA"
LOCATE 16, 27
PRINT anal
LOCATE 20, 5
PRINT "RESP.FREC.DIGITAL"
LOCATE 20, 27
PRINT digi
   preentr$ = entr
   presali$ = sali
   preanal$ = anal
   predigi$ = digi
DO
LINE (200, 358)-(400, 394), , B
LINE (202, 360)-(398, 392), , BF
      LOCATE 24, 29
      PRINT "CAMBIAR: no    si  "
      LOCATE 29, 2
      PRINT "<FLECHAS=Seleccionar>";
      CIRCLE (320, 376), 6
      CIRCLE (368, 376), 6
      CIRCLE (320, 376), 3, 15
      PAINT (320, 376), 15, 15
      salirsino% = 1
      DO
        DO
        v$ = INKEY$
        LOOP WHILE v$ = ""
        IF RIGHT$(v$, 1) = CHR$(77) AND (salirsino% = 1) THEN
           PAINT (320, 376), 0, 0
           CIRCLE (368, 376), 3, 15
           PAINT (368, 376), 15, 15
           salirsino% = 0
        ELSEIF RIGHT$(v$, 1) = CHR$(75) AND salirsino% = 0 THEN
           PAINT (368, 376), 0, 0
           CIRCLE (320, 376), 3, 15
           PAINT (320, 376), 15, 15
             salirsino% = 1
```

```
      END IF
    LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
LINE (200, 358)-(400, 394), 7, BF

 IF salirsino% <> 1 THEN
  LINE (7, 448)-(200, 464), 7, BF
  CALL inpu(49, numero, entr, 1, 8, 27)
     LOCATE 8, 27
     PRINT blanquear$
     LOCATE 8, 27
     PRINT entr
  CALL inpu(49, numero, sali, 1, 12, 27)
     LOCATE 12, 27
     PRINT blanquear$
     LOCATE 12, 27
     PRINT sali
  CALL inpu(49, numero, anal, 1, 16, 27)
     LOCATE 16, 27
     PRINT blanquear$
     LOCATE 16, 27
     PRINT anal
  CALL inpu(49, numero, digi, 1, 20, 27)
     LOCATE 20, 27
     PRINT blanquear$
     LOCATE 20, 27
     PRINT digi
 END IF
LOOP WHILE salirsino% = 0

     LINE (7, 448)-(359, 464), 7, BF
     LINE (551, 448)-(639, 464), 7, BF
REDIM borrarporcentaje(8000)
    GET (310, 160)-(550, 320), borrarporcentaje

 IF (preentr$ <> entr  OR men = 5 OR be% = 1) AND entr <> "" AND
 sali <> "" THEN
     OPEN entr FOR RANDOM AS #2 LEN = 2
     maximo = LOF(2)
     CLOSE (2)
  IF  MID$(entr, 2,  1) <> ":"  OR LEN(entr)  = 1 OR  maximo = 0
 THEN
     be% = 1
     ERROR 63
  ELSE
     be% = 0
   nn = 0
   IF RIGHT$(entr, 3) = "wav" THEN
      CALL porcentaje
      LOCATE 14, 43
      PRINT "INTERPRETANDO"
      LOCATE 15, 43
      PRINT "EL FORMATO *.WAV..."
      LINE (360, 255)-(510, 275), 0, BF
      archivowav$ = entr
      final% = INSTR(entr, ".wav")
      MID$(entr, final% + 1, 3) = "chs"
      OPEN entr FOR RANDOM AS #2 LEN = 2
```

```
    OPEN archivowav$ FOR RANDOM AS #1 LEN = 1
    nreg = 43
    DO
        nreg = nreg + 1
        GET #1, nreg, prewa
        xx = ASC(prewa) / 128 - 1
        IF xx = -1 THEN xx = 0
        yy = FIX(xx * 2048)
        IF xx > 7 THEN
          xx = 7.5
        END IF
        wa = MKI$(yy)
        PUT #2, nreg - 43, wa
    LOOP WHILE NOT EOF(1)
    CLOSE #1
    CLOSE #2

    CALL inspeccion(max, maximo, entr)
    CALL salidatotal

    saliwav$ = sali
    final% = INSTR(presali$, ".")
    MID$(saliwav$, final% + 1, 3) = "wav"
    OPEN sali FOR RANDOM AS #2 LEN = 2
    OPEN saliwav$ FOR RANDOM AS #1 LEN = 1
    OPEN archivowav$ FOR RANDOM AS #3 LEN = 1
    nreg = 0
    DO
      nreg = nreg + 1
      GET #3, nreg, prewa
      PUT #1, nreg, prewa
      LOOP WHILE nreg <= 43
    CLOSE #3
    DO
      nreg = nreg + 1
      GET #2, nreg, wa
      yy = CVI(wa)
      xx = yy / 2048
      IF xx = 0 THEN xx = -1
         prewa = MKI$((xx + 1) * 128)
         PUT #1, nreg, prewa
      LOOP WHILE NOT EOF(2)
    CLOSE #1
    CLOSE #2

  ELSE  ´Cuando no sean archivos *.wav
     CALL inspeccion(max, maximo, entr)
     CALL salidatotal
  END IF
     presali$ = sali
 END IF
END IF

 ´Si cambio solo el archivo de salida
 IF ((presali$ <> sali AND preentr$ = entr) OR bs% = 1) AND
                                  sali <> "" AND entr <> "" THEN
     bs% = 1
```

```
  IF MID$(sali, 2, 1) <> ":" OR LEN(sali) = 1 THEN
     ERROR 63
  ELSE
     CALL salidatotal
     bs% = 0
  END IF
END IF

'Si cambio el archivo de la resp. analógica
IF (preanal$ <> anal OR men = 5 OR ba% = 1) AND
                                                    anal <> "" THEN
      ba% = 1
   IF MID$(anal, 2, 1) <> ":" OR LEN(anal) = 1 THEN
      ERROR 63
   ELSE
      OPEN anal FOR RANDOM AS #2 LEN = 2
         CALL porcentaje
         LOCATE 14, 43
         PRINT "ARCHIVANDO...."
         LOCATE 15, 43
         PRINT "RESP.FREC.ANALOGICA"
         LINE (360, 255)-(510, 275), 0, BF
         nreg = 0
         IF trans = 1 THEN
         FOR i = 1 TO 304
              nreg = nreg + 1
              yy = FIX(hat(i) * 2048)
              ra = MKI$(yy)
              PUT #2, nreg, ra
         NEXT i
         ELSE
         FOR i = 1 TO 304
              nreg = nreg + 1
              yy = FIX(ha(i) * 2048)
              ra = MKI$(yy)
              PUT #2, nreg, ra
         NEXT i
         END IF
         CLOSE #2
      ba% = 0
   END IF
END IF

'Si cambio el archivo de resp.digital
IF (predigi$ <> digi OR men = 5 OR bd% = 1) AND
                                                    digi <> "" THEN
         bd% = 1
      IF MID$(digi, 2, 1) <> ":" OR LEN(digi) = 1 THEN
         ERROR 63
      ELSE
         OPEN digi FOR RANDOM AS #2 LEN = 2
         CALL porcentaje
         LOCATE 14, 43
         PRINT "ARCHIVANDO..."
         LOCATE 15, 43
         PRINT "RESP.FREC.DIGITAL"
         LINE (360, 255)-(510, 275), 0, BF
```

```
            nreg = 0
            SELECT CASE tipconv
            CASE 1:
              FOR i = 1 TO 304
                 nreg = nreg + 1
                 yy = FIX(hi(i) * 2048)
                 rd = MKI$(yy)
                 PUT #2, nreg, rd
              NEXT i
            CASE 2:
              FOR i = 1 TO 304
                 nreg = nreg + 1
                 yy = FIX(hb(i) * 2048)
                 rd = MKI$(yy)
                 PUT #2, nreg, rd
              NEXT i
            CASE 3:
              FOR i = 1 TO 304
                 nreg = nreg + 1
                 yy = FIX(hp(i) * 2048)
                 rd = MKI$(yy)
                 PUT #2, nreg, rd
              NEXT i
            END SELECT
            CLOSE #2
            bd% = 0
         END IF
  END IF
END IF
PUT (310, 160), borrarporcentaje, PSET
LOCATE 29, 27
PRINT "<ENTER=Aceptar>";
LOCATE 29, 2
PRINT "<FLECHAS=Seleccionar>";
LOCATE 29, 70
PRINT "<F10=Menú>";
ERASE borrarporcentaje
END SUB



´Subrutina "inspeccion"



SUB inspeccion (max, maximo, entr AS STRING)
     ´Calcula el maximo y el max
          OPEN entr FOR RANDOM AS #2 LEN = 2
          CALL porcentaje
          LOCATE 14, 43
          PRINT "INSPECCIONANDO..."
          LOCATE 15, 43
          PRINT "     ..........ARCHIVOS"
          LINE (360, 255)-(510, 275), 0, BF
          maximo = LOF(2)
          FIELD 2, 2 AS r$
```

```
          max = 0
          nreg = 0
          DO
            nreg = nreg + 1
            GET #2, nreg
            IF ABS(CVI(r$) / 2048) > max THEN
               max = ABS(CVI(r$) / 2048)
            END IF
          LOOP WHILE NOT EOF(2)
          CLOSE #2
END SUB
```

```
'Subrutina "valoresiniciales"
```

```
SUB valoresiniciales
     men = 1
     empezar% = 0
     entr = "b:\vocalih.013"
     sali = "b:\filtrada.dat"
     digi = "b:\d.dig"
     anal = "b:\a.anl"
     comp = 5
     tip = 1
     tipana = 1
     tipconv = 1
     attp = 1
     atts = 20
     fl = 1200
     fu = 1800
     fup = 4800
     fm = 12000
     trans = 2
     max = .25292968#
     maximo = 1280
 END SUB
```

```
 'Subrutina "graficos"
```

```
 SUB graficos
  REDIM va(601) AS SINGLE
  REDIM vb(601) AS SINGLE
  DIM arch1 AS STRING
  DIM arch2 AS STRING
  DO
   CALL caratula(tip, tipana, tipconv, trans)
   CALL graficosmenu(comp)
   SELECT CASE comp
   CASE 1:
```

```
    DO
      errorabrir% = 0
      LOCATE 22, 27
      PRINT " NOMBRE: "; arch1
      CALL inpu(19, numero, arch1, 1, 22, 36)
      LOCATE 22, 36
      PRINT ; arch1
      IF LEN(arch1) = 1 OR MID$(arch1, 2, 1) <> ":" THEN
        ERROR 63
      ELSE
        OPEN arch1 FOR RANDOM AS #2 LEN = 2
        IF errorabrir% = 0 THEN
          IF (LOF(2) = 0) THEN ERROR 63
        END IF
      END IF
    LOOP UNTIL NOT errorabrir%
      IF arch1 <> "" THEN
      FIELD 2, 2 AS r$
      nreg = 0
      FOR i = 1 TO 304
           nreg = nreg + 1
           GET #2, nreg
           va(i) = CVI(r$) / 2048
           IF ABS(va(i)) > maxi THEN
                maxi = ABS(va(i))
           END IF
      NEXT i
      CLOSE #2
      FOR i = 1 TO 304
        va(i) = va(i) / maxi
      NEXT
      bandera% = 1
      END IF
CASE 2:
    DO
      errorabrir% = 0
      LOCATE 22, 27
      PRINT " NOMBRE: "; arch2
      CALL inpu(19, numero, arch2, 1, 22, 36)
      LOCATE 22, 36
      PRINT ; arch2
      IF LEN(arch2) = 1 OR MID$(arch2, 2, 1) <> ":" THEN
        ERROR 63
      ELSE
        OPEN arch2 FOR RANDOM AS #2 LEN = 2
        IF errorabrir% = 0 THEN
          IF (LOF(2) = 0) THEN ERROR 63
        END IF
      END IF
    LOOP UNTIL NOT errorabrir%
      IF arch2 <> "" THEN
      FIELD 2, 2 AS r$
      nreg = 0
      FOR i = 1 TO 304
           nreg = nreg + 1
           GET #2, nreg
           vb(i) = CVI(r$) / 2048
```

```
                IF ABS(vb(i)) > maxi THEN
                    maxi = ABS(vb(i))
                END IF
            NEXT i
            CLOSE #2
            FOR i = 1 TO 304
               vb(i) = vb(i) / maxi
            NEXT
            bandera% = 1
            END IF
    CASE 3:
       CALL caratula(tip, tipana, tipconv, trans)
            'Dibujo de líneas punteadas para comparación
            LINE (8, 448)-(192, 464), 0, BF
            LINE (20, 66)-(620, 426), , BF
            LOCATE 4, 18
            PRINT "COMPARACION DE DOS RESPUESTAS DE FRECUENCIA"
            LOCATE 5, 6
            PRINT "|H|"
            LINE (50, 96)-(50, 396), 0
            LINE -(590, 396), 0
            FOR k = 1 TO 10
                 FOR i = 1 TO 100
                      PSET (k * 54 + 50, i * 3 + 96), 0
                 NEXT i
            NEXT k
            FOR k = 1 TO 10
                 FOR i = 1 TO 100
                      PSET (i * 5.4 + 50, k * 30 + 66), 0
                 NEXT i
            NEXT k
            FOR k = 1 TO 4
                 LOCATE 26, 4 + 14 * k
                 PRINT .2 * k; "π"
            NEXT k
            LOCATE 26, 73
            PRINT " π "
            LOCATE 7, 6
            PRINT "1"
            LOCATE 16, 5
            PRINT ".5"
            LOCATE 25, 6
            PRINT "0"
          KEY(10) OFF
          WINDOW (0, 0)-(152, 1)
          VIEW (50, 96)-(590, 396)
          IF bandera% = 1 THEN
           FOR i = 1 TO 151
              PSET (i, va(i)), 3
              LINE -(i + 1, va(i + 1)), 3
              PSET (i, vb(i)), 0
              LINE -(i + 1, vb(i + 1)), 0
           NEXT
          ELSE
             IF trans = 2 OR (tip = 1 AND tipana <> 4) THEN
                  FOR i = 1 TO 151
                       PSET (i, ha(i)), 3
```

```
                    LINE -(i + 1, ha(i + 1)), 3
                NEXT
            ELSEIF trans = 1 THEN
                FOR i = 1 TO 151
                    PSET (i, hat(i)), 3
                    LINE -(i + 1, hat(i + 1)), 3
                NEXT
            END IF
            IF tipconv = 1 THEN
                FOR i = 1 TO 151
                    PSET (i, hi(i)), 0
                    LINE -(i + 1, hi(i + 1)), 0
                NEXT
            ELSEIF tipconv = 2 THEN
                FOR i = 1 TO 151
                    PSET (i, hb(i)), 0
                    LINE -(i + 1, hb(i + 1)), 0
                NEXT
            ELSEIF tipconv = 3 THEN
                FOR i = 1 TO 151
                    PSET (i, hp(i)), 0
                    LINE -(i + 1, hp(i + 1)), 0
                NEXT
            END IF
        END IF
        VIEW
        WINDOW
        KEY(10) ON
    DO
     DO
     v$ = INKEY$
     LOOP WHILE v$ = ""
    LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
  CASE 5:
    CALL dibujo
 END SELECT
 LOOP
 ERASE va
 ERASE vb
END SUB



'Subrutina "menu"


SUB menu (men AS INTEGER)
     LINE (7, 60)-(230, 396), , BF
     LINE (8, 61)-(229, 395), 0, B
     LINE (181, 80)-(200, 376), 0, BF
     LINE (40, 357)-(200, 376), 0, BF
     LINE (19, 70)-(181, 357), 0, BF
     LINE (20, 71)-(180, 356), , B
     LINE (20, 102)-(180, 102)
     LINE (20, 133)-(180, 133)
     LINE (20, 165)-(180, 165)
```

```
    LINE (20, 196)-(180, 196)
    LINE (20, 228)-(180, 228)
    LINE (20, 260)-(180, 260)
    LINE (20, 292)-(180, 292)
    LINE (20, 324)-(180, 324)
REDIM priletra(200)
    LOCATE 6, 5
    PRINT "TIPO"
    GET (31, 81)-(40, 94), priletra
    PUT (31, 81), priletra, PRESET
    LOCATE 8, 5
    PRINT "ANALOGICO"
    GET (31, 113)-(38, 126), priletra
    PUT (31, 113), priletra, PRESET
    LOCATE 10, 5
    PRINT "CONVERSION"
    GET (31, 145)-(39, 158), priletra
    PUT (31, 145), priletra, PRESET
    LOCATE 12, 5
    PRINT "TRANSFORMACION"
    GET (40, 177)-(47, 190), priletra
    PUT (40, 177), priletra, PRESET
    LOCATE 14, 5
    PRINT "SIMULACION"
    GET (31, 209)-(40, 222), priletra
    PUT (31, 209), priletra, PRESET
    LOCATE 16, 5
    PRINT "ARCHIVAR"
    GET (65, 241)-(70, 254), priletra
    PUT (65, 241), priletra, PRESET
    LOCATE 18, 5
    PRINT "GRAFICOS"
    GET (31, 273)-(39, 286), priletra
    PUT (31, 273), priletra, PRESET
    LOCATE 20, 5
    PRINT "AYUDA"
    GET (40, 305)-(47, 318), priletra
    PUT (40, 305), priletra, PRESET
    LOCATE 22, 5
    PRINT "SALIR"
    GET (48, 337)-(55, 350), priletra
    PUT (48, 337), priletra, PRESET

DO
SELECT CASE men
CASE 1:
    IF anteriormen% = 9 THEN
        LINE (21, 325)-(179, 355), 0, BF
        LOCATE 22, 5
        PRINT "SALIR"
        GET (48, 337)-(55, 350), priletra
        PUT (48, 337), priletra, PRESET
    ELSEIF anteriormen% = 2 THEN
        LINE (21, 103)-(179, 132), 0, BF
        LOCATE 8, 5
        PRINT "ANALOGICO"
        GET (31, 113)-(38, 126), priletra
```

```
            PUT (31, 113), priletra, PRESET
         END IF
       anteriormen% = 1
            LINE (20, 71)-(180, 102), , BF
            LOCATE 6, 5
            PRINT "TIPO"
            GET (31, 81)-(40, 94), priletra
            PUT (31, 81), priletra, PRESET
CASE 2:
       IF anteriormen% = 1 THEN
            LINE (21, 72)-(179, 101), 0, BF
            LOCATE 6, 5
            PRINT "TIPO"
            GET (31, 81)-(40, 94), priletra
            PUT (31, 81), priletra, PRESET
       ELSEIF anteriormen% = 3 THEN
            LINE (21, 134)-(179, 164), 0, BF
            LOCATE 10, 5
            PRINT "CONVERSION"
            GET (31, 145)-(39, 158), priletra
            PUT (31, 145), priletra, PRESET
       END IF
       anteriormen% = 2
            LINE (21, 102)-(179, 133), , BF
            LOCATE 8, 5
            PRINT "ANALOGICO"
            GET (31, 113)-(38, 126), priletra
            PUT (31, 113), priletra, PRESET
CASE 3:
       IF anteriormen% = 2 THEN
            LINE (21, 103)-(179, 132), 0, BF
            LOCATE 8, 5
            PRINT "ANALOGICO"
            GET (31, 113)-(38, 126), priletra
            PUT (31, 113), priletra, PRESET
       ELSEIF anteriormen% = 4 THEN
            LINE (21, 166)-(179, 195), 0, BF
            LOCATE 12, 5
            PRINT "TRANSFORMACION"
            GET (40, 177)-(47, 190), priletra
            PUT (40, 177), priletra, PRESET
       END IF
       anteriormen% = 3
            LINE (21, 133)-(179, 165), , BF
            LOCATE 10, 5
            PRINT "CONVERSION"
            GET (31, 145)-(39, 158), priletra
            PUT (31, 145), priletra, PRESET
CASE 4:
       IF anteriormen% = 5 THEN
            LINE (21, 197)-(179, 227), 0, BF
            LOCATE 14, 5
            PRINT "SIMULACION"
            GET (31, 209)-(40, 222), priletra
            PUT (31, 209), priletra, PRESET
       ELSEIF anteriormen% = 3 THEN
            LINE (21, 134)-(179, 164), 0, BF
```

```
            LOCATE 10, 5
            PRINT "CONVERSION"
            GET (31, 145)-(39, 158), priletra
            PUT (31, 145), priletra, PRESET
        END IF
        anteriormen% = 4
            LINE (21, 165)-(179, 196), , BF
            LOCATE 12, 5
            PRINT "TRANSFORMACION"
            GET (40, 177)-(47, 190), priletra
            PUT (40, 177), priletra, PRESET
    CASE 5:
        IF anteriormen% = 6 THEN
            LINE (21, 229)-(179, 259), 0, BF
            LOCATE 16, 5
            PRINT "ARCHIVAR"
            GET (65, 241)-(70, 254), priletra
            PUT (65, 241), priletra, PRESET
        ELSEIF anteriormen% = 4 THEN
            LINE (21, 166)-(179, 195), 0, BF
            LOCATE 12, 5
            PRINT "TRANSFORMACION"
            GET (40, 177)-(47, 190), priletra
            PUT (40, 177), priletra, PRESET
        END IF
        anteriormen% = 5
            LINE (21, 196)-(179, 228), , BF
            LOCATE 14, 5
            PRINT "SIMULACION"
            GET (31, 209)-(40, 222), priletra
            PUT (31, 209), priletra, PRESET
    CASE 6:
        IF anteriormen% = 5 THEN
            LINE (21, 197)-(179, 227), 0, BF
            LOCATE 14, 5
            PRINT "SIMULACION"
            GET (31, 209)-(40, 222), priletra
            PUT (31, 209), priletra, PRESET
        ELSEIF anteriormen% = 7 THEN
            LINE (21, 261)-(179, 291), 0, BF
            LOCATE 18, 5
            PRINT "GRAFICOS"
            GET (31, 273)-(39, 286), priletra
            PUT (31, 273), priletra, PRESET
        END IF
        anteriormen% = 6
            LINE (21, 228)-(179, 260), , BF
            LOCATE 16, 5
            PRINT "ARCHIVAR"
            GET (65, 241)-(70, 254), priletra
            PUT (65, 241), priletra, PRESET
    CASE 7:
        IF anteriormen% = 6 THEN
            LINE (21, 229)-(179, 259), 0, BF
            LOCATE 16, 5
            PRINT "ARCHIVAR"
            GET (65, 241)-(70, 254), priletra
```

```
        PUT (65, 241), priletra, PRESET
    ELSEIF anteriormen% = 8 THEN
        LINE (21, 293)-(179, 323), 0, BF
        LOCATE 20, 5
        PRINT "AYUDA"
        GET (40, 305)-(47, 318), priletra
        PUT (40, 305), priletra, PRESET
    END IF
    anteriormen% = 7
        LINE (21, 260)-(179, 292), , BF
        LOCATE 18, 5
        PRINT "GRAFICOS"
        GET (31, 273)-(39, 286), priletra
        PUT (31, 273), priletra, PRESET
CASE 8:
    IF anteriormen% = 9 THEN
        LINE (21, 325)-(179, 355), 0, BF
        LOCATE 22, 5
        PRINT "SALIR"
        GET (48, 337)-(55, 350), priletra
        PUT (48, 337), priletra, PRESET
    ELSEIF anteriormen% = 7 THEN
        LINE (21, 261)-(179, 291), 0, BF
        LOCATE 18, 5
        PRINT "GRAFICOS"
        GET (31, 273)-(39, 286), priletra
        PUT (31, 273), priletra, PRESET
    END IF
    anteriormen% = 8
        LINE (21, 292)-(179, 324), , BF
        LOCATE 20, 5
        PRINT "AYUDA"
        GET (40, 305)-(47, 318), priletra
        PUT (40, 305), priletra, PRESET
CASE 9:
    IF anteriormen% = 8 THEN
        LINE (21, 293)-(179, 323), 0, BF
        LOCATE 20, 5
        PRINT "AYUDA"
        GET (40, 305)-(47, 318), priletra
        PUT (40, 305), priletra, PRESET
    ELSEIF anteriormen% = 1 THEN
        LINE (21, 72)-(179, 101), 0, BF
        LOCATE 6, 5
        PRINT "TIPO"
        GET (31, 81)-(40, 94), priletra
        PUT (31, 81), priletra, PRESET
    END IF
    anteriormen% = 9
        LINE (21, 324)-(179, 356), , BF
        LOCATE 22, 5
        PRINT "SALIR"
        GET (48, 337)-(55, 350), priletra
        PUT (48, 337), priletra, PRESET
END SELECT
    DO
    v$ = INKEY$
```

```
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        men = men - 1
     IF men = 0 THEN men = 9
    ELSEIF RIGHT$(v$, 1) = CHR$(80) THEN
        men = men + 1
     IF men = 10 THEN men = 1
    ELSEIF RIGHT$(v$, 1) = CHR$(84) OR
                                   RIGHT$(v$, 1) = CHR$(116) THEN
        men = 1
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(65) OR
                                   RIGHT$(v$, 1) = CHR$(97) THEN
        men = 2
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(67) OR
                                   RIGHT$(v$, 1) = CHR$(99) THEN
        men = 3
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(82) OR
                                   RIGHT$(v$, 1) = CHR$(114) THEN
        men = 4
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(83) OR
                                   RIGHT$(v$, 1) = CHR$(115) THEN
        men = 5
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(73) OR
                                   RIGHT$(v$, 1) = CHR$(105) THEN
        men = 6
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(71) OR
                                   RIGHT$(v$, 1) = CHR$(103) THEN
        men = 7
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(89) OR
                                   RIGHT$(v$, 1) = CHR$(121) THEN
        men = 8
        v$ = CHR$(13)
    ELSEIF RIGHT$(v$, 1) = CHR$(76) OR
                                   RIGHT$(v$, 1) = CHR$(108) THEN
        men = 9
        v$ = CHR$(13)
     END IF
LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
END SUB
```

b) SUBRUTINAS QUE REALIZAN CALCULOS

```
´Subrutina "simulacion"


SUB simulacion
 CALL caratula(tip, tipana, tipconv, trans)
        LINE (7, 448)-(356, 464), 0, BF
 CALL porcentaje
 CALL waps(alfaso, alfasola, Bo, fl, fm, fu, fup, kso, ksola,
                          tip, tipana, trans, u, wap, was)
 CALL orden
 CALL polos                   ´Obtención de polos
 CALL frecuenciana(c(), gama(), ha(), kanaloga,  p, sec,
                                                 tipana)
SELECT CASE tipconv
 CASE 1, 3          ´Selección del método invarianza de impulso
     CALL fparciales(aa(), c(), kanaloga, n, s(), sec, tipana)
     IF tipconv = 1 THEN
      CALL transformacionimpulso(aa(), alfa(), alfaso,
                                alfasola, beta(), kpulso(),
                                     kso, p, s(), sec, tip)
     ELSEIF tipconv = 3 THEN
      CALL transformacionpulso(aa(), alfa(),  alfaso, alfasola,
                             beta(), kpulso(), kso,  p, s(),
                                                sec, tip)
     END IF
      CALL  transfrecuenciapulso(alfa(), beta(), hi(), hp(),
                             kanaloga, kpulso(), p, sec,
                                    tip, tipana, tipconv)
 CASE 2             ´Selección del método transformada bilineal
     IF trans = 1 THEN
      CALL frecuencianat
      IF tip = 1 THEN
       CALL bilineal
       CALL frecuenciabilineal
      ELSE
       CALL  transformacionbilineal(alfa(), alfaso, alfasola,
                                beta(), c(), kanaloga,
                                kbilineal,kso,l,n, p, s(),
                                sec, tip, tipana, trans, u)
       CALL transfrecuenciabilineal(alfa(), beta(), hb(),
                                kbilineal,p,sec,tip,tipana)
      END IF
     ELSE
      CALL bilineal
      CALL transformacionbilineal(alfa(), alfaso, alfasola,
                               beta(), c(), kanaloga,
                               kbilineal,kso, l, n,  p, s(),
                               sec, tip, tipana, trans, u)
      CALL transfrecuenciabilineal(alfa(), beta(), hb(),
                                kbilineal,p,sec,tip,tipana)
     END IF
```

```
END SELECT
     KEY(10) OFF
     CALL archivo(ha(), hat(), hb(), hi(), hp(),  nn, tipconv,
                  trans, anal, digi, entr, sali, men,
                  errorabrir%, max, maximo, be%,bs%, ba%, bd%)
     KEY(10) ON
     CALL caratula(tip, tipana, tipconv, trans)
END SUB



'Subrutina "waps"


SUB waps (alfaso, alfasola, Bo,  fl, fm, fu, fup, kso, ksola,
          tip AS INTEGER, tipana AS INTEGER,  trans AS INTEGER,
                                              u, wap, was)
  'Calculo de was y wap
 wap = 1
 SELECT CASE trans
 CASE 1:
  IF tipana <> 4 THEN
     SELECT CASE tip
     CASE 1:
         wap = 2 * fl * pi / fm
         was = 2 * fu * pi / fm
         u = 1
     CASE 2:
         u = 4 * TAN(wap / 2) * TAN(fu * pi / fm)
         was = 2 * ATN(u / TAN(fl * pi / fm) / 4)
     CASE 3:
         Bo = 4*(TAN(fup*pi/fm) - TAN(fl*pi/fm)) * TAN(wap/2)
         was = 2 * ATN(Bo / 4 /
                     (TAN((fup-fu+fl)*pi/fm)-TAN(fu*pi/fm)))
     CASE 4:
         Bo = (TAN(fup*pi/fm) - TAN(fu*pi/fm)) / TAN(wap / 2)
         was =  2 * ATN((TAN((fup + fu - fl) * pi / fm) -
                                   TAN(fl * pi / fm)) / Bo)
     END SELECT
   ELSE
     ksola = fl / fu
     SELECT CASE tip
     CASE 1:
        u = 2 * TAN(fl * pi / fm) / SQR(ksola)
     CASE 2:
        u = 2 * SQR(ksola) * TAN(fu * pi / fm)
     CASE 3:
        Bo = 2 * (TAN(fup * pi / fm) -  TAN(fl * pi / fm)) *
                                                 SQR(ksola)
     CASE 4:
        Bo = 2 * (TAN(fup * pi / fm) - TAN(fu * pi / fm)) /
                                                 SQR(ksola)
     END SELECT
   END IF

 CASE 2:
```

```
IF tipana <> 4 THEN
   SELECT CASE tip
   CASE 1:
    wap = 2 * fl * pi / fm
    was = 2 * fu * pi / fm
    alfasola = 0
   CASE 2:
      LOCATE 14, 46
      PRINT "ESPERE ...."
    alfasola = COS(wap/2 + fu * pi / fm) /
                                  COS(wap/2 - fu * pi / fm)
       was = 0
       DO
       was = was + 1 / 1000
       prealfasola = alfasola-COS(was/2 + fl*pi/fm) /
                                     COS(was/2 - fl*pi/fm)
       LOOP WHILE ABS(prealfasola) > .01
   CASE 3:
    alfaso = COS(fl * pi / fm + fup * pi / fm) /
                                  COS(fup*pi/fm - fl*pi/fm)
    kso = TAN(fup * pi / fm - fl * pi / fm) * TAN(wap / 2)
    was = 2 * ATN(kso / TAN((fup + fl - 2 * fu) * pi / fm))
   CASE 4:
    alfaso = COS(fu * pi / fm + fup * pi / fm) /
                                  COS(fup*pi/fm - fu*pi/fm)
    kso = (TAN(fup * pi / fm -  fu * pi / fm)) ^ (-1) *
                                            TAN(wap / 2)
    was = 2 * ATN(kso * TAN((fup + fu - 2 * fl) * pi / fm))
   END SELECT
 ELSE
    ksola = fl / fu
    SELECT CASE tip
    CASE 1:
     alfasola = SIN(SQR(ksola) / 2 - fl  * pi / fm) /
                            SIN(SQR(ksola) / 2 + fl * pi / fm)
    CASE 2:
     alfasola = COS(SQR(ksola) / 2 + fu  * pi / fm) /
                            COS(SQR(ksola) / 2 - fu * pi / fm)
    CASE 3:
     alfaso = COS(fl * pi / fm + fup * pi / fm) /
                             COS(fup * pi / fm - fl * pi / fm)
     kso = TAN(fup * pi / fm - fl * pi / fm) *
                                        TAN(SQR(ksola) / 2)
    CASE 4:
     alfaso = COS(fu * pi / fm + fup * pi / fm) /
                             COS(fup * pi / fm - fu * pi / fm)
     kso = (TAN(fup * pi / fm - fu * pi / fm)) ^ (-1) *
                                        TAN(SQR(ksola) / 2)
    END SELECT
  END IF
END SELECT
END SUB
```

```
´Subrutina "orden"


SUB orden
     ´Calculo de ∈
     ep1 = SQR(10 ^ (attp / 10) - 1)
     ´Calculo de lamda
     lam = SQR(10 ^ (atts / 10) - 1)
  SELECT CASE tipana
  CASE 1
     SELECT CASE tipconv
     CASE 1, 3
     ´Orden del filtro
       nt = LOG(lam / ep1) / (LOG(was / wap))
       n = INT(nt) + 1
     ´Cálculo de Wac frecuencia análoga para tener un módulo  a
                                                            -3dB
       wac = wap / ep1 ^ (1 / n)
     CASE 2
     ´Orden del filtro
       wap = 2 * TAN(wap / 2)
       was = 2 * TAN(was / 2)
       nt = LOG(lam / ep1) / LOG(was / wap)
       n = INT(nt) + 1
     ´Cálculo de Wac frecuencia análoga para tener un módulo  a
                                                            -3dB
       wac = was / lam ^ (1 / n)
     END SELECT
  CASE 2, 3
     IF tipconv = 2 THEN
     wap = 2 * TAN(wap / 2)
     was = 2 * TAN(was / 2)
     END IF
     nt = LOG(lam / ep1 + SQR((lam / ep1) ^ 2 - 1)) /
                    LOG(was / wap + SQR((was / wap) ^ 2 - 1))
     n = INT(nt) + 1
     ´Calculo del alfa para chevishev
     IF tipana = 2 THEN
          alf = LOG(1 / ep1 + SQR((1 / ep1) ^ 2 + 1)) / n
     ELSE
          alf = LOG(lam + SQR(lam ^ 2 + 1)) / n
     END IF
  CASE 4
     d = lam ^ 2 / ep1 ^ 2
     kprima = SQR(1 - ksola ^ 2)
     qo = (1 - SQR(kprima)) / (2 * (1 + SQR(kprima)))
     q = qo + 2 * qo ^ 5 + 15 * qo ^ 9 + 150 * qo ^ 13
     nt = LOG(16 * d) / LOG(1 / q)
     n = INT(nt) + 1
  END SELECT
   ´Analisis de la paridad
     n1 = n / 2
     n2 = n1 - FIX(n1)
   IF n2 = 0 THEN
     p = 1
     sec = n / 2
   ELSE
```

```
        p = 2
        sec = (n - 1) / 2
     END IF
  END SUB



  'Subrutina "polos"


  SUB polos
    SELECT CASE tipana
    CASE 1
       FOR i = 1 TO sec
            s(i).real = wac * COS(pi* (2 * i + n - 1) / (2 * n))
            s(n - i + 1).real = s(i).real
            s(i).imag = wac * SIN(pi* (2 * i + n - 1) / (2 * n))
            s(n - i + 1).imag = -s(i).imag
            gama(1, i) = -2 * s(i).real
            gama(2, i) = (s(i).real) ^ 2 + (s(i).imag) ^ 2
       NEXT
        IF p = 2 THEN
         s(sec + 1).real = wac*COS(pi*(2*(sec+1)+n-1) / (2 * n))
         s(sec + 1).imag = 0
         gama(2, sec + 1) = -s(sec + 1).real
        END IF
       kanaloga = wac ^ n
    CASE 2, 3   'Polos para el filtro de Chevishev
       kanaloga = 1
       FOR i = 1 TO sec
            s(i).real = wap  * (EXP(alf) - EXP(-alf)) / 2 *
                               COS(pi * (2 * i + n - 1) / (2 * n))
            s(n - i + 1).real = s(i).real
            s(i).imag = wap  * (EXP(alf) + EXP(-alf)) / 2 *
                               SIN(pi * (2 * i + n - 1) / (2 * n))
            s(n - i + 1).imag = -s(i).imag
            gama(1, i) = -2 * s(i).real
            gama(2, i) = (s(i).real) ^ 2 + (s(i).imag) ^ 2
            kanaloga = kanaloga * gama(2, i)
       NEXT
       IF p = 2 THEN
        s(sec + 1).real = wap * (EXP(alf) - EXP(-alf)) / 2 *
                    COS(pi * (2 * (sec + 1) + n - 1) / (2 * n))
        s(sec + 1).imag = 0
        gama(2, sec + 1) = -s(sec + 1).real
        kanaloga = kanaloga * gama(2, sec + 1)
       ELSE
        kanaloga = kanaloga * 10 ^ (-attp / 20)
       END IF
       IF tipana = 3 THEN
        kanaloga = 1
       FOR i = 1 TO sec
            B = was * wap / (s(i).real ^ 2 + s(i).imag ^ 2)
            s(i).real = s(i).real * B
            s(n + 1 - i).real = s(i).real
            s(i).imag = s(i).imag * B
```

```
            s(n + 1 - i).imag = -s(i).imag
            c(i) = (was / SIN((2 * i + n - 1)*pi / (2 * n))) ^ 2
            gama(1, i) = -2 * s(i).real
            gama(2, i) = (s(i).real) ^ 2 + (s(i).imag) ^ 2
            kanaloga = kanaloga * gama(2, i) / c(i)
       NEXT
            IF p = 2 THEN
                 s(sec + 1).real = was * wap / s(sec + 1).real
                 s(sec + 1).imag = 0
                 gama(2, sec + 1) = -s(sec + 1).real
                 kanaloga = kanaloga * gama(2, sec + 1)
            END IF
         END IF
CASE 4:
   LAMBDA = LOG((10^(attp/20) + 1) / (10^(attp/20) - 1))/(2*n)
     bb = 0
     m% = 0
    DO
     B = (-1) ^ m% * q ^ (m% * (m% + 1)) *
        (EXP((2*m% + 1)*LAMBDA) - EXP(-(2*m% + 1)*LAMBDA)) / 2
     bb = bb + B
     m% = m% + 1
    LOOP WHILE ABS(B) > .000001
     cc = 0
     m% = 1
    DO
     c = (-1) ^ m% * q ^ (m% ^ 2) *
        (EXP((2*m% + 1)*LAMBDA) + EXP(-(2*m% + 1)*LAMBDA)) / 2
     cc = cc + c
     m% = m% + 1
    LOOP WHILE ABS(B) > .000001
    sigmacero = 2 * q ^ (1 / 4) * bb / (1 + 2 * cc)
    doblev = SQR((1 + ksola * sigmacero ^ 2) *
                                  (1 + sigmacero ^ 2 / ksola))
     kanaloga = 1
    FOR i = 1 TO sec
          IF p = 2 THEN
               bb = 0
               m% = 0
              DO
               B = (-1) ^ m% * q ^ (m% * (m% + 1)) *
                                 SIN((2 * m% + 1) * pi * i / n)
               bb = bb + B
               m% = m% + 1
              LOOP WHILE ABS(B) > .000001
               cc = 0
               m% = 1
              DO
               c = (-1) ^ m% * q ^ (m% ^ 2) *
                                       COS(2 * m% * pi * i / n)
               cc = cc + c
               m% = m% + 1
              LOOP WHILE ABS(B) > .000001
           ELSE
               bb = 0
               m% = 0
              DO
```

```
          B = (-1) ^ m% * q ^ (m% * (m% + 1)) *
                         SIN((2 * m% + 1) * pi * (i - .5) / n)
          bb = bb + B
          m% = m% + 1
         LOOP WHILE ABS(B) > .000001
          cc = 0
          m% = 1
         DO
          c = (-1) ^ m% * q ^ (m% ^ 2) *
                              COS(2 * m% * pi * (i - .5) / n)
          cc = cc + c
          m% = m% + 1
          LOOP WHILE ABS(B) > .000001
       END IF
      omega = 2 * q ^ (1 / 4) * bb / (1 + 2 * cc)
      v = SQR((1 - ksola * omega^2) *(1 - omega^2 / ksola))
      c(i) = 1 / omega ^ 2
      de = 1 + (sigmacero * omega) ^ 2
    gama(2, i) = ((sigmacero*v)^2 + (omega*doblev)^2) / de^2
    gama(1, i) = (2 * sigmacero * v) / de
    s(i).real = gama(1, i) / -2
    s(n + 1 - i).real = s(i).real
    s(i).imag = SQR(gama(2, i) - s(i).real ^ 2)
    s(n + 1 - i).imag = -s(i).imag
    kanaloga = kanaloga * gama(2, i) / c(i)
   NEXT
      IF p = 2 THEN
        gama(2, sec + 1) = sigmacero
        s(sec + 1).real = -sigmacero
        s(sec + 1).imag = 0
        kanaloga = kanaloga * sigmacero
      ELSE
        kanaloga = kanaloga * 10 ^ (-attp * .05)
      END IF
END SELECT
END SUB


'Subrutina "frecuenciana"


SUB frecuenciana (c(), gama(), ha(), kanaloga, p AS INTEGER,
                              sec AS INTEGER, tipana AS INTEGER)
 LOCATE 14, 46
  PRINT "CALCULANDO....H(s)"
 DIM a AS DOUBLE
 FOR i = 1 TO 304
     w = i * 2 * pi / 304
     a = 1
     FOR k = 1 TO sec
       a = a * SQR((gama(2, k) - w^2)^2 + (gama(1, k) * w)^2)
     NEXT
       IF p = 2 THEN
           a = a * SQR(gama(2, sec + 1) ^ 2 + w ^ 2)
       END IF
```

```
    ha(i) = kanaloga / a
    IF (tipana = 3 OR tipana = 4) THEN
         B = 1
         FOR k = 1 TO sec
         B = B * (c(k) - w ^ 2)
         NEXT
         ha(i) = ha(i) * ABS(B)
    END IF
 LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
 NEXT
    LINE (383, 256)-(461, 264), 0, BF
END SUB



´Subrutina "fparciales"


SUB fparciales (aa() AS complejo, c(), kanaloga, n  AS INTEGER,
           s() AS complejo, sec AS INTEGER, tipana AS INTEGER)
   REDIM den(1 TO 200) AS complejo
   REDIM in(0 TO 200) AS complejo
   REDIM ini(0 TO 200) AS complejo
   REDIM numini(0 TO 200) AS complejo
   REDIM numin(0 TO 200) AS complejo
   REDIM num(1 TO 200) AS complejo

  FOR j = 1 TO n
   m% = 0
   FOR i = 1 TO n
      IF j <> i THEN
        m% = m% + 1  ´Suma de complejos
        ini(m%).real = (s(j).real - s(i).real)
        ini(m%).imag = (s(j).imag - s(i).imag)
       END IF
   NEXT
      FOR ii = 2 TO (n - 1) ´Producto de complejos
        in(ii).real = ini(ii).real * ini(ii - 1).real -
                                ini(ii).imag * ini(ii - 1).imag
        ini(ii).imag = ini(ii).real * ini(ii - 1).imag  +
                                ini(ii).imag * ini(ii - 1).real
        ini(ii).real = in(ii).real
      NEXT
           den(j).real = ini(n - 1).real
           den(j).imag = ini(n - 1).imag
           deno = (den(j).real) ^ 2 + (den(j).imag) ^ 2
   IF (tipana = 1 OR tipana = 2) THEN
       aa(j).real = den(j).real * kanaloga / deno
       aa(j).imag = -den(j).imag * kanaloga / deno
   ELSE
      FOR i = 1 TO sec
           numini(i).real = s(j).real^ 2 - s(j).imag ^ 2 + c(i)
           numini(i).imag = 2 * s(j).real * s(j).imag
      NEXT
      FOR ii = 2 TO (sec)  ´Producto de complejos
        numin(ii).real = numini(ii).real*numini(ii - 1).real  -
```

```
                              numini(ii).imag*numini(ii - 1).imag
        numini(ii).imag  = numini(ii).real*numini(ii - 1).imag +
                              numini(ii).imag*numini(ii - 1).real
        numini(ii).real = numin(ii).real
      NEXT
      num(j).real = numini(sec).real
      num(j).imag = numini(sec).imag
      aa(j).real = (den(j).real*num(j).real +
                         den(j).imag*num(j).imag)*kanaloga / deno
      aa(j).imag = (-den(j).imag*num(j).real +
                         den(j).real*num(j).imag)*kanaloga / deno
   END IF
  NEXT
      ERASE den
      ERASE in
      ERASE ini
      ERASE numin
      ERASE numini
      ERASE num
 END SUB



 'Subrutina "transformacionimpulso"


 SUB transformacionimpulso (aa() AS complejo, alfa(),  alfaso,
                            alfasola, beta(), kpulso(), kso,
                            p AS INTEGER, s() AS complejo,
                               sec AS INTEGER, tip AS INTEGER)
  FOR i = 1 TO sec
   beta(0, i) = 2 * aa(i).real
   beta(1, i) = -2  * EXP(s(i).real) * (COS(s(i).imag) *
                     aa(i).real + SIN(s(i).imag) * aa(i).imag)
   alfa(1, i) = -2 * EXP(s(i).real) * COS(s(i).imag)
   alfa(2, i) = EXP(2 * s(i).real)
  NEXT
   IF p = 2 THEN
    beta(0, sec + 1) = aa(sec + 1).real
    alfa(1, sec + 1) = -EXP(s(sec + 1).real)
   END IF
 SELECT CASE tip
  CASE 1:
   FOR i = 1 TO sec
     pri = 1 - alfa(1, i) * alfasola + alfa(2, i) * alfasola^ 2
     segu = -2 * alfasola * (1 + alfa(2, i)) +
                                    alfa(1, i)*(1 + alfasola^2)
    alfa(2, i) = (alfasola ^ 2 - alfa(1, i) * alfasola +
                                               alfa(2, i)) / pri
    alfa(1, i) = segu / pri
    kpulso(i) = 1 / pri
     pri = beta(0, i) - beta(1, i) * alfasola
     segu = -2 * beta(0, i) * alfasola +
                                    beta(1, i)*(1 + alfasola^2)
    beta(1, i) = alfasola * (beta(0, i) * alfasola -
```

```
                                                   beta(1, i)) / pri
    beta(0, i) = segu / pri
    kpulso(i) = kpulso(i) * pri
   NEXT
    IF p = 2 THEN
      pri = beta(0, sec + 1)
     beta(0, sec + 1) = -beta(0, sec + 1) * alfasola / pri
      segu = 1 - alfa(1, sec + 1) * alfasola
     alfa(1, sec + 1) = (alfa(1, sec + 1) - alfasola) / segu
     kpulso(sec + 1) = pri / segu
    END IF
  CASE 2:
   FOR i = 1 TO sec
     pri = 1 + alfa(1, i) * alfasola + alfa(2, i) * alfasola^ 2
     segu = -2 * alfasola - alfa(1, i) * (1 + alfasola ^ 2) -
                                          2 * alfasola * alfa(2, i)
    alfa(2, i) = (alfasola ^ 2 + alfa(1, i) * alfasola +
                                                 alfa(2, i)) / pri
    alfa(1, i) = segu / pri
    kpulso(i) = 1 / pri
     pri = beta(0, i) + beta(1, i) * alfasola
     segu = -beta(0, i) * alfasola - beta(1, i) -
                alfasola * (beta(0, i) + beta(1, i) * alfasola)
    beta(1, i) = alfasola * (beta(0, i) * alfasola +
                                                 beta(1, i)) / pri
    beta(0, i) = segu / pri
    kpulso(i) = kpulso(i) * pri
   NEXT
    IF p = 2 THEN
      pri = beta(0, sec + 1)
     beta(0, sec + 1) = -beta(0, sec + 1) * alfasola / pri
      segu = 1 + alfa(1, sec + 1) * alfasola
     alfa(1, sec + 1) = (-alfa(1, sec + 1) - alfasola) / segu
     kpulso(sec + 1) = pri / segu
    END IF
  CASE 3:
    v1 = -2 * alfaso / (kso + 1)
    v2 = (1 - kso) / (kso + 1)
   FOR i = 1 TO sec
     pri = 1 + alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
     segu = v1 * (2 + alfa(1, i) * (1 + v2) + 2*v2* alfa(2, i))
     terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) +
                             alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
    alfa(3, i) = v1 * (2 * v2 + alfa(1, i) * (1 + v2) +
                                          2 * alfa(2, i)) / pri
    alfa(4, i) = (v2 ^ 2 + alfa(1, i) * v2 + alfa(2, i)) / pri
    alfa(1, i) = segu / pri
    alfa(2, i) = terc / pri
    kpulso(i) = 1 / pri
     pri = beta(0, i) + beta(1, i) * v2
     segu = v1 * (2 * beta(0, i) + beta(1, i) * (1 + v2))
     terc = (2 * v2 + v1) * beta(0, i) +
                             beta(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
    beta(2, i) = v1 * (v2 * (beta(0, i) + beta(1, i)) +
                             beta(0, i) * v2 + beta(1, i)) / pri
    beta(3, i) = v2 * (beta(0, i) * v2 + beta(1, i)) / pri
    beta(0, i) = segu / pri
```

```
   beta(1, i) = terc / pri
   kpulso(i) = kpulso(i) * pri
  NEXT
   IF p = 2 THEN
     pri = 1 + alfa(1, sec + 1) * v2
     segu = v1 * (1 + alfa(1, sec + 1))
    alfa(2, sec + 1) = (v2 + alfa(1, sec + 1)) / pri
    alfa(1, sec + 1) = segu / pri
     segu = beta(0, sec + 1)
     terc = beta(0, sec + 1) * v1
    beta(1, sec + 1) = v2 * beta(0, sec + 1) / segu
    beta(0, sec + 1) = terc / segu
    kpulso(sec + 1) = segu / pri
   END IF
CASE 4:
   v1 = -2 * alfaso * kso / (kso + 1)
   v2 = (kso - 1) / (kso + 1)
  FOR i = 1 TO sec
     pri = 1 - alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
     segu = v1*(2 - alfa(1, i) *(1 + v2) + 2 * v2 * alfa(2, i))
     terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) -
                                 alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
    alfa(3, i) = v1 * (2 * v2 - alfa(1, i) * (1 + v2) +
                                             2 * alfa(2, i)) / pri
    alfa(4, i) = (v2 ^ 2 - alfa(1, i) * v2 + alfa(2, i)) / pri
    alfa(1, i) = segu / pri
    alfa(2, i) = terc / pri
    kpulso(i) = 1 / pri
     pri = beta(0, i) - beta(1, i) * v2
     segu = v1 * (2 * beta(0, i) - beta(1, i) * (1 + v2))
     terc = (2 * v2 + v1 ^ 2) * beta(0, i) -
                                 beta(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
    beta(2, i) = v1 * (v2 * (beta(0, i)-beta(1, i)) +
                                beta(0, i) * v2 - beta(1, i)) / pri
    beta(3, i) = v2 * (beta(0, i) * v2 - beta(1, i)) / pri
    beta(0, i) = segu / pri
    beta(1, i) = terc / pri
    kpulso(i) = kpulso(i) * pri
   NEXT
    IF p = 2 THEN
      pri = 1 - alfa(1, sec + 1) * v2
      segu = v1 * (1 - alfa(1, sec + 1))
     alfa(2, sec + 1) = (v2 - alfa(1, sec + 1)) / pri
     alfa(1, sec + 1) = segu / pri
      segu = beta(0, sec + 1)
      terc = beta(0, sec + 1) * v1
     beta(1, sec + 1) = v2 * beta(0, sec + 1) / segu
     beta(0, sec + 1) = terc / segu
     kpulso(sec + 1) = segu / pri
    END IF
END SELECT
END SUB
```

´Subrutina "transformacionpulso"

```
SUB transformacionpulso (aa() AS complejo, alfa(), alfaso,
                          alfasola, beta(), kpulso(), kso,
                          p AS INTEGER, s() AS complejo,
                               sec AS INTEGER, tip AS INTEGER)
 FOR k = 1 TO sec
   conts = (s(k).real ^ 2 + s(k).imag ^ 2)
  beta(0, k) = -2  * ((aa(k).real * s(k).real + aa(k).imag *
               s(k).imag) * (1 - EXP(s(k).real) *
               COS(s(k).imag)) + (aa(k).imag * s(k).real -
               aa(k).real * s(k).imag)  * EXP(s(k).real) *
               SIN(s(k).imag)) / conts
  beta(1, k) = 2  * EXP(s(k).real) * ((COS(s(k).imag) -
               EXP(s(k).real)) * (aa(k).real * s(k).real +
               aa(k).imag * s(k).imag) + SIN(s(k).imag) *
               (aa(k).imag * s(k).real - aa(k).real *
               s(k).imag)) / conts
  alfa(1, k) = -2 * EXP(s(k).real) * COS(s(k).imag)
  alfa(2, k) = EXP(2 * s(k).real)
 NEXT
  IF p = 2 THEN
   beta(0, sec + 1)  = -aa(sec+1).real*(1 - EXP(s(sec+1).real))
                                            / s(sec + 1).real
   alfa(1, sec + 1) = -EXP(s(sec + 1).real)
  END IF
 SELECT CASE tip
 CASE 1:
  FOR i = 1 TO sec
    pri = 1 - alfa(1, i) * alfasola + alfa(2, i) * alfasola ^ 2
    segu = -2 * alfasola + alfa(1, i) * (1 + alfasola ^ 2) -
                                  2 * alfasola * alfa(2, i)
   alfa(2, i) = (alfasola ^ 2 - alfa(1, i) * alfasola +
                                        alfa(2, i)) / pri
   alfa(1, i) = segu / pri
   kpulso(i) = 1 / pri
    pri = -beta(0, i) * alfasola + beta(1, i) * alfasola ^ 2
    segu = beta(0, i) * (1 + alfasola ^ 2) -
                                  2 * alfasola * beta(1, i)
   beta(1, i) = (-beta(0, i) * alfasola + beta(1, i)) / pri
   beta(0, i) = segu / pri
   kpulso(i) = kpulso(i) * pri
  NEXT
   IF p = 2 THEN
     pri = -beta(0, sec + 1) * alfasola
    beta(0, sec + 1) = beta(0, sec + 1) / pri
     segu = 1 - alfa(1, sec + 1) * alfasola
    alfa(1, sec + 1) = (alfa(1, sec + 1) - alfasola) / segu
    kpulso(sec + 1) = pri / segu
   END IF
 CASE 2:
  FOR i = 1 TO sec
    pri = 1 + alfa(1, i) * alfasola + alfa(2, i) * alfasola ^ 2
    segu = -2 * alfasola - alfa(1, i) * (1 + alfasola ^ 2) -
                                  2 * alfasola * alfa(2, i)
   alfa(2, i) = (alfasola ^ 2 + alfa(1, i) * alfasola +
```

```
                                                    alfa(2, i)) / pri
  alfa(1, i) = segu / pri
  kpulso(i) = 1 / pri
   pri = beta(0, i) * alfasola + beta(1, i) * alfasola ^ 2
   segu = -beta(0, i) * (1 + alfasola ^ 2) -
                                    2 * alfasola * beta(1, i)
  beta(1, i) = (beta(0, i) * alfasola + beta(1, i)) / pri
  beta(0, i) = segu / pri
  kpulso(i) = kpulso(i) * pri
 NEXT
  IF p = 2 THEN
    pri = beta(0, sec + 1) * alfasola
   beta(0, sec + 1) = -beta(0, sec + 1) / pri
    segu = 1 + alfa(1, sec + 1) * alfasola
   alfa(1, sec + 1) = (-alfa(1, sec + 1) - alfasola) / segu
   kpulso(sec + 1) = pri / segu
  END IF
CASE 3:
  v1 = -2 * alfaso / (kso + 1)
  v2 = (1 - kso) / (kso + 1)
 FOR i = 1 TO sec
   pri = 1 + alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
   segu = v1 * (2 + alfa(1, i) * (1 + v2) +
                                         2 * v2 * alfa(2, i))
   terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) +
                             alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
  alfa(3, i) = v1 * (2 * v2 + alfa(1, i) * (1 + v2) +
                                         2 * alfa(2, i)) / pri
  alfa(4, i) = (v2 ^ 2 + alfa(1, i) * v2 + alfa(2, i)) / pri
  alfa(1, i) = segu / pri
  alfa(2, i) = terc / pri
  kpulso(i) = 1 / pri
   pri = beta(0, i) * v2 + beta(1, i) * v2 ^ 2
   segu = v1 * (beta(0, i) * (1 + v2) + 2 * v2 * beta(1, i))
   terc = (2 * v2 + v1 ^ 2) * beta(1, i) +
                             beta(0, i) * (1 + v1 ^ 2 + v2 ^ 2)
  beta(2, i) = v1 * (beta(0, i) * (1 + v2) +
                                         2 * beta(1, i)) / pri
  beta(3, i) = (beta(0, i) * v2 + beta(1, i)) / pri
  beta(0, i) = segu / pri
  beta(1, i) = terc / pri
  kpulso(i) = kpulso(i) * pri
 NEXT
  IF p = 2 THEN
    pri = 1 + alfa(1, sec + 1) * v2
    segu = v1 * (1 + alfa(1, sec + 1))
   alfa(2, sec + 1) = (v2 + alfa(1, sec + 1)) / pri
   alfa(1, sec + 1) = segu / pri
    segu = v2 * beta(0, sec + 1)
    terc = beta(0, sec + 1) * v1
   beta(1, sec + 1) = beta(0, sec + 1) / segu
   beta(0, sec + 1) = terc / segu
   kpulso(sec + 1) = segu / pri
  END IF
 CASE 4:
  v1 = -2 * alfaso * kso / (kso + 1)
  v2 = (kso - 1) / (kso + 1)
```

```
FOR i = 1 TO sec
  pri = 1 - alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
  segu = v1 * (2 - alfa(1, i) * (1 + v2) +
                                        2 * v2 * alfa(2, i))
  terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) -
                           alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
 alfa(3, i) = v1 * (2 * v2 - alfa(1, i) * (1 + v2) +
                                       2 * alfa(2, i)) / pri
 alfa(4, i) = (v2 ^ 2 - alfa(1, i) * v2 + alfa(2, i)) / pri
 alfa(1, i) = segu / pri
 alfa(2, i) = terc / pri
 kpulso(i) = 1 / pri
  pri = -beta(0, i) * v2 + beta(1, i) * v2 ^ 2
  segu = v1 * (-beta(0, i) * (1 + v2) + 2 * v2 * beta(1, i))
  terc = (2 * v2 + v1 ^ 2) * beta(1, i) -
                          beta(0, i) * (1 + v1 ^ 2 + v2 ^ 2)
 beta(2, i) = v1 * (-beta(0, i) * (1 + v2) +
                                      2 * beta(1, i)) / pri
 beta(3, i) = (-beta(0, i) * v2 + beta(1, i)) / pri
 beta(0, i) = segu / pri
 beta(1, i) = terc / pri
 kpulso(i) = kpulso(i) * pri
NEXT
 IF p = 2 THEN
    pri = 1 - alfa(1, sec + 1) * v2
    segu = v1 * (1 - alfa(1, sec + 1))
   alfa(2, sec + 1) = (v2 - alfa(1, sec + 1)) / pri
   alfa(1, sec + 1) = segu / pri
    segu = -v2 * beta(0, sec + 1)
    terc = -beta(0, sec + 1) * v1
   beta(1, sec + 1) = -beta(0, sec + 1) / segu
   beta(0, sec + 1) = terc / segu
   kpulso(sec + 1) = segu / pri
  END IF
END SELECT
END SUB



'Subrutina "transfrecuenciapulso"



SUB transfrecuenciapulso (alfa(), beta(), hi(), hp(),
                          kanaloga,  kpulso(), p AS INTEGER,
                          sec AS INTEGER, tip AS INTEGER,
                        tipana AS INTEGER, tipconv AS INTEGER)
CONST pi = 3.141592654#
  LOCATE 14, 46
  PRINT "CALCULANDO....H(z)"
 SELECT CASE tip
 CASE 1, 2:
  FOR i = 1 TO 304
     w = i * 2 * pi / 304
     real = 0
     imag = 0
```

```
    FOR k = 1 TO sec
          a = 1 + beta(0, k) * COS(w) + beta(1, k) * COS(2* w)
          b = -beta(0, k) * SIN(w) - beta(1, k) * SIN(2 * w)
          c = 1 + alfa(1, k) * COS(w) + alfa(2, k) * COS(2* w)
          d = -(alfa(1, k) * SIN(w) + alfa(2, k) * SIN(2 * w))
          real = kpulso(k) * (a*c + b* d) / (c^2 + d^2) + real
          imag = kpulso(k) *  (b* c - a * d) / (c ^ 2  + d ^ 2)
+ imag
     NEXT
     IF ((tipana = 3 OR tipana = 4) AND p = 1) THEN
         real = real + kanaloga
     END IF
    IF p = 2 THEN
          a = 1 + beta(0, sec + 1) * COS(w)
          b = -beta(0, sec + 1) * SIN(w)
          c = 1 + alfa(1, sec + 1) * COS(w)
          d = -alfa(1, sec + 1) * SIN(w)
          real = kpulso(sec+1)*(a*c + b*d)/(c^2 + d^2) + real
          imag = kpulso(sec+1)*(b*c - a*d)/(c^2 + d^2) + imag
    END IF
    IF tipconv = 3 THEN
     hp(i) = SQR(real ^ 2 + imag ^ 2)
    ELSEIF tipconv = 1 THEN
     hi(i) = SQR(real ^ 2 + imag ^ 2)
    END IF
   LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
  NEXT
   LINE (383, 256)-(461, 264), 0, BF
  CASE 3, 4:
   FOR i = 1 TO 304
      w = i * 2 * pi / 304
      real = 0
      imag = 0
     FOR k = 1 TO sec
         a = 1 + beta(0, k) * COS(w) +
                 beta(1, k) * COS(2 * w) +
            beta(2, k) * COS(3 * w) + beta(3, k) * COS(4 * w)
         b = -beta(0, k) * SIN(w) - beta(1, k)  * SIN(2 * w) -
            beta(2, k) * SIN(3 * w) - beta(3, k) * SIN(4 * w)
         c = 1  + alfa(1, k) * COS(w) + alfa(2, k) * COS(2 * w)
           + alfa(3, k) * COS(3 * w) + alfa(4, k) * COS(4 * w)
         d = -(alfa(1, k) * SIN(w) + alfa(2,  k) * SIN(2 * w) +
            alfa(3, k) * SIN(3 * w) + alfa(4, k) * SIN(4 * w))
         real = kpulso(k) * (a * c + b*d) / (c^2 + d^2) + real
         imag = kpulso(k) * (b * c - a*d) / (c^2 + d^2) + imag
      NEXT
      IF ((tipana = 3 OR tipana = 4) AND p = 1) THEN
          real = real + kanaloga
      END IF
     IF p = 2 THEN
          a = 1 + beta(0,  sec + 1) * COS(w) + beta(1, sec + 1)
 * COS(2 * w)
          b = -beta(0, sec + 1) * SIN(w) -
                                beta(1, sec + 1) * SIN(2 * w)
          c = 1 + alfa(1, sec + 1) * COS(w) +
                                alfa(2, sec + 1) * COS(2 * w)
          d = -alfa(1, sec + 1) * SIN(w) -
```

```
                                        alfa(2, sec + 1) * SIN(2 * w)
          real = kpulso(sec+1)*(a*c + b*d)/(c^2 + d^2) + real
          imag = kpulso(sec+1)*(b*c - a*d)/(c^2 + d^2) + imag
    END IF
     IF tipconv = 3 THEN
     hp(i) = SQR(real ^ 2 + imag ^ 2)
     ELSEIF tipconv = 1 THEN
     hi(i) = SQR(real ^ 2 + imag ^ 2)
     END IF
  LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
NEXT
  LINE (383, 256)-(461, 264), 0, BF
END SELECT
END SUB
```

```
'Subrutina "frecuencianat"
```

```
SUB frecuencianat
     LOCATE 14, 46
     PRINT "CALCULANDO...HT(s)"
 SELECT CASE tip
 CASE 1:
  IF tipana = 4 THEN
     FOR i = 1 TO n
       s(i).real = s(i).real * u
       s(i).imag = s(i).imag * u
     NEXT
      IF (tipana = 1 OR tipana = 2) THEN
        kanaloga = kanaloga * u ^ n
      ELSE
        FOR i = 1 TO sec
          c(i) = c(i) * u ^ 2
        NEXT
         IF p = 2 THEN
           kanaloga = kanaloga * u
         END IF
      END IF
  FOR i = 1 TO sec
    gama(1, i) = -2 * s(i).real
    gama(2, i) = (s(i).real) ^ 2 + (s(i).imag) ^ 2
  NEXT
    IF p = 2 THEN
               gama(2, sec + 1) = -s(sec + 1).real
    END IF
   FOR i = 1 TO 304
     w = i * 2 * pi / 304
     a = 1
     FOR k = 1 TO sec
          a = a * SQR((gama(2,k) - w^2)^2 + (gama(1,k) * w)^2)
     NEXT
     IF p = 2 THEN
```

```
            a = a * SQR(gama(2, sec + 1) ^ 2 + w ^ 2)
        END IF
        hat(i) = kanaloga / a
        IF (tipana = 3 OR tipana = 4) THEN
            B = 1
            FOR k = 1 TO sec
            B = B * (c(k) - w ^ 2)
            NEXT
            hat(i) = hat(i) * ABS(B)
        END IF
         LINE (382, 255)-(382 + i * 80 / 304, 265), , BF .
      NEXT
     END IF
    CASE 2:
          FOR i = 1 TO 304
                w = i * 2 * pi / 304
                a = 1
              FOR k = 1 TO n
                 a = a/SQR((s(k).real)^2 + (u/w + s(k).imag)^2)
              NEXT
            hat(i) = kanaloga * a
            LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
          IF (tipana = 3 OR tipana = 4) THEN
            v = 1
            FOR k = 1 TO sec
                v = v * ABS((-u ^ 2 / w ^ 2) + c(k))
            NEXT
            hat(i) = hat(i) * v
          END IF
          NEXT
    CASE 3:
        wo = 4 * TAN(fup * pi / fm) * TAN(fl * pi / fm)
        raiz = SQR(Bo ^ 2 + 4 * wo)
        l = (-Bo + raiz) / 2
        u = (Bo + raiz) / 2
          FOR i = 1 TO 304
            w = i * 2 * pi / 304
            a = 1
            FOR k = 1 TO n
             a = a  / SQR((s(k).real * (w^2 - u * l))^2 +
                   (s(k).imag * (w^2 - u * l) + w * (u - l))^2)
            NEXT
            hat(i) = kanaloga * a * ABS(-w ^ 2 + u * l) ^ n
           IF (tipana = 3 OR tipana = 4) THEN
            v = 1
            FOR k = 1 TO sec
             v = v * ABS((-w ^ 2 * (u - l) ^ 2) /
                                (-w ^ 2 + u * l) ^ 2 + c(k))
            NEXT
            hat(i) = hat(i) * v
           END IF
            LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
         NEXT
    CASE 4:
          wo = 4 * TAN(fup * pi / fm) * TAN(fu * pi / fm)
          raiz = SQR(Bo ^ 2 + 4 * wo)
          l = (-Bo + raiz) / 2
```

```
        u = (Bo + raiz) / 2
        FOR i = 1 TO 304
              w = i * 2 * pi / 304
              a = 1
             FOR k = 1 TO n
               a = a / SQR((s(k).imag * w*(u-1)+u*1-w^2)^2 +
                               (s(k).real * w * (u - 1)) ^ 2)
             NEXT
          hat(i) = kanaloga * a * w ^ n * (u - 1) ^ n
        IF (tipana = 3 OR tipana = 4) THEN
          v = 1
          FOR k = 1 TO sec
            v = v *  ABS(w^4 - (2 * u * 1 + c(k) * (u - 1)^2) *
                                                w^2 + (u * 1)^2)
          NEXT
            IF p = 1 THEN
               hat(i) = hat(i) * v / (w ^ n * (u - 1) ^ n)
            ELSE
               hat(i) = hat(i) * v / (w^(n-1) * (u - 1)^(n-1))
            END IF
        END IF
          LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
       NEXT
 END SELECT
 LINE (383, 256)-(461, 264), 0, BF
END SUB
```

```
'Subrutina "bilineal"
```

```
SUB bilineal
  DIM segu AS DOUBLE
  segu = 1
     FOR i = 1 TO sec
          pri = 4 + 2 * gama(1, i) + gama(2, i)
          alfa(1, i) = (-8 + 2 * gama(2, i)) / pri
          alfa(2, i) = (4 - 2 * gama(1, i) + gama(2, i)) / pri
          segu = pri * segu
     NEXT
          IF p = 2 THEN
               pri = 2 + gama(2, sec + 1)
               alfa(1, sec + 1) = (gama(2, sec + 1) - 2) / pri
               segu = pri * segu
          END IF
     kbilineal = kanaloga / segu

     IF (tipana = 3 OR tipana = 4) THEN
     FOR i = 1 TO sec
          pri = 4 + c(i)
          beta(0, i) = (-8 + 2 * c(i)) / pri
          beta(1, i) = 1
          kbilineal = pri * kbilineal
```

```
      NEXT
      END IF
END SUB
```

´Subrutina "frecuenciabilineal"

```
SUB frecuenciabilineal
  CONST pi = 3.141592654#
  LOCATE 14, 46
   PRINT "CALCULANDO....H(z)"
  FOR i = 1 TO 304
     w = i * 2 * pi / 304
     IF (tipana = 1 OR tipana = 2) THEN
         IF tip = 1 THEN
          a = kbilineal * (SQR((1 + COS(w))^2 + (SIN(w))^2))^n
         ELSEIF tip = 2 THEN
          a = kbilineal * (SQR((1 - COS(w))^2 + (SIN(w))^2))^n
         END IF
     ELSE
       a = kbilineal
       FOR k = 1 TO sec
        a = a * SQR((1 + beta(0, k) * COS(w) + beta(1,  k) *
                COS(2 * w)) ^ 2 + (beta(0, k) * SIN(w) +
                beta(1, k) * SIN(2 * w)) ^ 2)
       NEXT
         IF p = 2 THEN
          IF tip = 1 THEN
            a = a * SQR((1 + COS(w)) ^ 2 + (SIN(w)) ^ 2)
          ELSEIF tip = 2 THEN
            a = a * SQR((1 - COS(w)) ^ 2 + (SIN(w)) ^ 2)
          END IF
         END IF
     END IF
     b = 1
     FOR k = 1 TO sec
          b = b * SQR((1 + alfa(1, k) * COS(w) +
              alfa(2,k)*COS(2 * w))^2 + (alfa(1,k) * SIN(w) +
              alfa(2, k) * SIN(2 * w)) ^ 2)
     NEXT
          IF p = 2 THEN
               b = b * SQR((1 + alfa(1,sec+1) * COS(w))^2 +
                              (alfa(1, sec + 1) * SIN(w)) ^ 2)
          END IF
     hb(i) = a / B
   LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
  NEXT
   LINE (383, 256)-(461, 264), 0, BF
 END SUB
```

'Subrutina "transformacionbilineal"

```
SUB transformacionbilineal (alfa(),  alfaso, alfasola, beta(),
                            c(),  kanaloga, kbilineal, kso, l,
                            n AS INTEGER, p AS INTEGER,
                            s() AS complejo,  sec AS INTEGER,
                            tip  AS INTEGER, tipana AS INTEGER,
                            trans AS INTEGER, u)
CONST pi = 3.141592654#
SELECT CASE trans
CASE 1:
  SELECT CASE tip
  CASE 2:
  kbilineal = kanaloga * 2 ^ n
   FOR i = 1 TO sec
    pri = u ^  2 - 4 * u * s(i).real + 4 * (s(i).real ^ 2 +
                                            s(i).imag ^ 2)
    alfa(1, i) = (2*u^2 - 8*(s(i).real^2 + s(i).imag^2)) / pri
    alfa(2, i) = (u^2 + 4*u*s(i).real + 4 * (s(i).real^2 +
                                        s(i).imag ^ 2)) / pri
    kbilineal = kbilineal / pri
   NEXT
  IF p = 2 THEN
   pri = u - 2 * s(sec + 1).real
   alfa(1, sec + 1) = (u + 2 * s(sec + 1).real) / pri
   kbilineal = kbilineal / pri
  END IF
  IF (tipana = 3 OR tipana = 4) THEN
   FOR i = 1 TO sec
    pri = u ^ 2 + 4 * c(i)
    beta(0, i) = (2 * u ^ 2 - 8 * c(i)) / pri
    beta(1, i) = 1
    kbilineal = pri * kbilineal
   NEXT
    IF p = 1 THEN
     kbilineal = kbilineal / 2 ^ n
    ELSE
     kbilineal = kbilineal / 2 ^ (n - 1)
    END IF
   END IF
   CALL frecuenciabilineal
   CASE 3:
    kbilineal = kanaloga
    FOR i = 1 TO sec
     sr = s(i).real
     si = s(i).imag
     pri = 4 * (u - 1)^2 - 4 * (u - 1) * (4 + u * 1) * sr +
                                  (4 + u * 1)^2 * (sr^2 + si^2)
     alfa(1, i) = (-2 * (4 + u * 1) * (8 - 2 * u * 1) *
                      (sr ^ 2 + si ^ 2) + 4 * sr * (u - 1) *
                                          (8 - 2 * u * 1)) / pri
     alfa(2, i) = (-8 * (u - 1)^2 + (2 * (4 + u*1)^2 +
                          (8 - 2*u*1)^2) * (sr^2 + si^2)) / pri
     alfa(3, i) = (-2 * (4 + u * 1) * (8 - 2 * u * 1) *
                      (sr ^ 2 + si ^ 2) - 4 * sr * (u - 1) *
```

```
                                    (8 - 2 * u * 1)) / pri
     alfa(4, i) = (4 * (u - 1) ^ 2 + 4 * (u - 1) *
                       (4 + u * 1) * sr + (4 + u * 1) ^ 2 *
                                    (sr ^ 2 + si ^ 2)) / pri
  kbilineal = kbilineal / pri
    NEXT
     segu = (4 + u * 1) ^ 2
     beta(0, 1) = (-64 + 4 * (u * 1) ^ 2) / segu
     beta(1, 1) = (96 - 16 * u * 1 + 6 * (u * 1) ^ 2) / segu
     beta(2, 1) = beta(0, 1)
     beta(3, 1) = 1
     kbilineal = kbilineal * segu ^ sec
    IF (tipana = 3 OR tipana = 4) THEN
     FOR i = 1 TO sec
      pri = 16 * c(i) + 4 *((u - 1)^2 + 2 * c(i) * u * 1) +
                                         c(i) * (u * 1) ^ 2
      beta(0, i) = (-64 * c(i) + 4 * c(i) * (u * 1) ^ 2) / pri
      beta(1, i) = (96 * c(i) - 8 * (2 * u * 1 * c(i) +
                    (u - 1) ^ 2) + 6 * c(i) * (u * 1) ^ 2) / pri
      beta(2, i) = beta(0, i)
      beta(3, i) = 1
      kbilineal = kbilineal * pri
     NEXT
     kbilineal = kbilineal / segu ^ sec
    END IF
    IF p = 2 THEN
     sr = s(sec + 1).real
     pri = -4 * sr + 2 * (u - 1) - sr * u * 1
     alfa(1, sec + 1) = (8 * sr - 2 * sr * u * 1) / pri
     alfa(2, sec + 1) = (-4*sr - 2*(u - 1) - sr * u * 1) / pri
     segu = 4 + u * 1
     beta(0, sec + 1) = (-8 + 2 * u * 1) / segu
     beta(1, sec + 1) = 1
     kbilineal = kbilineal * segu / pri
    END IF
   CASE 4:
    kbilineal = kanaloga * (2 * (u - 1)) ^ n
    FOR i = 1 TO sec
     sr = s(i).real
     si = s(i).imag
     pri = (4 + u * 1)^2 - (4 + u*1)*(u-1)*4*sr +
                                    4*(sr^2 + si^2) * (u-1)^2
     alfa(1, i) = (-8 + 2 * u * 1) * ((4 + u * 1) * 2 -
                                      4 * (u - 1) * sr) / pri
     alfa(2, i) = ((4 + u * 1) ^ 2 - 8 * (u - 1) ^ 2 *
                   (sr ^ 2 + si ^ 2) + (4 + u * 1) ^ 2 +
                                 (-8 + 2 * u * 1) ^ 2) / pri
     alfa(3, i) = (-8 + 2 * u * 1) * ((4 + u * 1) * 2 +
                                      4 * (u - 1) * sr) / pri
     alfa(4, i) = ((4 + u * 1) ^ 2 + (4 + u * 1) *
                   (u - 1) * 4 * sr + 4 * (sr^2 + si^2) *
                                            (u - 1) ^ 2) / pri
     kbilineal = kbilineal / pri
    NEXT
    IF p = 2 THEN
     pri = 4 - 2 * s(sec + 1).real * (u - 1) + u * 1
     alfa(1, sec + 1) = (-8 + 2 * u * 1) / pri
```

```
    alfa(2, sec + 1) = (4 + 2*s(sec+1).real*(u-1) + u*1)/ pri
    kbilineal = kbilineal / pri
   END IF
   IF (tipana = 3 OR tipana = 4) THEN
    FOR i = 1 TO sec
     pri = 16 + 4 * (2 * u * 1 + c(i) * (u-1)^2) + (u * 1)^ 2
     beta(0, i) = (-64 + 4 * (u * 1) ^ 2) / pri
     beta(1, i) = (96 - 8 * (2*u*1 + c(i)*(u - 1)^2) +
                                        6 * (u * 1) ^ 2) / pri
     beta(2, i) = (-64 + 4 * (u * 1) ^ 2) / pri
     beta(3, i) = (16 + 4 * (2*u*1 + c(i)*(u - 1)^2) +
                                            (u * 1) ^ 2) / pri
     kbilineal = kbilineal * pri
    NEXT
    IF p = 1 THEN
     kbilineal = kbilineal / (2 * (u - 1)) ^ n
    ELSE
     kbilineal = kbilineal / (2 * (u - 1)) ^ (n - 1)
    END IF
   END IF
  END SELECT

CASE 2:
  SELECT CASE tip
  CASE 1:
   kbilineal = kbilineal * (1 - alfasola) ^ n
   FOR i = 1 TO sec
    pri = 1 - alfa(1, i) * alfasola + alfa(2,i) * alfasola^ 2
    segu = -2 * alfasola + alfa(1,i)*(1 + alfasola^2) -
                                      2 * alfasola * alfa(2,i)
    alfa(2, i) = (alfasola^2 - alfa(1,i)*alfasola +
                                            alfa(2, i)) / pri
    alfa(1, i) = segu / pri
    kbilineal = kbilineal / pri
   NEXT
   IF p = 2 THEN
    pri = 1 - alfa(1, sec + 1) * alfasola
    alfa(1, sec + 1) = (alfa(1, sec + 1) - alfasola) / pri
    kbilineal = kbilineal / pri
   END IF
   IF (tipana = 3 OR tipana = 4) THEN
    FOR i = 1 TO sec
     pri = 1 - beta(0,i) * alfasola + beta(1,i) * alfasola^ 2
     segu = -2*alfasola + beta(0,i) * (1 + alfasola^2) -
                                      2 * alfasola * beta(1, i)
     beta(1, i) = (alfasola^2 - beta(0,i)*alfasola +
                                            beta(1, i)) / pri
     beta(0, i) = segu / pri
     kbilineal = kbilineal * pri
    NEXT
    IF p = 1 THEN
      kbilineal = kbilineal / (1 - alfasola) ^ n
    ELSE
      kbilineal = kbilineal / (1 - alfasola) ^ (n - 1)
    END IF
   END IF
   CALL frecuenciabilineal
```

```
CASE 2:
 kbilineal = kbilineal * (1 + alfasola) ^ n
 FOR i = 1 TO sec
  pri = 1 + alfa(1,i) * alfasola + alfa(2,i) * alfasola ^ 2
  segu = -2*alfasola - alfa(1,i)*(1 + alfasola^2) -
                                    2 * alfasola * alfa(2, i)
  alfa(2, i) = (alfasola^2 + alfa(1,i)*alfasola +
                                          alfa(2, i)) / pri
  alfa(1, i) = segu / pri
  kbilineal = kbilineal / pri
 NEXT
 IF p = 2 THEN
  pri = 1 + alfa(1, sec + 1) * alfasola
  alfa(1, sec + 1) = (-alfa(1, sec + 1) - alfasola) / pri
  kbilineal = kbilineal / pri
 END IF
 IF (tipana = 3 OR tipana = 4) THEN
  FOR i = 1 TO sec
   pri = 1 + beta(0,i) * alfasola + beta(1,i) * alfasola^ 2
   segu = -2*alfasola - beta(0,i)*(1 + alfasola^2) -
                                    2 * alfasola * beta(1, i)
   beta(1, i) = (alfasola^2 + beta(0,i)*alfasola +
                                          beta(1, i)) / pri
   beta(0, i) = segu / pri
   kbilineal = kbilineal * pri
  NEXT
  IF p = 1 THEN
   kbilineal = kbilineal / (1 + alfasola) ^ n
  ELSE
   kbilineal = kbilineal / (1 + alfasola) ^ (n - 1)
  END IF
 END IF
 CALL frecuenciabilineal
 CASE 3:
   v1 = -2 * alfaso / (kso + 1)
   v2 = (1 - kso) / (kso + 1)
   FOR i = 1 TO sec
        pri = 1 + alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
        segu = 2 * v1 + alfa(1, i) * v1 * (1 + v2) +
                                     2 * v1 * v2 * alfa(2, i)
        terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) +
                        alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
        alfa(3, i) = (2 * v1 * v2 + alfa(1, i) * v1 *
                     (1 + v2) + 2 * v1 * alfa(2, i)) / pri
        alfa(4, i) = (v2^2 + alfa(1,i)*v2 + alfa(2,i)) / pri
        alfa(1, i) = segu / pri
        alfa(2, i) = terc / pri
        kbilineal = kbilineal / pri
   NEXT
   IF (tipana = 1 OR tipana = 2) THEN
        segu = (1 + v2) ^ 2
        beta(0, 1) = (4 * v1 * (1 + v2)) / segu
        beta(1, 1) = (4 * v1 ^ 2 + 2 * (1 + v2) ^ 2) / segu
        beta(2, 1) = beta(0, 1)
        beta(3, 1) = 1
        kbilineal = kbilineal * segu ^ sec
   ELSE
```

```
    FOR i = 1 TO sec
      pri = 1 + beta(0, i) * v2 + beta(1, i) * v2 ^ 2
      segun = 2 * v1 + beta(0, i) * v1 * (1 + v2) +
                                     2 * v1 * v2 * beta(1, i)
      terc = (2 * v2 + v1 ^ 2) * (1 + beta(1, i)) +
                                 beta(0, i)*(1 + v1^2 + v2^2)
      beta(2, i) = (2 * v1 * v2 + beta(0,i) *
                           v1*(1 + v2) + 2*v1*beta(1,i)) / pri
      beta(3, i) = (v2^2 + beta(0,i)*v2 + beta(1,i)) / pri
      beta(0, i) = segun / pri
      beta(1, i) = terc / pri
      kbilineal = kbilineal * pri
    NEXT
  END IF
      IF p = 2 THEN
         pri = 1 + alfa(1, sec + 1) * v2
         segu = v1 * (1 + alfa(1, sec + 1))
         alfa(2, sec + 1) = (v2 + alfa(1, sec + 1)) / pri
         alfa(1, sec + 1) = segu / pri
         segu = 1 + v2
         beta(0, sec + 1) = 2 * v1 / segu
         beta(1, sec + 1) = 1
         kbilineal = kbilineal * segu / pri
      END IF
CASE 4:
  v1 = -2 * alfaso * kso / (kso + 1)
  v2 = (kso - 1) / (kso + 1)
  kbilineal = kbilineal * (1 - v2) ^ n
  FOR i = 1 TO sec
      pri = 1 - alfa(1, i) * v2 + alfa(2, i) * v2 ^ 2
      segu = 2*v1 - alfa(1,i)*v1*(1 + v2) +
                                     2 * v1 * v2 * alfa(2, i)
      terc = (2 * v2 + v1 ^ 2) * (1 + alfa(2, i)) -
                             alfa(1, i) * (1 + v1 ^ 2 + v2 ^ 2)
      alfa(3, i) = (2 * v1 * v2 - alfa(1, i) *
                           v1*(1 + v2) + 2*v1*alfa(2,i)) / pri
      alfa(4, i) = (v2^2 - alfa(1,i)*v2 + alfa(2,i)) / pri
      alfa(1, i) = segu / pri
      alfa(2, i) = terc / pri
      kbilineal = kbilineal / pri
  NEXT
    IF p = 2 THEN
       pri = 1 - alfa(1, sec + 1) * v2
       segu = v1 * (1 - alfa(1, sec + 1))
       alfa(2, sec + 1) = (v2 - alfa(1, sec + 1)) / pri
       alfa(1, sec + 1) = segu / pri
       kbilineal = kbilineal / pri
    END IF
  IF (tipana = 3 OR tipana = 4) THEN
    FOR i = 1 TO sec
      pri = 1 - beta(0, i) * v2 + beta(1, i) * v2 ^ 2
      segu = 2 * v1 - beta(0, i) * v1 * (1 + v2) +
                                     2 * v1 * v2 * beta(1, i)
      terc = (2 * v2 + v1 ^ 2) * (1 + beta(1, i)) -
                             beta(0, i) * (1 + v1 ^ 2 + v2 ^ 2)
      beta(2, i) = (2 * v1 * v2 - beta(0, i) *
                           v1*(1 + v2) + 2*v1*beta(1,i)) / pri
```

```
            beta(3, i) = (v2^2 - beta(0,i)*v2 + beta(1,i)) / pri
            beta(0, i) = segu / pri
            beta(1, i) = terc / pri
            kbilineal = kbilineal * pri
          NEXT
           IF p = 1 THEN
            kbilineal = kbilineal / (1 - v2) ^ n
           ELSE
            kbilineal = kbilineal / (1 - v2) ^ (n - 1)
           END IF
        END IF
      END SELECT
  END SELECT
  END SUB




  'Subrutina "transfrecuenciabilineal"


  SUB transfrecuenciabilineal (alfa(),  beta(), hb(), kbilineal,
                                p AS INTEGER, sec AS INTEGER,
                              tip AS INTEGER, tipana AS INTEGER)
  CONST pi = 3.141592654#
  LOCATE 14, 46
  PRINT "CALCULANDO....H(z)"
  SELECT CASE tip
  CASE 3:
    FOR i = 1 TO 304
       w = i * 2 * pi / 304
       IF (tipana = 1 OR tipana = 2) THEN
          bb = 1 + beta(0,1) * COS(w)  + beta(1,1) * COS(2 * w) +
                 beta(2,1) * COS(3 * w) + beta(3,1) * COS(4 * w)
          cc = beta(0,1) * SIN(w) + beta(1,1) * SIN(2 * w) +
                 beta(2,1) * SIN(3 * w) + beta(3,1) * SIN(4 * w)
          a = kbilineal * (SQR(bb ^ 2 + cc ^ 2)) ^ sec
       ELSE
          a = kbilineal
         FOR k = 1 TO sec
          bb = 1 + beta(0,k) * COS(w) + beta(1,k) * COS(2 * w) +
               beta(2, k) * COS(3 * w) + beta(3, k) * COS(4 * w)
          cc = beta(0, k) * SIN(w) + beta(1, k) * SIN(2 * w) +
               beta(2, k) * SIN(3 * w) + beta(3, k) * SIN(4 * w)
          a = a * SQR(bb ^ 2 + cc ^ 2)
         NEXT
       END IF
       B = 1
       FOR k = 1 TO sec
          bb = 1 + alfa(1,k) * COS(w) + alfa(2,k) * COS(2 * w) +
                 alfa(3,k) * COS(3 * w) + alfa(4,k) * COS(4 * w)
          cc = alfa(1,k) * SIN(w) + alfa(2,k) * SIN(2 * w) +
                 alfa(3,k) * SIN(3 * w) + alfa(4,k) * SIN(4 * w)
          b = b * SQR(bb ^ 2 + cc ^ 2)
       NEXT
          IF p = 2 THEN
```

```
          a = a * SQR((1 + beta(0, sec + 1) * COS(w) +
                    beta(1, sec +  1) * COS(2 * w)) ^ 2 +
                       (beta(0, sec + 1) * SIN(w) +
                          beta(1, sec + 1) * SIN(2 * w)) ^ 2)
          b = b * SQR((1 + alfa(1, sec + 1) * COS(w) +
                    alfa(2, sec + 1) * COS(2 * w)) ^ 2 +
                       (alfa(1, sec + 1) * SIN(w) +
                          alfa(2, sec + 1) * SIN(2 * w)) ^ 2)
        END IF
     hb(i) = a / b
     LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
 NEXT
     LINE (383, 256)-(461, 264), 0, BF
CASE 4:
  FOR i = 1 TO 304
     w = i * 2 * pi / 304
     IF (tipana = 1 OR tipana = 2) THEN
       a = kbilineal*(SQR((1 - COS(2*w))^2 + (SIN(2*w))^2))^ n
     ELSE
       a = kbilineal
      FOR k = 1 TO sec
       bb = 1 + beta(0,k) * COS(w) + beta(1,k) * COS(2 * w) +
               beta(2,k) * COS(3 * w) + beta(3,k) * COS(4 * w)
       cc = beta(0, k) * SIN(w) + beta(1, k) * SIN(2 * w) +
              beta(2, k) * SIN(3 * w) + beta(3, k) * SIN(4 * w)
       a = a * SQR(bb ^ 2 + cc ^ 2)
      NEXT
        IF p = 2 THEN
          a = a * SQR((1 - COS(2 * w)) ^ 2 + (SIN(2 * w)) ^ 2)
        END IF
     END IF
     b = 1
     FOR k = 1 TO sec
       bb = 1 + alfa(1,k) * COS(w) + alfa(2,k) * COS(2 * w) +
               alfa(3,k) * COS(3 * w) + alfa(4,k) * COS(4 * w)
       cc = alfa(1,k) * SIN(w) + alfa(2,k) * SIN(2 * w) +
               alfa(3,k) * SIN(3 * w) + alfa(4,k) * SIN(4 * w)
       b = b * SQR(bb ^ 2 + cc ^ 2)
     NEXT
       IF p = 2 THEN
         b = b * SQR((1 + alfa(1, sec + 1) * COS(w) +
                  alfa(2, sec + 1) * COS(2 * w)) ^ 2 +
                     (alfa(1, sec + 1) * SIN(w) +
                          alfa(2, sec + 1) * SIN(2 * w)) ^ 2)
        END IF
     hb(i) = a / b
     LINE (382, 255)-(382 + i * 80 / 304, 265), , BF
  NEXT
  LINE (383, 256)-(461, 264), 0, BF
  END SELECT
 END SUB
```

´Subrutina "salidatotal"

```
SUB salidatotal
DIM rs AS STRING * 2
 OPEN sali FOR RANDOM AS #1 LEN = 2
 OPEN entr FOR RANDOM AS #2 LEN = 2
        IF max <> 0 THEN
         CALL porcentaje
         LOCATE 14, 46
         PRINT "CALCULANDO....y(n)"
        END IF
        x(1) = 0
        x(2) = 0
        x(3) = 0
        x(4) = 0
        FOR k = 0 TO (sec + 1)
          y(1, k) = 0
          y(2, k) = 0
          y(3, k) = 0
          y(4, k) = 0
        NEXT
        nreg = 0
     DO
         FIELD 2, 2 AS r$
         nreg = nreg + 1
         GET #2, nreg
         x(5) = .8 * CVI(r$) / 2048 / max
        SELECT CASE tipconv
         CASE 1, 3 ´Selección del método invarianza de impulso
           CALL salidapulso
         CASE 2    ´Selección del método transformada bilineal
           CALL salidabilineal
        END SELECT
               yy = FIX(y(5, 0) * 2048)
               rs = MKI$(yy)
               PUT #1, nreg, rs
     LOOP WHILE NOT EOF(2)
  CLOSE #1
  CLOSE #2
 END SUB
```

´Subrutina "salidabilineal"

```
SUB salidabilineal
SELECT CASE tip
CASE 1, 2:
 IF (tipana = 1 OR tipana = 2) THEN
  y(5, 0) = 0
  IF tip = 1 THEN
```

```
 y(5, 1) = kbilineal * (x(5) + 2 * x(4) + x(3)) -
                alfa(1, 1) * y(4, 1) - alfa(2, 1) * y(3, 1)
 FOR k = 2 TO sec
  y(5, k) = y(5, k - 1) + 2 * y(4, k - 1) +  y(3, k - 1) -
                alfa(1, k) * y(4, k) - alfa(2, k) * y(3, k)
 NEXT
 y(5, 0) = y(5, sec)
ELSEIF tip = 2 THEN
 y(5, 1) = kbilineal * (x(5) - 2 * x(4) + x(3)) -
                alfa(1, 1) * y(4, 1) - alfa(2, 1) * y(3, 1)
 FOR k = 2 TO sec
  y(5, k) = y(5, k - 1) - 2 * y(4, k - 1) + y(3, k - 1) -
                alfa(1, k) * y(4, k) - alfa(2, k) * y(3, k)
 NEXT
  y(5, 0) = y(5, sec)
END IF
ELSE
 y(5, 0) = 0
 y(5, 1) = kbilineal * (x(5) + beta(0, 1) * x(4) +
            beta(1, 1) * x(3)) - alfa(1, 1) * y(4, 1) -
             alfa(2, 1) * y(3, 1)
 FOR k = 2 TO sec
 y(5, k) = y(5, k - 1) + beta(0, k) * y(4, k - 1) +
            beta(1, k) * y(3,k - 1) - alfa(1, k) * y(4, k) -
             alfa(2, k) * y(3, k)
 NEXT
 y(5, 0) = y(5, sec)
END IF
IF p = 2 THEN
 y(5, sec + 1) = y(5, sec) + y(4, sec) -
                               alfa(1, sec + 1) * y(4, sec + 1)
 y(5, 0) = y(5, sec + 1)
END IF
 reb = nreg / maximo * 160
 LINE (382, 255)-(382 + reb, 265), , BF
 x(3) = x(4)
 x(4) = x(5)
FOR k = 0 TO (sec + 1)
 y(3, k) = y(4, k)
 y(4, k) = y(5, k)
NEXT
CASE 3:
   IF (tipana = 1 OR tipana = 2) THEN
     y(5, 0) = 0
     y(5, 1)  = kbilineal * (x(5) + beta(0, 1) * x(4) +
                beta(1, 1) * x(3) + beta(2, 1) * x(2) +
                beta(3, 1) * x(1))
     y(5, 1) = y(5, 1) - alfa(1, 1) * y(4, 1) -
                alfa(2, 1) * y(3, 1) - alfa(3, 1) *  y(2, 1) -
                 alfa(4, 1) * y(1, 1)
      FOR k = 2 TO sec
       y(5, k) = y(5, k - 1) + beta(0,  1) * y(4, k - 1) +
                 beta(1,1)*y(3,k-1) + beta(2,1)  * y(2,k-1) +
                 beta(3, 1) * y(1, k - 1)
       y(5, k) = y(5, k) - alfa(1, k) * y(4, k) -
                alfa(2,  k) * y(3, k) - alfa(3, k) * y(2, k) -
                alfa(4, k) * y(1, k)
```

```
NEXT
  y(5, 0) = y(5, sec)
ELSE
    y(5, 0) = 0
    y(5, 1) = kbilineal * (x(5) + beta(0, 1) * x(4) +
                beta(1, 1) * x(3) + beta(2, 1)  * x(2) +
                beta(3, 1) * x(1))
    y(5, 1) = y(5, 1) - alfa(1, 1) * y(4, 1) -
              alfa(2, 1) * y(3, 1) - alfa(3, 1) * y(2, 1) -
                alfa(4, 1) * y(1, 1)
    FOR k = 2 TO sec
       y(5, k) = y(5, k - 1) + beta(0, k) * y(4, k - 1) +
                  beta(1,k)*y(3,k-1) + beta(2,k)*y(2,k-1) +
                   beta(3, k) * y(1, k - 1)
       y(5, k) = y(5, k) - alfa(1, k) * y(4, k) -
                  alfa(2,k) * y(3,k) - alfa(3,k) * y(2,k) -
                   alfa(4, k) * y(1, k)
      NEXT
      y(5, 0) = y(5, sec)
    END IF
      IF p = 2 THEN
       y(5,sec+1) = y(5,sec) + beta(0,sec+1) * y(4,sec) +
                           beta(1, sec + 1) *  y(3, sec) -
                            alfa(1, sec +  1) * y(4, sec + 1) -
                               alfa(2, sec + 1) * y(3, sec + 1)
       y(5, 0) = y(5, sec + 1)
      END IF
       reb = nreg / maximo * 160
       LINE (382, 255)-(382 + reb, 265), , BF
  x(1) = x(2)
  x(2) = x(3)
  x(3) = x(4)
  x(4) = x(5)
  FOR k = 0 TO (sec + 1)
   y(1, k) = y(2, k)
   y(2, k) = y(3, k)
   y(3, k) = y(4, k)
   y(4, k) = y(5, k)
  NEXT

CASE 4:
     IF (tipana = 1 OR tipana = 2) THEN
          y(5, 0) = 0
          y(5, 1) = kbilineal * (x(5) - 2 * x(3) + x(1))  -
              alfa(1, 1) * y(4, 1) - alfa(2, 1) * y(3,  1) -
                alfa(3, 1) * y(2, 1) - alfa(4, 1) * y(1, 1)
             FOR k = 2 TO sec
              y(5, k) = y(5, k - 1) - 2 * y(3, k - 1) +
                     y(1, k - 1) -  alfa(1, k) * y(4, k) -
                     alfa(2, k) * y(3, k) - alfa(3, k) *
                           y(2, k) - alfa(4, k) * y(1, k)
             NEXT
             y(5, 0) = y(5, sec)
   ELSE
      y(5, 0) = 0
      y(5, 1) = kbilineal * (x(5) + beta(0, 1) * x(4)  +
                beta(1, 1) * x(3) + beta(2, 1) * x(2)  +
```

```
                                          beta(3, 1) * x(1))
        y(5, 1) = y(5, 1) - alfa(1, 1) * y(4, 1) -
                alfa(2, 1) * y(3, 1) - alfa(3, 1) * y(2, 1) -
                                         alfa(4, 1) * y(1, 1)
        FOR k = 2 TO sec
        y(5, k) = y(5, k - 1) + beta(0, k) * y(4, k - 1)  +
                  beta(1, k)  * y(3, k - 1) + beta(2, k) *
                  y(2, k - 1) + beta(3, k) * y(1, k - 1)
        y(5, k) = y(5, k) - alfa(1, k) * y(4, k) -
                  alfa(2, k) * y(3, k) - alfa(3, k) *
                  y(2, k) - alfa(4, k) * y(1, k)
        NEXT
        y(5, 0) = y(5, sec)
      END IF
        IF p = 2 THEN
        y(5, sec + 1) = y(5, sec) - y(3, sec) -
                             alfa(1, sec + 1) * y(4, sec + 1) -
                              alfa(2, sec + 1) * y(3, sec + 1)
        y(5, 0) = y(5, sec + 1)
        END IF
        reb = nreg / maximo * 160
        LINE (382, 255)-(382 + reb, 265), , BF
    x(1) = x(2)
    x(2) = x(3)
    x(3) = x(4)
    x(4) = x(5)
    FOR k = 0 TO (sec + 1)
     y(1, k) = y(2, k)
     y(2, k) = y(3, k)
     y(3, k) = y(4, k)
     y(4, k) = y(5, k)
    NEXT
END SELECT
END SUB


´Subrutina "salidapulso"


SUB salidapulso
SELECT CASE tip
CASE 1, 2:
  y(5, 0) = 0
    FOR k = 1 TO sec
     y(5, k) = kpulso(k) * (x(5) + beta(0, k) * x(4) +
               beta(1, k) * x(3)) - alfa(1, k)  * y(4, k) -
               alfa(2, k) * y(3, k)
     y(5, 0) = y(5, k) + y(5, 0)
    NEXT
  IF ((tipana = 3 OR tipana = 4) AND p = 1) THEN
     y(5, 0) = y(5, 0) + x(5) * kanaloga
  END IF
    IF p = 2 THEN
     y(5, sec + 1) = kpulso(k) * (x(5) +
                          beta(0, sec + 1) * x(4)) -
```

```
                            alfa(1, sec + 1) * y(4, sec + 1)
     y(5, 0) = y(5, 0) + y(5, sec + 1)
    END IF
    rep = nreg / maximo * 160
  LINE (382, 255)-(382 + rep, 265), , BF
     x(3) = x(4)
     x(4) = x(5)
     FOR k = 0 TO (sec + 1)
      y(3, k) = y(4, k)
      y(4, k) = y(5, k)
     NEXT
CASE 3, 4:
  y(5, 0) = 0
    FOR k = 1 TO sec
     y(5, k) = kpulso(k) * (x(5) + beta(0, k) * x(4) +
                beta(1, k) * x(3) + beta(2, k) * x(2) +
                 beta(3, k) * x(1))
     y(5, k) = y(5, k) - alfa(1, k) * y(4, k) -
                alfa(2, k) * y(3, k)  - alfa(3, k) * y(2, k) -
                 alfa(4, k) * y(1, k)
     y(5, 0) = y(5, k) + y(5, 0)
    NEXT
   IF ((tipana = 3 OR tipana = 4) AND p = 1) THEN
     y(5, 0) = y(5, 0) + x(5) * kanaloga
   END IF
     IF p = 2 THEN
      y(5, sec + 1) = kpulso(k) * (x(5) + beta(0,sec+1)*x(4) +
                  beta(1, sec + 1) * x(3)) - alfa(1, sec + 1) *
              y(4, sec + 1) - alfa(2, sec + 1) * y(3, sec + 1)
      y(5, 0) = y(5, 0) + y(5, sec + 1)
     END IF
      rep = nreg / maximo * 160
   LINE (382, 255)-(382 + rep, 265), , BF
      x(1) = x(2)
      x(2) = x(3)
      x(3) = x(4)
      x(4) = x(5)
      FOR k = 0 TO (sec + 1)
       y(1, k) = y(2, k)
       y(2, k) = y(3, k)
       y(3, k) = y(4, k)
       y(4, k) = y(5, k)
      NEXT
END SELECT
END SUB
```

c) SUBRUTINAS DE PRESENTACION

```
'Subrutina "dibujo"


SUB dibujo
CLS
CALL caratula(tip, tipana, tipconv, trans)
LINE (319, 45)-(319, 440)
LINE (2, 47)-(317, 438), , BF
     FOR i = 0 TO 10
          LINE (20 + 28 * i, 86)-(20 + 28 * i, 246), 7
     NEXT
     FOR i = 0 TO 5
          LINE (20, 246 - 32 * i)-(300, 246 - 32 * i), 7
     NEXT
     'Dibujo del 1
          PSET (15, 117), 0
          PSET (15, 120), 0
          PSET (16, 116), 0
          PSET (16, 117), 0
          PSET (16, 118), 0
          PSET (16, 119), 0
          PSET (16, 120), 0
          PSET (17, 120), 0
     FOR i = 0 TO 10
          LINE (20 + 28 * i, 276)-(20 + 28 * i, 436), 7
     NEXT
     FOR i = 0 TO 5
          LINE (20, 436 - 32 * i)-(300, 436 - 32 * i), 7
     NEXT
     'Dibujo del 1
          PSET (15, 307), 0
          PSET (15, 310), 0
          PSET (16, 306), 0
          PSET (16, 307), 0
          PSET (16, 308), 0
          PSET (16, 309), 0
          PSET (16, 310), 0
          PSET (17, 310), 0
     'Dibujo del 0
          PSET (19, 251), 0
          PSET (19, 252), 0
          PSET (19, 253), 0
          PSET (20, 250), 0
          PSET (20, 254), 0
          PSET (21, 251), 0
          PSET (21, 252), 0
          PSET (21, 253), 0
     'Dibujo del 2 pi'
          PSET (297, 251), 0
          PSET (297, 253), 0
          PSET (297, 254), 0
          PSET (298, 250), 0
```

```
        PSET (298, 252), 0
        PSET (298, 254), 0
        PSET (299, 251), 0
        PSET (299, 254), 0
        PSET (300, 252), 0
        PSET (301, 252), 0
        PSET (301, 253), 0
        PSET (301, 254), 0
        PSET (302, 252), 0
        PSET (303, 252), 0
        PSET (303, 253), 0
        PSET (303, 254), 0
        PSET (304, 252), 0
      'Dibujo de pi
        PSET (158, 251), 0
        PSET (159, 251), 0
        PSET (159, 252), 0
        PSET (159, 253), 0
        PSET (160, 251), 0
        PSET (161, 251), 0
        PSET (161, 252), 0
        PSET (161, 253), 0
        PSET (162, 251), 0
LOCATE 5, 5
PRINT "RESPUESTA DE FRECUENCIA ANALOGICA"
LOCATE 17, 5
PRINT "RESPUESTA DE FRECUENCIA DIGITAL"
          KEY(10) OFF
          WINDOW (0, 0)-(304, 1.25)
          FOR i = 1 TO 303
              VIEW (20, 86)-(300, 246)
               PSET (i, ha(i)), 9
               LINE -(i + 1, ha(i + 1)), 9
               IF trans = 1 AND (tip <> 1 OR tipana = 4) THEN
                    PSET (i - 1, hat(i)), 9
                    LINE -(i, hat(i + 1)), 9
               END IF
              VIEW (20, 276)-(300, 436)
              IF tipconv = 1 THEN
                    PSET (i - 1, hi(i)), 12
                     LINE -(i, hi(i + 1)), 12
              ELSEIF tipconv = 2 THEN
                    PSET (i - 1, hb(i)), 12
                    LINE -(i, hb(i + 1)), 12
              ELSEIF tipconv = 3 THEN
                    PSET (i - 1, hp(i)), 12
                    LINE -(i, hp(i + 1)), 12
              END IF
              VIEW
          NEXT
       WINDOW
       KEY(10) ON
LOCATE 4, 50
PRINT "SEÑAL A FILTRAR"
LOCATE 16, 50
PRINT "SEÑAL FILTRADA"
LOCATE 4, 70
```

```
PRINT " N = "; n
CALL filtraje(max, maximo, nn, entr, sali, x(), y())
END SUB
```

´Subrutina "filtraje"

```
SUB filtraje (max, maximo, nn AS INTEGER, entr AS STRING,  sali
                AS STRING, x(), y())
  REDIM cuadroa(1100)
  REDIM cuadrob(1100)
  maxim = INT(maximo / 600 - 1)
  LINE (444, 429)-(526, 439), , B
  LINE (445 + nn * 70 / maxim, 430)-(455 + nn * 70 / maxim,
        438), , BF
  x(4) = 0
  y(4, 0) = 0
  DO
  IF RIGHT$(v$, 1) = CHR$(75) THEN
   IF nn <> 0 THEN
      LINE (445 + nn * 70 / maxim, 430)-(455 + nn * 70 /
                                             maxim, 438), 7, BF
      nn = nn - 1
   END IF
  ELSEIF RIGHT$(v$, 1) = CHR$(77) THEN
   IF nn <> maxim THEN
      LINE (445 + nn * 70 / maxim, 430)-(455 + nn * 70 /
                                             maxim, 438), 7, BF
      nn = nn + 1
   END IF
  END IF
  LINE (445 + nn * 70 / maxim, 430)-(455 + nn * 70 / maxim,
                                                  438), , BF
     LOCATE 28, 43
     PRINT "CUADRO#"; 1 + nn
     GET (336, 432)-(423, 444), cuadroa
     PUT (336, 432), cuadroa, PRESET
     LOCATE 28, 69
     PRINT "CUADRO#"; 2 + nn
     GET (544, 432)-(631, 444), cuadrob
     PUT (544, 432), cuadrob, PRESET
   LINE (330, 260)-(636, 420), 0, BF
   LINE (330, 70)-(636, 230), 0, BF
   LINE (330, 340)-(634, 340)
   LINE (330, 150)-(634, 150)
   LINE (483, 260)-(483, 420)
   LINE (483, 70)-(483, 230)
   OPEN entr FOR RANDOM AS #2 LEN = 2
    FIELD 2, 2 AS r$
   OPEN sali FOR RANDOM AS #1 LEN = 2
    FIELD 1, 2 AS rs$
    KEY(10) OFF
    WINDOW (1, -1)-(600, 1)
```

```
  nreg = nn * 300
    DO
       nreg = nreg + 1
       GET #2, nreg
       x(5) = .8 * CVI(r$) / 2048 / max
       GET #1, nreg
       y(5, 0) = CVI(rs$) / 2048
            VIEW (331, 71)-(630, 229)
            PSET (nreg - nn * 300, x(4))
            LINE -(nreg - nn * 300 + 1, x(5))
            VIEW
            VIEW (332, 261)-(634, 419)
            PSET (nreg - nn * 300, y(4, 0))
            LINE -(nreg - nn * 300 + 1, y(5, 0))
            VIEW
        x(4) = x(5)
        y(4, 0) = y(5, 0)
    LOOP WHILE nreg < 600 + nn * 300
    WINDOW
   CLOSE #1
   CLOSE #2
  KEY(10) ON
   DO
   v$ = INKEY$
   LOOP WHILE v$ = ""
 LOOP WHILE (RIGHT$(v$,1)=  CHR$(75) OR RIGHT$(v$,1)= CHR$(77))
  ERASE cuadroa
  ERASE cuadrob
END SUB
```

```
´Subrutina "ayuda"
```

```
SUB ayuda
     CALL caratula(tip, tipana, tipconv, trans)
     LINE (2, 47)-(637, 478), 7, BF
     LINE (11, 67)-(629, 469), 0, BF
     LINE (12, 68)-(628, 468), , B
     LOCATE 4, 34
     PRINT " AYUDA "
     LINE (30, 445)-(137, 465), , BF
     LINE (28, 443)-(139, 467), , B
     LINE (500, 445)-(610, 465), , BF
     LINE (498, 443)-(612, 467), , B
     LOCATE 29, 65
     PRINT " F10=MENU ";
     LOCATE 29, 6
     PRINT " MAS=ENTER ";
DO
     LINE (13, 69)-(627, 442), 0, BF
     IF ayud% < 2 THEN
          ayud% = ayud% + 1
```

```
    ELSE
        ayud% = 1
    END IF

SELECT CASE ayud%
CASE 1:
LOCATE 6, 30
PRINT "TESIS DE GRADO"
LOCATE 8, 5
PRINT "      DESARROLLO DEL  SOFTWARE  PARA  LA SIMULACION  DE
FILTROS DIGITALES"
LOCATE 9, 5
PRINT "      RECURSIVOS  PARTIENDO DEL  DISEÑO DE  LOS FILTROS
ANALOGICOS DE"
LOCATE 10, 5
PRINT "                BUTTERWORTH, CHEVISHEV Y ELIPTICOS "
LOCATE 12, 5
PRINT "                Por:  CARLOS HERNAN SUAREZ LUNA"
LINE (12, 200)-(628, 202), , B
COLOR 7
LOCATE 15, 4
PRINT "  El sistema utiliza un  MENU  que aparece  al presionar
         la tecla <F10>"
LOCATE 16, 4
PRINT "ó cuando el programa  ha concluido una tarea.  Este menu
       tiene 9 "
LOCATE 17, 4
PRINT "posibilidad a elegir por medio de las <FLECHAS>.
       Cualquier función se "
LOCATE 18, 4
PRINT "se ejecuta con <ENTER>."
COLOR 15
LOCATE 20, 4
PRINT "TIPO: Función utilizada para seleccionar un filtro
       Pasa-Bajos, Pasa-Altos,"
LOCATE 21, 4
PRINT " Pasa-Banda o Elimina-Banda. Luego de haber
        seleccionado la"
LOCATE 22, 4
PRINT " banda, aparece una pantalla para introducir las
        frecuencias de corte, "
LOCATE 23, 4
PRINT " las atenuaciones y la frecuencia de muestreo. Con la
        tecla <ENTER>"
LOCATE 24, 4
PRINT " movemos el cursor y luego de situar en el punto de
        interés introducimos"
LOCATE 25, 4
PRINT " el dato, y volviendo a presionar <ENTER> aceptamos el
        valor."
COLOR 7
LOCATE 26, 4
PRINT "ANALOGICO: Para seleccionar el tipo de filtro analógico
        a partir del cual"
LOCATE 27, 4
PRINT " se realizará el diseño."
COLOR 15
```

```
CASE 2:
LOCATE 7, 4
PRINT "CONVERSION: Para seleccionar el tipo de conversión de
        filtros analogicos"
LOCATE 8, 4
PRINT " en digitales."
COLOR 7
LOCATE 9, 4
PRINT "TRANSFORMACION: Para seleccionar el método de
        transformación que se "
LOCATE 10, 4
PRINT " empleará, en el cambio de banda. Esta función esta
        activa solo en la"
LOCATE 11, 4
PRINT " transformación bilineal. Para las otras conversiones
        la transformación"
LOCATE 12, 4
PRINT " utilizada siempre es Digital-Digital. "
COLOR 15
LOCATE 13, 4
PRINT "SIMULACION: Esta opción efectúa la simulación del
        filtro, que consiste"
LOCATE 14, 4
PRINT " en calcular las funciones de transferencia analógica
        H(s) y digital H(z),"
LOCATE 15, 4
PRINT " y la señal filtrada y(n).  Además se calculará la
        función analógica "
LOCATE 16, 4
PRINT " transformada HT(s), solo cuando la transformación sea
        Analógica-Analógica."
LOCATE 17, 4
COLOR 7
PRINT "ARCHIVOS: Se utiliza para localizar los archivo de
        entrada y salida del"
LOCATE 18, 4
PRINT " sistema. También podemos archivar las respuestas de
        frecuencia, para"
LOCATE 19, 4
PRINT " efectuar comparaciones."
LOCATE 20, 4
COLOR 15
PRINT "GRAFICOS: Permite observar los resultados."
LOCATE 21, 4
PRINT " RESULTADO GLOBAL, se ven las respuestas de frecuencias
        analógica y digital"
LOCATE 22, 4
PRINT "   y la señal antes y después de filtrarla. En estas
          podemos elegir un"
LOCATE 23, 4
PRINT "   tramo de observación, utilizando las flechas."
LOCATE 24, 4
PRINT " COMPARACION, para comparar dos respuestas de
        frecuencia que pueden ser"
LOCATE 25, 4
PRINT "   especificadas en ARCHIVO 1 y ARCHIVO 2."
COLOR 7
```

```
LOCATE 26, 4
PRINT "AYUDA: Da al usuario una información ligera del uso del
        sistema."
COLOR 15
LOCATE 27, 4
PRINT "SALIR: Se usa para terminar la sesión de trabajo.  "

END SELECT
     DO
     v$ = INKEY$
     LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
LOOP
END SUB
```

```
'Subrutina "caratula"
```

```
SUB caratula (tip AS INTEGER, tipana AS INTEGER,
              tipconv AS INTEGER, trans AS INTEGER)
  SCREEN 12
  CLS

 LINE (0, 0)-(639, 479), , B
  PAINT (1, 1), 7, 15
 LINE (0, 45)-(639, 45)
 LOCATE 2, 10
  PRINT "           F I L T R O           D I G I T A L  "
 LINE (1, 441)-(638, 478), 0, BF
  FOR i = 1 TO 5
      LINE (0, 479)-(639, 480 - i * 4)
      SOUND 440 + 44 * i, 1
  NEXT
 LOCATE 29, 2
  PRINT "<FLECHAS=Seleccionar>";
 LOCATE 29, 46
  PRINT "<F2=Características>";
 LOCATE 29, 70
  PRINT "<F10=Menú>";
 LOCATE 29, 27
  PRINT "<ENTER=Aceptar>";

END SUB
```

```
'Subrutina "caratulainicial"
```

```
SUB caratulainicial
CLS
  FOR i = 1 TO 10
```

```
    LINE (390 - 14 * i, 190 - 9 * i)-
                                (390 + 14 * i, 190 + 9 * i), , B
    SOUND 440 + 44 * i, 1
 NEXT
 LINE (251, 101)-(519, 279), 0, BF
 LINE (300, 185)-(300, 120)
 LINE -(340, 120)
 LINE (300, 155)-(330, 155)
 LOCATE 12, 44
  PRINT " I L T R O"
 LINE (340, 210)-(340, 270)
 LINE -(380, 270)
 LINE -(380, 220)
 LINE -(370, 210)
 LINE -(340, 210)
 LOCATE 17, 51
  PRINT "I G I T A L"
 LINE (351, 280)-(450, 325), , B
 LINE (280, 325)-(515, 360), , B
 LOCATE 22, 38
  PRINT "Por:    CARLOS SUAREZ LUNA"
 LINE (20, 410)-(120, 435), , B
 LOCATE 27, 5
  PRINT "<F10=MENU>"

 RANDOMIZE TIMER
  FOR i = 1 TO 1200
     PSET (RND * 620 + 10, RND * 460 + 10), INT(RND * 3)
  NEXT
 DO
   FOR i = 1 TO 7
     LINE (280 - i + 1, 325 - i + 1)-
                          (515 + i - 1, 360 + i -     1), 0, B
     LINE (280 - i, 325 - i)-(515 + i, 360 + i), , B
     LINE (351, 280)-(351, 325 - i + 1), 0
     LINE (351, 280)-(351, 325 - i)
     LINE (450, 280)-(450, 325 - i + 1), 0
     LINE (450, 280)-(450, 325 - i)
     FOR j = 1 TO 1000
     NEXT
   NEXT
   FOR i = 7 TO 1 STEP -1
     LINE (280 - i, 325 - i)-(515 + i, 360 + i), 0, B
     LINE (280 - i + 1, 325 - i + 1)-
                                (515 + i - 1, 360 + i - 1), , B
     LINE (351, 280)-(351, 325 - i), 0
     LINE (351, 280)-(351, 325 - i + 1)
     LINE (450, 280)-(450, 325 - i), 0
     LINE (450, 280)-(450, 325 - i + 1)
     FOR j = 1 TO 1000
     NEXT
   NEXT
  LOOP
END SUB
```

```
´Subrutina "graficosmenu"


SUB graficosmenu (comp AS INTEGER)
compa% = comp
LINE (450, 140)-(190, 370), , BF
LINE (449, 141)-(191, 369), 0, B
LINE (219, 169)-(401, 296), 0, BF
LINE (240, 190)-(420, 320), 0, BF
LINE (400, 170)-(220, 200), , B
LINE -(400, 230), , B
LINE -(220, 263), , B
LINE -(400, 295), , B
LOCATE 12, 30
PRINT " RESULTADO GLOBAL "
LOCATE 14, 30
PRINT " ARCHIVO 1 "
LOCATE 16, 30
PRINT " ARCHIVO 2 "
LOCATE 18, 30
PRINT " COMPARAR "

 DO
 SELECT CASE compa%
 CASE 1:
      IF anteriorcompa% = 5 THEN
           LINE (221, 171)-(399, 199), 0, BF
           LOCATE 12, 30
           PRINT " RESULTADO GLOBAL "
        END IF

      IF anteriorcompa% = 2 THEN
           LINE (399, 231)-(221, 262), 0, BF
           LOCATE 16, 30
           PRINT " ARCHIVO 2 "
        END IF

      anteriorcompa% = 1
      LINE (220, 200)-(400, 230), , BF
      LOCATE 14, 30
      PRINT " ARCHIVO 1 "
      DO
      v$ = INKEY$
      LOOP WHILE v$ = ""

      IF RIGHT$(v$, 1) = CHR$(72) THEN
           compa% = 5
      END IF
      IF RIGHT$(v$, 1) = CHR$(80) THEN
           compa% = 2
      END IF

 CASE 2:
      IF anteriorcompa% = 1 THEN
           LINE (221, 201)-(399, 229), 0, BF
           LOCATE 14, 30
```

```
        PRINT " ARCHIVO 1 "
      END IF
    IF anteriorcompa% = 3 THEN
        LINE (221, 264)-(399, 294), 0, BF
        LOCATE 18, 30
        PRINT " COMPARAR "
      END IF

    anteriorcompa% = 2

    LINE (400, 230)-(220, 263), , BF
    LOCATE 16, 30
    PRINT " ARCHIVO 2 "
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        compa% = 1
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        compa% = 3
    END IF

CASE 3:
    IF anteriorcompa% = 5 THEN
        LINE (221, 171)-(399, 199), 0, BF
        LOCATE 12, 30
        PRINT " RESULTADO GLOBAL "
      END IF
    IF anteriorcompa% = 2 THEN
        LINE (399, 231)-(221, 262), 0, BF
        LOCATE 16, 30
        PRINT " ARCHIVO 2 "
      END IF

    anteriorcompa% = 3
    LINE (220, 263)-(400, 295), , BF
    LOCATE 18, 30
    PRINT " COMPARAR "
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        compa% = 2
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        compa% = 5
    END IF

CASE 5:
    IF anteriorcompa% = 3 THEN
        LINE (221, 264)-(399, 294), 0, BF
        LOCATE 18, 30
        PRINT " COMPARAR "
      END IF
```

```
    IF anteriorcompa% = 1 THEN
         LINE (221, 201)-(399, 229), 0, BF
         LOCATE 14, 30
         PRINT " ARCHIVO 1 "
      END IF
    anteriorcompa% = 5
    LINE (220, 170)-(400, 200), , BF
    LOCATE 12, 30
    PRINT " RESULTADO GLOBAL "
    DO
    v$ = INKEY$
    LOOP WHILE v$ = ""

    IF RIGHT$(v$, 1) = CHR$(72) THEN
        compa% = 3
    END IF
    IF RIGHT$(v$, 1) = CHR$(80) THEN
        compa% = 1
    END IF
 END SELECT
 LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
  comp = compa%
END SUB



'Subrutina "inpu"


SUB inpu (longitudt%, numero, palabra$, nol%, fila%, columna%)
  IF nol% = 0 THEN
     palabra$ = SPACE$(longitudt% + 1)
  ELSE
     longitud = LEN(palabra$)
     FOR i = longitud + 1 TO longitudt%
     palabra$ = palabra$ + " "
     NEXT
  END IF
  posicion = 1
  REDIM letra(100)

DO
   LOCATE fila%, columna% + posicion - 1
   PRINT MID$(palabra$, posicion, 1)
   GET ((columna% + posicion - 2) * 8, (fila% - 1) * 16) -
    ((columna% + posicion - 1) * 8 - 1, fila% * 16 - 1), letra
   PUT ((columna% + posicion - 2) * 8, (fila% - 1) * 16),
                                                    letra, PRESET
   mletra = 0
   DO
   v$ = INKEY$
   LOOP WHILE v$ = ""
  IF nol% = 0 THEN
   IF RIGHT$(v$, 1) <> CHR$(48) AND
                                RIGHT$(v$, 1) <> CHR$(53) THEN
   IF RIGHT$(v$, 1) <> CHR$(49) AND
```

```
                                  RIGHT$(v$, 1) <> CHR$(54) THEN
  IF RIGHT$(v$, 1) <> CHR$(50) AND
                                  RIGHT$(v$, 1) <> CHR$(55) THEN
  IF RIGHT$(v$, 1) <> CHR$(51) AND
                                  RIGHT$(v$, 1) <> CHR$(56) THEN
  IF RIGHT$(v$, 1) <> CHR$(52) AND RIGHT$(v$, 1) <> CHR$(57)
                              AND RIGHT$(v$, 1) <> CHR$(46) THEN
    mletra = 1
  END IF
  END IF
  END IF
  END IF
  END IF
END IF
IF RIGHT$(v$, 1) <> CHR$(77) AND RIGHT$(v$, 1) <>  CHR$(75)
           AND RIGHT$(v$, 1) <> CHR$(72) AND
               RIGHT$(v$, 1) <> CHR$(80) THEN
  IF RIGHT$(v$, 1) <> CHR$(13) AND RIGHT$(v$, 1) <> CHR$(83)
           AND RIGHT$(v$, 1) <> CHR$(8) THEN
   IF nol% <> 0 OR mletra <> 1 THEN
   MID$(palabra$, posicion, 1) = RIGHT$(v$, 1)
    IF longitudt% > longitud THEN
      longitud = longitud + 1
    END IF
    IF longitudt% > posicion THEN
      posicion = posicion + 1
    END IF
   END IF
  END IF
ELSEIF (RIGHT$(v$, 1) = CHR$(77)) AND
                                     (longitud > posicion) THEN
  posicion = posicion + 1
ELSEIF (RIGHT$(v$, 1) = CHR$(75)) AND (1 < posicion) THEN
  posicion = posicion - 1
END IF
IF RIGHT$(v$, 1) = CHR$(8) AND (posicion > 1) THEN
  FOR i = 1 TO (longitud - posicion + 1)
  MID$(palabra$, posicion +  i - 2, 1) = MID$(palabra$,
                                       posicion + i - 1, 1)
  NEXT
  MID$(palabra$, longitud, 1) = " "
  posicion = posicion - 1
END IF
IF RIGHT$(v$, 1) = CHR$(83) THEN
  FOR i = 1 TO (longitud - posicion)
  MID$(palabra$, posicion + i - 1, 1) = MID$(palabra$,
                                           posicion + i, 1)
  NEXT
  MID$(palabra$, longitud, 1) = " "
END IF

  IF RIGHT$(v$, 1) <> CHR$(13) THEN
    LOCATE fila%, columna%
    PRINT palabra$
  END IF
LOOP WHILE RIGHT$(v$, 1) <> CHR$(13)
 IF nol% = 0 THEN
```

```
    punto = INSTR(palabra$, ".")
    final = INSTR(palabra$, " ")
    IF punto <> 0 THEN
      entera$ = LEFT$(palabra$, punto - 1)
      decimal$ = RIGHT$(palabra$, longitudt% + 1 - punto)
      numero = VAL(entera$) + VAL(decimal$) / 10^(final-1-
                                                        punto)
    ELSE
      numero = VAL(palabra$)
    END IF
  ELSE
      final = INSTR(palabra$, " ")
      FOR i = final TO longitudt%
      MID$(palabra$, i, 1) = " "
      NEXT
      palabra$ = RTRIM$(palabra$)
  END IF
  ERASE letra
END SUB



'Subrutina "instantanea"



SUB instantaneo (tip AS INTEGER, tipana AS INTEGER,
                 tipconv AS INTEGER, trans AS INTEGER)

 REDIM instante(8000)
 GET (400, 10)-(630, 150), instante
 LINE (400, 10)-(630, 150), , BF
 LINE (402, 12)-(628, 148), 0, B
 LINE (410, 45)-(598, 115), 0, BF
 LINE (412, 47)-(596, 113), , B
 LINE (432, 114)-(616, 123), 0, BF
 LINE (598, 60)-(616, 123), 0, BF

 LOCATE 2, 52
 PRINT "Características del Filtro:"
 LOCATE 4, 53
     SELECT CASE tip
          CASE 1
          PRINT "PASA-BAJOS  "
          CASE 2
          PRINT "PASA-ALTOS  "
          CASE 3
          PRINT "ELIMINA-BANDA"
          CASE 4
          PRINT "PASA-BANDA  "
     END SELECT
 LOCATE 5, 53
     SELECT CASE tipana
          CASE 1
          PRINT "BUTTERWORTH    "
          CASE 2
```

```
            PRINT "CHEVISHEV TIPO I "
            CASE 3
            PRINT "CHEVISHEV TIPO II"
            CASE 4
            PRINT "ELIPTICO      "
     END SELECT
 LOCATE 6, 53
 SELECT CASE tipconv
            CASE 1
            PRINT "INVARIANZA DE IMPULSO"
            CASE 2
            PRINT "TRANSFORMADA BILINEAL"
            CASE 3
            PRINT "INVARIANZA DE PULSO "
     END SELECT
 LOCATE 7, 53
 SELECT CASE trans
            CASE 1
            PRINT "ANALOGI-ANALOGI"
            CASE 2
            PRINT "DIGITAL-DIGITAL"
     END SELECT

 LINE (510, 125)-(545, 147), 0, B
 LINE (512, 127)-(543, 145), 0, BF
 LOCATE 9, 65
 PRINT "<OK>"
      DO
      v$ = INKEY$
      LOOP WHILE v$ <> CHR$(13)

 PUT (400, 10), instante, PSET
 ERASE instante

END SUB



'Subrutina "porcentaje"



SUB porcentaje
     LINE (310, 160)-(550, 320), , BF
     LINE (311, 161)-(549, 319), 0, B
     LINE (320, 170)-(520, 290), 0, BF
     LINE (322, 172)-(518, 288), , B
     LINE (520, 190)-(535, 305), 0, BF
     LINE (340, 290)-(535, 305), 0, BF
     LINE (382, 265)-(462, 255), , B
     LOCATE 17, 46
     PRINT "0%"
     LOCATE 17, 60
     PRINT "100%"
END SUB
```

GLOSARIO

ABSCISA: Recta horizontal del plano cartesiano que pasa por el punto origen del sistema.

ADIMENSIONAL: Expresión que no tiene dimensión. Surge de la relación de dos magnitudes iguales.

ALABEO: Deformación que experimenta la variable frecuencia al utilizar la Transformada Bilineal.

ALIASING: Denominación que tiene el cruce de espectros.

ATENUACION: Acción de disminuir una magnitud.

CATENARIA: Función matemática que representa la curva de equilibrio de un hilo flexible suspendido por sus extremos bajo la única acción de la gravedad.

CEROS: Valor de la variable independiente de una función, que hace cero a esta.

CONVERGENCIA: Propiedad que tienen las funciones cuando su valor es finito.

CONVOLUCION: Operación matemática efectuada entre dos funciones y que es utilizada en el análisis de los sistemas lineales.

DECIBELIOS: Décima parte de un bel, unidad de medida para expresar la relación de dos magnitudes iguales.

EQUIRRIZADO: Variaciones de igual amplitud de una señal, alrededor del valor verdadero de esta.

ESPECTRO: Función que representa la amplitud de las componentes de frecuencia de una señal o sistema.

HIPERBOLICO: Función que tiene la forma de una hipérbola.

IDENTIDAD TRIGONOMETRICA: Igualdad matemática, determinada al analizar las propiedades del triángulo rectángulo.

INVARIANCIA: Que no tiene variación. Término utilizado para especificar que las respuestas de frecuencia de las señales impulso y pulso no han variado.

LAPLACE: Transformada de Laplace. Operación matemática que transforma señales en el dominio del tiempo al dominio de la frecuencia compleja.

MONOTONA: Que se produce con cierta igualdad y suavidad.

ORDENADA: Recta vertical del plano cartesiano que pasa por el origen del sistema.

PARIDAD: Condición de los números enteros múltiplos de dos.

PERIODICIDAD: Propiedad que tienen las funciones, al repetirse el valor de su amplitud a intervalos fijos de la variable independiente.

PIXEL: Punto definido en la pantalla de un monitor.

POLOS: Valor de la variable independiente de una función, que hace cero al denominador de esta, ocasionando una indeterminación.

RADIAN: Unidad de medida de ángulos.

RECURRENCIA: Obtención de ciertos valores a partir de los anteriores.

SIMULACION: Acción de representar un sistema real, para efecto de análisis o tratamiento.

SUBRUTINA: Estructura de programación que encierra un conjunto de instrucciones.

SUPRESION: Banda de Supresión. Rango de la respuesta de frecuencia de un filtro donde se suprimen las componentes de frecuencia.

BIBLIOGRAFIA

- A.ANTONION, Digital Filters: Analysis and Design , McGraw-Hill Book Company, New York, 1979.

- FLOYD M. GARDNER, "A Transformation for Digital Simulation of Analog Filters", IEEE Transaction on Comunications , Vol COM-34, NO.7, July 1986.

- STEVEN NAMEROFF, QuickBASIC:Manual de referencia , McGraw-Hill Interamericana de España, 1989.

- A.V.OPPENHEIM AND R.W.SCHAFER, Digital Signal Processing Prentice-Hall, 1975.

- L.WEINBERG, Network Analysis and Synthesis , McGraw-Hill New York, 1962.