

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

ANALISIS COMPARATIVO DEL FILTRO TDL Y DEL FILTRO DE
MALLA PARA ECUALIZACION ADAPTIVA

POR OMAR FLORES VILLANIL.

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN
ELECTRONICA Y TELECOMUNICACIONES POR LA ESCUELA
POLITECNICA NACIONAL

QUITO, DICIEMBRE 28 DE 1992.

Certifico que el presente
trabajo ha sido realizado
en su totalidad por el Sr.
Carlos Omar Flores Villamil.

A handwritten signature in cursive script, reading "Erwin Barriga", is written over a horizontal dashed line.

Ing. Erwin Barriga.

Director de tesis.

DEDICATORIA

A mis Padres,

A mis Hermanos,

A Patricia.

AGRADECIMIENTO.

Mi sincero agradecimiento al Ingeniero Erwin Barriga, por su acertada dirección. A todas las personas que de una u otra forma contribuyeron a la culminación de esta tesis.

INDICE

CAPITULO I: INTRODUCCION

1.1 Generalidades.	1
1.2 Procesos Adaptivos.	3
1.3 Relación entre los métodos AR de Predicción Lineal y la Ecuación Adaptiva.	3
1.4 Obtención de la Función de Transferencia del Ecuación.	5
1.5 Tipos de ecualizadores.	7

CAPITULO II: ANALISIS DEL SISTEMA DE TRANSMISION A USAR.

2.1 Descripción del canal a usar y del tipo de transmisión a analizar.	11
2.2 Los ecualizadores como filtros de forzamiento de zero.	15
2.3 Criterios de Nyquist de Interferencia Intersímbolo.	18
2.4 Determinación de la Función de Transferencia de los Filtros de Transmisión y de Recepción.	30
2.5 Recomendaciones del CCITT para limitar la distorsión de amplitud y de fase del canal de transmisión.	37
2.6 Modelo de banda base para transmisión BPSK y explicación de las limitaciones que deben observarse en lo que respecta a amplitud y frecuencia en el caso de generalizar el modelo para una señal QPSK	40

CAPITULO III: EL FILTRO TDL (LINEA DE RETARDOS CON DERIVACION).

3.1 Estructura del filtro TDL.	49
3.2 El Método de los Mínimos Cuadrados de Predicción Lineal y la Determinación de las Ganancias Optimas de los taps del filtro TDL.	51
3.3 Análisis del algoritmo recursivo para la determinación	

de las Ganancias de los taps.	53
3.4 Algoritmo para el Filtro TDL.	57

CAPITULO IV: ANÁLISIS DEL AUMENTO DE LA VELOCIDAD DE CONVERGENCIA.

4.1 Error cuadrático de exceso debido al uso de estimados ruidosos del gradiente.	59
4.2 Propiedades de convergencia del algoritmo recursivo TDL.	61
4.3 La Ecuación Homogénea Desacoplada que controla la Convergencia.	64
4.4 Condiciones que debe cumplir el coeficiente de convergencia μ	65
4.5 Aumento de la velocidad de convergencia del algoritmo mediante la ortogonalización de las señales de entrada al ecualizador.	67
4.6 Determinación del valor óptimo de μ	70
4.7 Deducción de la expresión para el error residual ϵ_r	71
4.8 Análisis del Ruido Gaussiano.	75

CAPITULO V: EL FILTRO DE MALLA O LATTICE

5.1 Propagación de señales en el filtro.	81
5.2 Propiedades de ortogonalización del Filtro de Malla.	81
5.3 Cálculo de los coeficientes de reflexión.	83
5.4 El proceso de ortogonalización de Gram-Schmidt.	85

CAPITULO VI: LA PREDICCIÓN LINEAL Y LAS ECUACIONES DE YULE-WALKER.

6.1 Las Ecuaciones de Yule-Walker.	93
6.2 El vector D ó coeficiente de Kalman y su escalar asociado, gama.	99
6.3 Ecuaciones para actualización del orden de A, C, del	

error directo y reverso, de la energía y de los coeficiente de reflexión.	101
6.4 Ecuaciones para actualización del orden de D y gama.	105
6.5 Ecuaciones para actualización del tiempo.	106
6.6 El filtro Lattice para procesos conjuntos, específico para la ecualización	111
6.7 Algoritmo del Filtro de Malla.	113
6.8 Diagrama de Flujo General del Programa.	116

**CAPITULO VII: RESULTADOS DE LA SIMULACION PARA
EL CANAL DE TRANSMISION.**

7.1 Simulación de la transmisión a través de un canal con distorsión.	121
7.2 Espectro de frecuencia de la señal NRZ transmitida.	122
7.3 Bits recibidos para diferentes velocidades de transmisión.	125
7.4 Explicación de la distorsión observada usando las curvas de distorsión del CCITT.	126

**CAPITULO VIII: RESULTADOS DE LA SIMULACION PARA TRANSMISION
CON ECUALIZACION Y ANALISIS COMPARATIVO.**

8.1 Resultados de la Ecualización con el Filtro TDL.	130
8.2 Convergencia del Error Cuadrático y de los Coeficientes del Filtro TDL.	132
8.3 Respuesta Impulsiva del Canal	134
8.4 Resultados obtenidos al variar el coeficiente de convergencia μ	136
8.5 Incidencia del número de coeficientes en el Error Cuadrático.	137
8.6 Resultados de la Ecualización con el Filtro de Malla.	138
8.7 Convergencia del Error Cuadrático y de los Coeficientes del Filtro de Malla.	140
8.8 Incidencia del factor de olvido w sobre la	

Convergencia del Error Cuadrático.	142
8.9 Resultados obtenidos en el período de autoajuste.	144
8.10 Límite inferior para el factor de olvido w	145
8.11 Análisis comparativo de los Filtros TDL y de Malla	148
8.12 Conclusiones Generales.	150
APENDICE I : EL LENGUAJE TURBO C++ Y EL MENU DE INGRESO.	A-1
APENDICE II: CODIGO FUENTE DEL PROGRAMA EN LENGUAJE C++.	A-6
APENDICE III: TRANSFORMADA DE FOURIER, SISTEMAS ESTOCÁSTICOS Y RUIDO.	A-28
APENDICE IV: ALGEBRA MATRICIAL.	A-49
APENDICE V: MANUAL DEL USUARIO.	A-82

BIBLIOGRAFIA: La Bibliografía se incluirá en forma de pie de página en donde fuere necesario, para mayor facilidad de uso.

CAPITULO I -

CAPITULO I

1.1 Generalidades.

El estudio de las distorsiones que sufre la transmisión de la información se remonta a los comienzos de la comunicación electromagnética por cables. Ya en 1928, Nyquist, en su documento fundamental¹ estableció la importancia de contrarrestar los efectos de distorsión debidos a la Interferencia Intersímbolo (ISI) y al ruido mediante la transmisión de pulsos que posean simetría en el dominio de la frecuencia, lo que en el dominio del tiempo se traduce entre otras posibles señales en los pulsos coseno elevado (raised cosine). Debido a la importancia que revisten este tipo de pulsos para la posterior simulación, en el Capítulo II detallaremos la afirmación anterior concerniente a la simetría para obtener la respuesta de coseno elevado².

La adecuada modelación del pulso de la señal a transmitir determina que en los tiempos de muestreo la señal valga cero. En caso contrario el corrimiento resultante se sobrepone aditivamente con los pulsos adyacentes dando lugar a interferencia, que puede ser causa de error en la recepción.

Este documento de Nyquist condujo a nuevos análisis para la corrección de las características de fase. La bibliografía consultada³ coincide en señalar que los primeros reportes para obtener un ecualizador de fase que se adapte recursivamente a las variaciones del canal se remontan a 1965 con el trabajo de Lucky

¹ Nyquist, Harry. Certain Topics in Telegraph Transmission Theory. Feb. 1928. Reproducido en L.E. Franks. Baseband Communications. IEEE Press. 1971

² Feher K, Digital Communications. 1983 Prentice Hall.

³ Franks, L.E. Baseband Communications. IEEE Press. 1971.

en los Bell Laboratories. En este documento, Lucky ⁴ usa una secuencia enviada durante el período de entrenamiento para ajustar las ganancias de los taps de un Filtro Transversal o Filtro "Línea de Retardos con Derivaciones" (TDL de sus iniciales en inglés de Tapped Delay Line) mediante un criterio de minimización de error conocido como forzamiento adaptivo de ceros, el cual lo analizaremos en el Capítulo II, literal 2.2, en su forma no adaptiva.

En un documento posterior, de Febrero de 1966, Lucky propone ciertas variaciones al ecualizador original de modo que si durante la etapa en la cual el canal está en servicio, existen variaciones de fase (por envejecimiento, etc), no sea necesario interrumpir la transmisión de información para actualizar las ganancias de los taps (ver figura 1). De esta forma se ve que la ecualización como se la concibe en la actualidad consiste de dos etapas:

- La primera etapa, llamada de entrenamiento, en la cual se envía una secuencia de pulsos conocida al canal, en base a los cuales se actualiza los parámetros del filtro. Por esta razón se le conoce a este método como ecualización adaptiva.

- La segunda etapa es la etapa de autoajuste.

Cuando el canal se encuentra en servicio, el tratar de enviar una secuencia de entrenamiento sería oneroso, a la vez que interrumpiría el flujo de información. Es por esto que el filtro debe disponer de realimentación, la cual actualiza continuamente los coeficientes.

Por lo dicho, la ecualización es un proceso dinámico de ajuste de las características del filtro en función de las

⁴ Lucky, R.W. Automatic Equalization. The Bell System Technical Journal, April 1965. Reproducido en L.E. Franks, obra citada.

variaciones del canal.

Este ajuste lo llevaremos a cabo mediante la implementación de métodos recursivos, los que calculan el valor siguiente de los coeficientes en función del conocimiento de los valores pasados de los mismos y se basan para el efecto en la minimización del error entre la señal recibida y la señal de entrenamiento almacenada (o del valor más probable en el caso de la etapa de autoajuste, cuya obtención es asimismo tema de la tesis).

1.2 Procesos adaptivos.

Los procedimientos adaptivos son conocidos desde hace mucho tiempo atrás. Se los ha usado exitosamente para la Cancelación Adaptiva de Ruido (Jiménez, M.S. Tesis. EPN. 1988), para la Determinación de Sinusoides en un Fondo de Ruido (Hidalgo, P. Métodos de Prony y Pisarenko para Estimación Espectral, Tesis. EPN. 1985), para la detección de fonemas, etc. La esencia de estos métodos radica en tratar de modelar el sistema mediante métodos de regresión lineal conocidos también como métodos regresivos (AR:Autoregression, MA:Moving Average, ARMA:AR + MA).

Para el caso específico de la ecualización se usa el método AR ya que este modelo ofrece control sobre las características de fase y además es inherentemente estable.⁵

1.3 Relación entre los métodos AR de Predicción Lineal y la Ecualización Adaptiva.

El modelo común para la Predicción Lineal y la Ecualización Adaptiva se muestra en la fig. 1.

⁵ Grant and Cowan. Adaptive Filters. 1985.

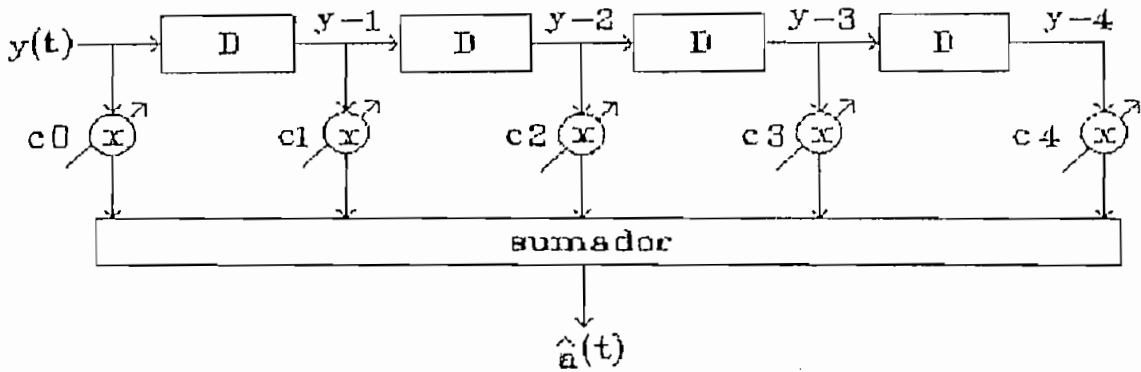


Figura 1 Modelo común de AR y Ecuación Adaptiva

La señal observada $y(n)$, puede ser modelada⁶ como una secuencia $s(n)$, la cual es corrompida por ruido aditivo blanco, $\Omega(n)$, con varianza σ^2 :

$$y(n) = s(n) + \Omega(n) \quad (1.1)$$

En la recepción, el estimado de $s(n)$, $\bar{a}(n)$, es definido en términos de un polinomio lineal finito en $y(n)$ -lo cual explicaremos en mayor detalle en el capítulo II, numeral 2.1:

$$\bar{a}(n) = \sum_{i=0}^{2N} c_i y(n-i) \quad (1.2)$$

donde los c_i son pesos individuales en un filtro como el de la figura 1. Para simplificar las expresiones y aclarar la manipulación matemática, la ecuación (1.2) puede ser reescrita en notación matricial-vectorial como

$$\bar{a}(n) = C^T Y(n) \quad (1.3)$$

⁶ Cowan. Optimum Estimation Techniques. Capítulo 2 en la obra citada de Grant and Cowan.

donde

$$C^T = [c_0, c_1, \dots, c_{2N}]$$

$$Y^T(n) = [y(n), y(n-1), \dots, y(n-2N)]$$

lo que puede observarse de la Fig.1 con $2N=4$. El superíndice T denota matriz transpuesta.

La diferencia entre los métodos de ecualización y los métodos AR de, digamos, detección de señal en un fondo ruidoso, radica en la forma de la señal $y(n)$. En los métodos AR, $y(n)$ toma la forma de la ecuación (1.1), donde $s(n)$ es la señal original sin distorsión. En tanto que en la ecualización,

$$y(n) = \sum_i a_i r(n-i) + \Omega(n) \quad (1.4)$$

donde $r(n)$ se obtiene de $a(n)$ según la ecuación (2.5) del capítulo II, y es la señal afectada por la función impulsiva del canal $h(n)$, la cual ha causado ISI, y $a(n)$ corresponde como en los métodos AR a la señal original sin distorsión.

1.4 Obtención de la Función de Transferencia del Ecualizador.

En el caso específico de la ecualización, la modelación del sistema conduce a obtener una función de transferencia (FT) del filtro adaptivo:

$$C(s) = 1 / H(s) \quad (1.5)$$

donde $H(s)$ es la FT del canal, que es el que introduce la ISI. El proceso de obtener la función inversa del canal es el siguiente, ver la fig. 2. Se envía el símbolo de información deseado $a(t)$ a través del canal $h(t)$ y se obtiene en la recepción la señal $y(t)$ en la entrada del mismo nombre ($y(t)$ del filtro adaptivo C), la

cual luego de pasar a través del filtro da $\hat{a}(t)$, que es el estimado de $y(t)$, que en el caso ideal será $\hat{a}(t) = a(t)$. Por otra parte, en la Recepción se debe disponer de la señal $a(t)$, que durante el período de entrenamiento debe estar almacenada, ó en el período de autoajuste debe ser obtenida mediante cuantización (a partir de $\hat{a}(t)$).

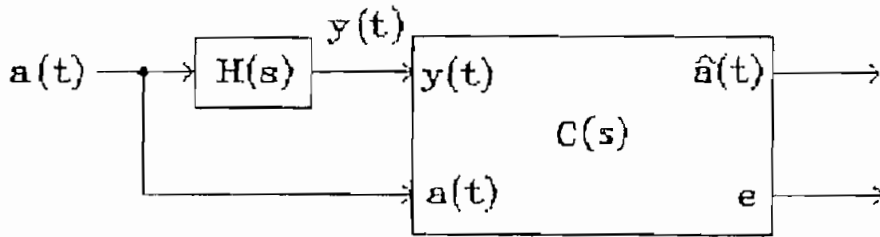


Figura 2. Inversión de la función del canal $H(s)$ usando un filtro adaptivo $C(s)$

En cualquier caso, se procede a formar la señal de error:

$$e = a(t) - y(t) \quad (1.6)$$

que en el límite cuando $e \rightarrow 0$ por acción del ecualizador $C(s)$, $y(t) \rightarrow a(t)$, consiguiendo entonces el mejor estimado de $a(t)$: $\hat{a} \rightarrow a$, lo que en el dominio de la frecuencia da:

$$\bar{A}(s) = C(s) Y(s) = C(s) H(s) A(s) \quad (1.7)$$

y en el límite, cuando $\bar{A} \rightarrow A$,

$$1 = C(s) H(s) \quad (1.8)$$

$$C(s) = 1 / H(s)$$

de lo que se puede concluir que el filtro adaptivo invierte la función de transferencia del canal para de esta manera cancelar

los cambios introducidos, entre ellos la ISI.

1.5 Tipos de ecualizadores.

La ecualización se usa, entre otras aplicaciones, en sistemas que usan modems para la transmisión de datos computarizados. La velocidad alcanzada en la década de los 70 era de alrededor de 1200 bits/s para transmisión BPSK, límite que ha podido ser superado gracias a las técnicas de ecualización.

El primer ecualizador usado exitosamente en 1965 por Lucky fue el ecualizador ya mencionado de forzamiento de cero (ZFE), el cual es todavía usado para canales con alta relación señal a ruido.

Sin embargo, el aparecimiento a inicios de los 70 del algoritmo del gradiente de descenso más pronunciado por parte de Widrow⁷ y otros, determinó que se use esta variedad de ecualizador durante la siguiente década. La razón de la preeminencia del LMS (least mean square) sobre el filtro de forzamiento de cero es debida a que la convergencia de este método no depende de los valores de las muestras, como en el ZFE, sino de un factor de convergencia que asegura convergencia en todos los casos.

Posteriormente hallamos dos tipos de ecualizadores adicionales: el ecualizador de espaciamento fraccional entre taps (FTE) y el ecualizador de realimentación en la decisión (DFE). El primero es una generalización del TDL de Widrow, y se demuestra que es mejor que el TDL, debido a que no precisa de un filtro acoplado a la entrada, ya que el espaciamento fraccionado

⁷ Widrow B, et al, Stationary and nonstationary learning characteristics of the LMS adaptive filter. Proceedings of the IEEE. Aug. 1976.

en el tiempo, ($\tau = T/M$) fig. 3., puede compensar por el uso de dicho filtro.⁸ Con el adiconamiento del filtro acoplado, la libertad en el espaciamiento de los taps puede ser usada para compensar una distorsión más severa que la que soporta el TDL.

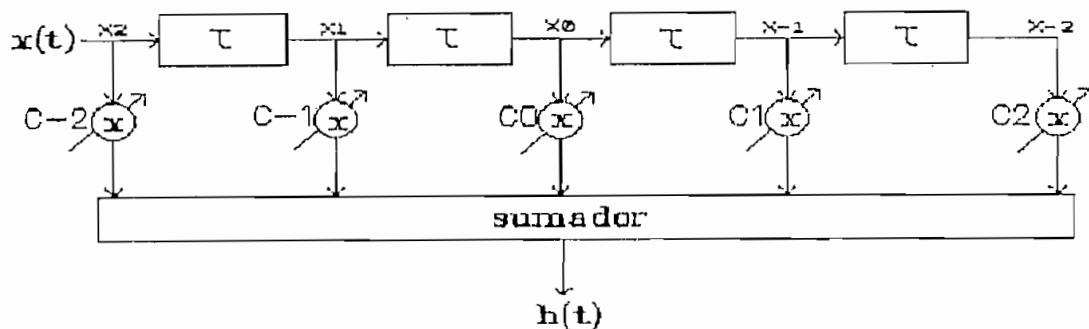


Figura 3 Ecuador de espaciamento fraccional

Por su parte, el ecualizador de realimentación en la decisión está constituido por dos TDL's, fig.4. Uno hacia adelante que va de la entrada hacia un sumador y otro hacia atrás, que va desde la salida del elemento de decisión a la entrada del sumador con signo menos. Como se demuestra en Monsen⁹, el TDL hacia adelante ecualiza los pulsos adelantados (para $t-i$), mientras que el TDL hacia atrás cancela los pulsos de la cola (para $t+i$). Se demuestra también que el funcionamiento es superior al TDL normal puesto que como el TDL hacia adelante está realizando la mitad del trabajo que en el TDL normal, el "resto" de su capacidad lo usa en la minimización del ruido, siendo por esta razón más inmune al ruido. La desventaja de este tipo de ecualizador es que en el caso de que exista un error, este error es realimentado, lo que da lugar a más errores ocurriendo un error explosivo (burst). Sin embargo, este error es controlado en

⁸ Qureshi, S. Adaptive Equalization. Proceedings of the IEEE. Sept. 1985.

⁹ Monsen, Feedback Equalization for Fading Dispersive Channels. IEEE Trans. on Information Theory. Jan. 1971.

un tiempo tal que disminuye sólo ligeramente el comportamiento general del filtro.

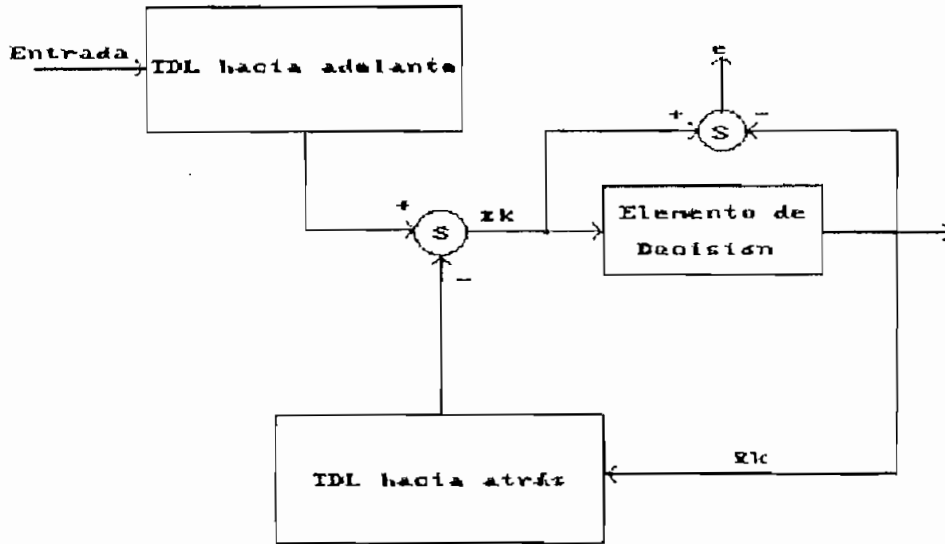


Figura 4 Ecuador con realimentación en la decisión.

Para mayor información sobre el filtro de espacionamiento fraccional y el de realimentación en la decisión, tenemos la tesis de Núñez¹⁰.

Nuestro trabajo presenta además del filtro TDL, el filtro de Malla, el cual es superior en lo que se refiere a la velocidad de convergencia.

Como se demostrará, esta mayor velocidad de convergencia radica en la ortogonalización de la matriz de covarianza.

De esta forma, en la búsqueda del ecualizador óptimo, se ha encontrado que el algoritmo óptimo es no lineal, dentro de los

¹⁰ Núñez, Tesis: Ecuación Digital. EPN 1986.

cuales se encuadra el ya mencionado ecualizador de espaciamiento fraccional.

La búsqueda del mejor ecualizador no lineal ligado al criterio de optimización de la probabilidad de error desembocó en el desarrollo del estimador de máxima probabilidad usando el algoritmo de Viterbi, y de una versión subóptima denominada ecualizador de realimentación en la decisión que ya lo mencionamos anteriormente.

Todos los ecualizadores mencionados han permitido de una u otra forma superar la señalización a 1200 b/s, lográndose tasas de 2400 b/s con la modulación BPSK, de 4800 b/s con la modulación QPSK y de hasta 9600 b/s con QAM (modulación simultánea de amplitud y de fase).

La investigación actual está encaminada a lograr mayores velocidades mediante la modulación que contempla la codificación con el código de Trellis, con lo que se pueden alcanzar velocidades de hasta 16800 b/s.

CAPITULO II.

CAPITULO II

2.1 Descripción del canal a usar y del tipo de transmisión a analizar.

La estructura básica a usar es un receptor adaptivo que es capaz de señalización digital a alta velocidad en un canal de banda limitada que puede variar lentamente.

Debe anotarse que la respuesta impulsiva del canal es desconocida para el receptor.

El elemento esencial del receptor adaptivo es un filtro: en nuestro caso usaremos el Filtro TDL y el Filtro de Malla. Los taps ó ganancias de los filtros (ci en la fig. 1), son ajustados mediante el método del descenso más pronunciado para minimizar el error cuadrático medio (MSE) debido tanto a la interferencia intersímbolo, cuanto al ruido aditivo.

Nuestro tratamiento será aplicado al procesamiento de datos digitales transmitidos por el método BPSK (Binary Phase Shift Keying), y oportunamente estableceremos las restricciones que deben observarse al generalizar los resultados a la modulación QPSK (Quadrature Phase Shift Keying).

Esta sección tratará con un modelo matemático del sistema de comunicación en el cual se usa el receptor adaptivo.

El modelo matemático usado para representar el sistema de comunicación digital se presenta en la Fig. 5.

La secuencia de información $\{a_n\}$ modula la forma de onda básica $s(t)$, que en el caso ideal corresponde a la función impulsiva delta de Dirac, y en el caso práctico a pulsos digitales de duración finita. La modulación se produce a una tasa

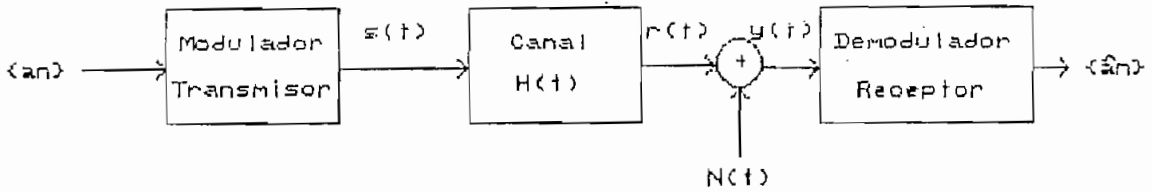


Figura 5 Modelo de Comunicación Digital.

de $1/T$.

Basándonos en la función $s(t)$ podemos establecer la señal total transmitida como:

$$s_t(t) = \dots + a_{-1}s(t+T) + a_0s(t) + a_1s(t-T) + \dots \quad (2.1)$$

que en notación abreviada es:

$$s_t(t) = \sum_{i=-N}^N a_i s(t-iT) \quad (2.2)$$

donde $s_t(t)$ = secuencia transmitida

a_i = secuencia moduladora

iT = tiempos de muestreo.

Para el caso concreto de la simulación usaremos los valores $+1$, -1 , como secuencia moduladora, los cuales corresponden al código NRZ. Recordemos, sin embargo, que modular la amplitud de una portadora sinusoidal con estos valores, es equivalente a proporcionar a la misma un cambio de fase de 180 grados, lo que constituye la modulación BPSK que es la que usaremos en la simulación.

El canal a través del cual es transmitida la información se caracteriza por su respuesta impulsiva $h(t)$, la cual asumimos desconocida en la recepción.

Sabemos de los conocimientos básicos de sistemas lineales, que la respuesta del canal a la función $s_t(t)$ es la convolución de $h(t)$ con $s_t(t)$:

$$y(t) = h(t) * s_t(t) \quad (2.3)$$

$$y(t) = \sum_{i=-N}^N a_i r(t-iT) \quad (2.4)$$

donde la convolución está definida por:

$$r(t) = \sum_{k=-NT}^{NT} s_t(k) h(t-k) \quad (2.5)$$

con : $h(t)$ = Respuesta Impulsiva del Canal

a partir de la cual se puede obtener usando la Transformada de Fourier: $H(s)$ = Función de Transferencia del Canal.

Por otra parte, de la figura 5 se ve que el ruido se incorpora aditivamente a la señal. En (2.6) hemos tomado en cuenta el ruido aditivo del canal, que asumiremos gaussiano.

$$y(t) = \sum_{i=-N}^N a_i r(t-iT) + n(t) \quad (2.6)$$

Si enviamos secuencias $s_t(t)$ a través del canal, en la recepción tendremos una señal $y(t)$ que refleja las distorsiones introducidas por el canal en la señal.

Supongamos que el símbolo deseado es a_0 , el cual es la salida del filtro transversal mostrado en la fig.1. Con c_i ($i=-N, \dots, 0, \dots, N$), como coeficientes de los taps se obtiene el estimado del símbolo deseado \bar{a}_0 :

$$\bar{a}_0 = \sum_{i=-N}^N c_i y(t-iT) \quad (2.7)$$

donde \bar{a}_0 = valor estimado

Para el instante $t_0 = t(0) = 0$,

$$\bar{a}_0 = \sum_{i=-N}^N c_i y(-iT) \quad (2.8)$$

donde para simplificar la expresión definamos,

$$y_{-i} = y(-iT) \quad (2.9)$$

El voltaje en (2.8) es la suma del término deseado a_0 más un término de interferencia intersímbolo:

$$\bar{a}_0 = y_0 c_0 + \underbrace{\sum_{i=-N}^N c_i y(-i)}_{\text{ISI}} \quad (2.10)$$

donde el apóstrofe en el sumatorio indica que se ha omitido el término $i = 0$, y

$$a_0 = y_0 c_0 = \text{símbolo deseado } (c_0 = 1)$$

\bar{a}_0 = símbolo estimado

OBSERVACION 1: de lo anterior se concluye que para que la interferencia intersímbolo al tiempo 0 sea nula, debe cumplirse que

$$\bar{a}(0) = \begin{cases} 1 & t=0 \\ 0 & \text{en otro caso} \end{cases}$$

De aquí surge el primer criterio de ecualización, a saber, que si forzamos ceros en los tiempos de muestreo T mediante un filtro transversal, se puede lograr anular la ISI¹¹.

2.2 Los ecualizadores como filtros de forzamiento de zero.

En la presente sección desarrollamos la forma no adaptiva de los ecualizadores usados inicialmente por Lucky. Partimos de la ecuación (2.7) que la reproducimos a continuación por conveniencia, en donde hemos hecho: Tiempo de muestreo T = 1 seg. por facilidad de cálculo:

$$\bar{a}(t) = \sum_{i=-N}^N c_i y(t-i), \quad (2.7.a)$$

Si aplicamos el resultado consignado en la OBSERVACION 1, es decir:

$$\bar{a}(t) = \begin{cases} 1 & t=0 \\ 0 & t \text{ diferente de } 0 \end{cases}$$

a la ecuación (2.7.a), obtenemos un sistema de $2N + 1$ incógnitas en c_i que nos permite el objetivo deseado: anular la ISI.

¹¹ Feher K, Digital Communications. 1983 Prentice Hall. Pg.115-118

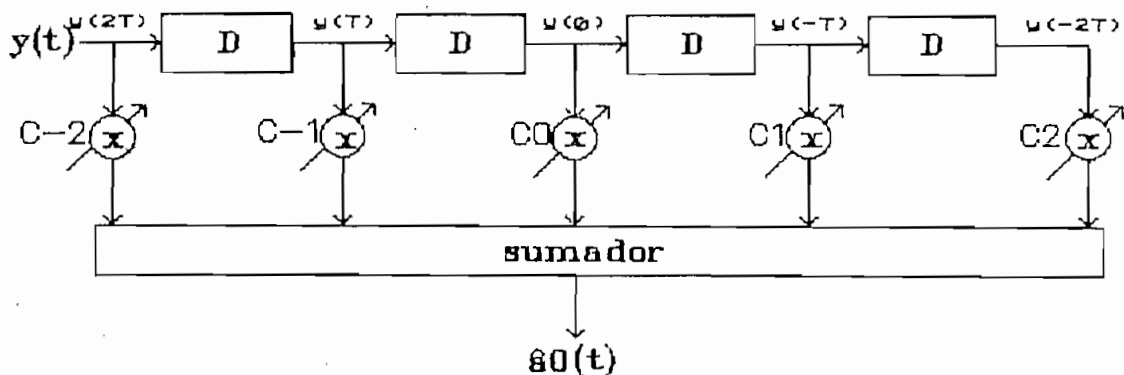


Figura 6 El Ecuador como Filtro Forzador de Ceros.

Para forzar ceros usaremos el filtro de la fig. 6, el cual es idéntico al de la figura 1. La diferencia estriba únicamente en la notación.

A continuación presentamos las ecuaciones que determinan las ganancias de los taps, c_i , considerando un sistema de 5 taps con $N=2$, a partir de (2.7.a):

$$\begin{aligned}
 t=-2 \quad \bar{a}(-2) = 0 &= c_{-2} y_0 + c_{-1} y_{-1} + c_0 y_{-2} + c_1 y_{-3} + c_2 y_{-4} \\
 t=-1 \quad \bar{a}(-1) = 0 &= c_{-2} y_1 + c_{-1} y_0 + c_0 y_{-1} + c_1 y_{-2} + c_2 y_{-3} \\
 t= 0 \quad \bar{a}(0) = 1 &= c_{-2} y_2 + c_{-1} y_1 + c_0 y_0 + c_1 y_{-1} + c_2 y_{-2} \\
 t= 1 \quad \bar{a}(1) = 0 &= c_{-2} y_3 + c_{-1} y_2 + c_0 y_1 + c_1 y_0 + c_2 y_{-1} \\
 t= 2 \quad \bar{a}(2) = 0 &= c_{-2} y_4 + c_{-1} y_3 + c_0 y_2 + c_1 y_1 + c_2 y_0
 \end{aligned}
 \tag{2.8.a.}$$

Lo que nos permite calcular los coeficientes c_i que aseguran transmisión libre de ISI.

Este método obviamente requiere que se efectúen mediciones de la respuesta impulsiva del canal, que a manera de ejemplo nos lo proporciona Feher en la citada referencia y es una función tipo $\{\text{sen}(y)/y\}$ con ruido adicional y da como muestras de $y(t)$ las siguientes:

$y(-2)=0.1$, $y(-1)=-.2$, $y(0)=1$, $y(1)=.1$ y $y(2)=0$;

lo que conduce al sistema de ecuaciones de la Tabla 1, en donde la primera columna contiene las a_i . A partir de ello es fácil demostrar que el sistema de ecuaciones tiene una solución única dada por el vector: $C = (c_{-2}, c_{-1}, c_0, c_1, c_2)$

donde

$c_{-2} = -0.054$
$c_{-1} = 0.2067$
$c_0 = 0.9595$
$c_1 = -0.094$
$c_2 = 0.009$

0	1	-0.2	0.1	0	0
0	0.1	1	-0.2	0.1	0
1	0	0.1	1	-0.2	0.1
0	0	0	0.1	1	-0.2
0	0	0	0	0.1	1

Tabla 1. Sistema de ecuaciones que fuerza el cruce por cero de $a(t)$

Todo lo cual nos permite tener una idea de la ingente cantidad de trabajo matemático que implica el proceso directo, sobre todo si hablamos de matrices de 11×11 , que son muy comunes.

Además de que si en el transcurso del tiempo el canal varía ligeramente, los coeficientes calculados por este método permanecen estáticos lo que daría lugar a una sustancial disminución del rendimiento del enlace.

OBSERVACION 2: El cálculo de los coeficientes nos permite una primera aproximación a lo que internamente sucede en la ecualización. Se ve que para un canal práctico, el corrimiento de

fase es más importante en los taps cercanos al tap central. Sin embargo debe anotarse que si bien el canal compensa el error en forma casi exacta para los tiempos analizados, al tiempo $t=-3$ el error es de $e = c^{-2} y^{-1} + c^{-1} y^{-2}$, (de (2.8.a) con $y^{-3} = y^{-4} = y^{-5} = 0$), lo que con los valores calculados de los taps da 0.3. Ello indica que el orden escogido de 5 no es suficiente. Si deseamos reducir el error, debemos aumentar el orden a 7, con lo que los nuevos coeficientes son:

$$C = (-.03, -.051, .2064, 0.9596, -.094, .009, -.0009).$$

Con el nuevo orden de 7, en $t=-3$, $e = c^{-3} y^0 + c^{-2} y^{-1} + c^{-3} y^{-2} = 0.00084$.

Para $t=-4$, $e = c^{-3} y^{-1} + c^{-2} y^{-2} = .0009$ y para $t = -5$, $e = c^{-3} y^{-2} = -.003$, con lo que ya no hace falta aumentar más el orden del filtro.

Por lo anteriormente visto, en lugar de un proceso matemático directo, se usará un método recursivo que trata de minimizar la ISI por el método del error cuadrático medio, el cual desemboca en una secuencia de cálculos para actualizar periódicamente los taps del TDL. Esto lo veremos en el siguiente capítulo.

2.3 Criterios de Nyquist de Interferencia Intersímbolo.

En esta sección veremos que la simetría en el dominio de la frecuencia que presenta la función de transferencia del canal, conduce a los pulsos coseno elevado para minimizar la interferencia intersímbolo, lo que constituye el primer enunciado de Nyquist.

Sabemos del análisis espectral de Fourier que la forma de onda correspondiente a un filtro pasabajos ideal de Frecuencia de

Corte f_n , es la función $\text{sen}(2 \pi f_n t)/(2 \pi f_n t)$, donde $f_n = B =$ Frecuencia de Nyquist, que será explicada posteriormente.

Este pulso lo mostramos en la fig. 7, de la cual se observa que la interferencia intersímbolo es anulada, puesto que en el caso ideal, de sincronización perfecta, los pulsos configurados con este filtro no interaccionan entre sí, ya que en los tiempos de muestreo el pulso resultante es el deseado, siendo la resultante de todos los demás pulsos igual a cero.

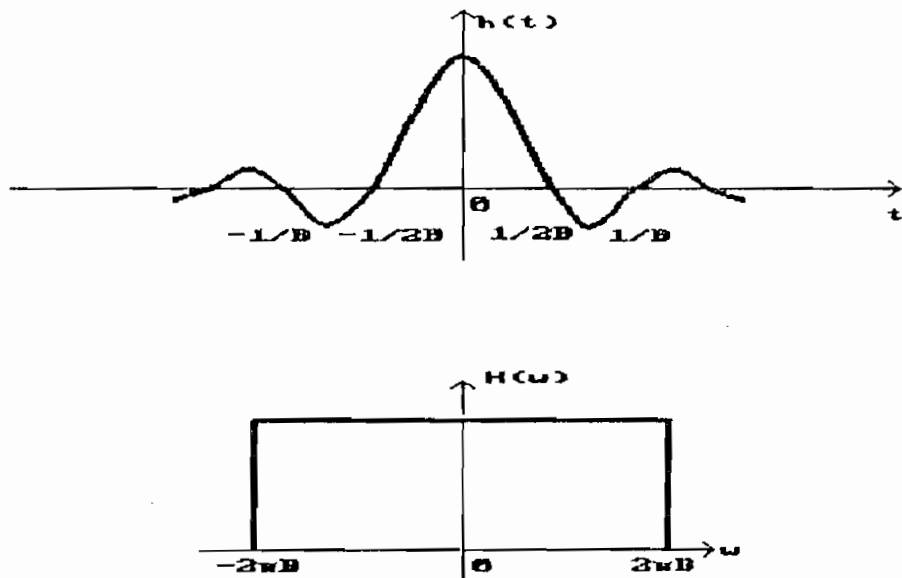


Figura 7 Pulso que proporciona una interferencia intersímbolos nula.

Si se escoge $1/2f_n$ como el intervalo de muestreo T ¹², ó en otras palabras: $f = 2 f_n$ (que es un caso especial del Teorema de Nyquist $f \geq 2 f_n$, donde f_n es el ancho de banda de la señal), se pueden transmitir $2f_n$ pulsos por segundo si se emplea esta forma de onda. Esta velocidad no es más que la conocida velocidad de Nyquist.

Sin embargo, existen dificultades prácticas con esta forma

¹² Schwartz, M. Comunicación Digital. 3ra Ed. Prentice Hall. 1987.

de onda:

1. El filtro pasabajos ideal es irrealizable y;

2. El pulso requiere una sincronización extremadamente precisa. Si el tiempo del receptor varía en algo de la sincronía exacta, la Interferencia Intersímbolos Nula desaparece, puesto que las colas de los pulsos podrían agregarse en forma divergente.

Para aliviar este problema, Nyquist (ob. citada) sugirió una clase de señales que eliminan la ISI y que son menos sensibles al sincronismo, que analizamos a continuación.

Si de la fig. 7 modificamos la característica de corte del FPB (filtro pasabajos) de modo que el corte sea gradual, el filtro es físicamente realizable. En particular:

- ▶ si la nueva función de transferencia se diseña para tener
- ▶ simetría impar alrededor del punto de corte pasabajos, se
- ▶ demostrará que la respuesta a impulso resultante mantiene la
- ▶ característica de tener ceros a intervalos uniformemente
- ▶ espaciados de tiempo.

Un ejemplo de tal espectro que se usa en la práctica es el espectro de amplitud coseno elevado. Este espectro se muestra en la fig.8. El espectro del coseno elevado está dado por:

$$H(w) = \begin{cases} 1/2 (1 + \cos (\pi w / 2 w_n)), & \text{abs } (w) \leq 2 w_n \\ = 0 & \text{otro caso} \end{cases} \quad (2.9)$$

Si medimos la frecuencia respecto a w_n haciendo $w=w_n + \Delta w$, véase la fig.8, tenemos:

$$H(w) = \begin{cases} 1/2 (1 + \cos \pi/2 (1+ \Delta w / w_n)) \\ = 1/2 (1 - \text{sen } \pi/2 \Delta w / w_n) \end{cases} \quad (2.10)$$

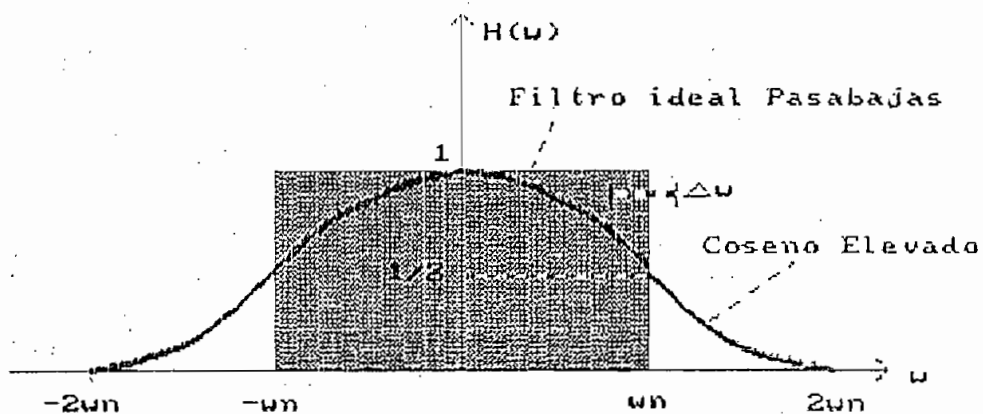


Figura 8 Espectro del coseno elevado.

Puesto que el término seno tiene simetría impar, el espectro del coseno elevado despliega la simetría impar indicada anteriormente.

La respuesta al impulso de un filtro con esta característica de frecuencia se demuestra (ver Apéndice III) que está dada por:

$$h(t) = \frac{\omega n}{\pi} \frac{\text{sen}(\omega n t)}{(\omega n t)} \cos(\omega n t) / (1 - (2\omega n t/\pi)^2) \quad (2.11)$$

Esta expresión contiene el término $\text{sen}(x)/x$ del filtro ideal multiplicado por un factor adicional que decrece con el aumento del tiempo.

El término $\text{sen}(x)/x$ garantiza los cruces por cero de $h(t)$ precisamente a los mismos espacios de tiempo igualmente distanciados que tiene el filtro ideal pasabajos. El factor adicional de multiplicación reduce las colas de los pulsos en forma considerable por debajo de $\text{sen}(x)/x$, de manera que cuando

este tipo de pulsos se emplean en la transmisión digital, son relativamente insensibles a las variaciones en la sincronización.

Para determinar cuantitativamente el porcentaje de disminución de la sensibilidad, expresemos (2.11) en forma normalizada:

sea $f = 2 f_n = 1/T$, $x = t/T$, entonces

$$h(t) = \frac{w_n}{\pi} \frac{\text{sen}(2 \pi f_n t)}{(2 \pi f_n t)} \cos(2 \pi f_n t) / (1 - (2 \pi f_n t / \pi)^2) \quad (2.12)$$

$$h(t) = \frac{w_n}{\pi} \frac{\text{sen}(\pi x)}{(\pi x)} \cos(\pi x) / (1 - (2 x)^2) \quad (2.13)$$

sea $f_n = 1/2$ Hz [para que la amplitud de (2.13) sea 1],

$$h(t) = \frac{\text{sen}(\pi x)}{(\pi x)} \cos(\pi x) / (1 - 4 x^2) \quad (2.14)$$

Para el caso del filtro ideal, cuando $h(t) = \text{sen}(\pi x) / (\pi x)$, el máximo valor negativo se produce cuando $t = 1.5 * T = 1.5 * 1 = 1.5$ ó $x = 1.5$, lo que da $h(1.5) = -.2122$; resultado que se deduce de la fig.7 con $f_n \triangleq B = 1/2$.

De (2.14), cuando se contempla la expresión completa, gráficamente hemos hallado que el mínimo valor se obtiene en $h(1.17) = -.0266$.

De esto se concluye que, aproximadamente, la sensibilidad disminuye en un orden de $-.2122 / -.0266 \approx 10$.

Empero, esta ventaja conlleva la desventaja de que, como se muestra en la fig.8, el ancho de banda necesario es el doble que

en el caso ideal.

Para reducir esta exigencia introduciremos posteriormente un factor de exceso de ancho de banda, alfa, el cual permite un compromiso entre disminución de sensibilidad al sincronismo y aumento del ancho de banda de transmisión.

A continuación se demuestra el hecho de que el espectro del coseno elevado es un ejemplo de una clase de espectros con simetría impar alrededor de ω_n (frecuencia angular de Nyquist o frecuencia de corte del Filtro Pasabajos [FPB] ideal), que proporcionan cruces por cero en intervalos igualmente espaciados de muestreo.

Supóngase que el FPB tiene la característica siguiente:

$$\begin{aligned} H(\omega) &= 1 + H_1(\omega) & \text{abs}(\omega) < \omega_n & & (2.15) \\ &= H_1(\omega) & \omega_n < \text{abs}(\omega) < 2\omega_n & \\ &= 0 & \text{en otro caso.} & \end{aligned}$$

(Si $H_1(\omega)=0$ se tiene el filtro ideal pasabajos)

Por ej. (ver la fig. 9), sea

$$H_1(\omega) = 1/2 (\cos(\pi/2 \omega/\omega_n) - 1) \quad \text{abs}(\omega) < \omega_n \quad (2.16a)$$

$$= 1/2 (\cos(\pi/2 \omega/\omega_n) + 1) \quad \omega_n < \text{abs}(\omega) < 2\omega_n \quad (2.16b)$$

que es justamente el caso del coseno elevado, como es inmediato de demostrar por comparación con (2.9).

En lo que sigue asumiremos que el corrimiento de fase del filtro es 0. Supóngase que $H(\omega)$ tiene simetría impar alrededor de ω_n . Entonces

$$H_1(\omega n + \Delta\omega) = -H_1(\omega n - \Delta\omega) \quad (2.17)$$

lo que se comprueba de (2.16b), y gráficamente se muestra en la fig. 9.

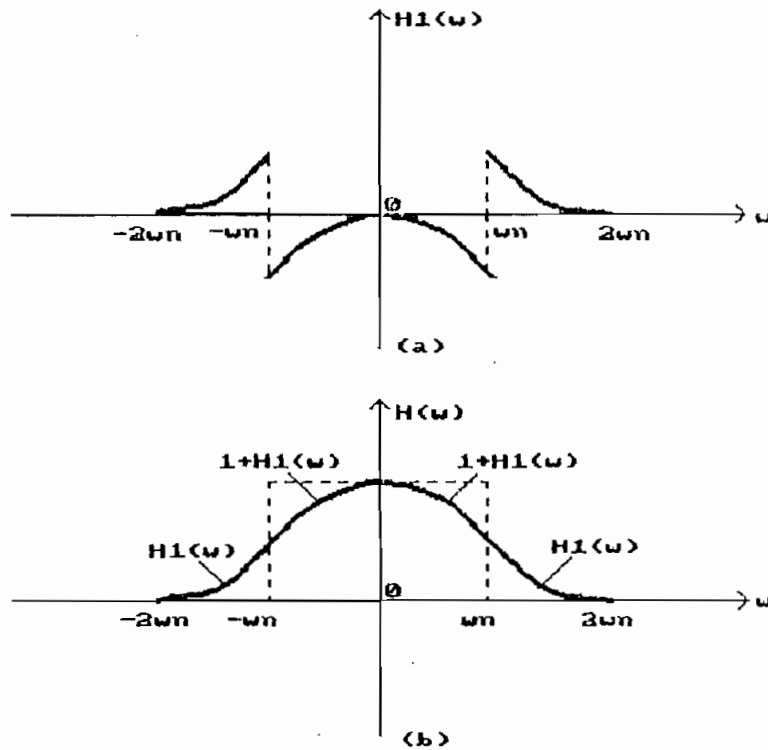


Figura 9 Filtro de Nyquist: a) Simetría impar alrededor de ωn . b) Filtro completo.

Tomando la Transformada de Fourier de la ecuación (2.15), se tiene, por superposición:

$$h(t) = \omega n / \pi \operatorname{sen}(\omega n t) / (\omega n t) + h_1(t) \quad (2.18)$$

$$\text{donde } h_1(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_1(\omega) \exp(j\omega t) d\omega \quad (2.19)$$

Se puede simplificar la expresión anterior tomando en cuenta

que H_1 es par, véase fig.9, y que todos los espectros de amplitud son simétricos respecto a $w=0$. Entonces (según se demuestra en el Apéndice III, para toda función par),

$$h_1(t) = \frac{1}{\pi} \int_0^{\infty} H_1(w) \cos(wt) dw \quad (2.19)$$

$$= \frac{1}{\pi} \int_0^{w_n} H_1(w) \cos(wt) dw + \frac{1}{\pi} \int_{w_n}^{2w_n} H_1(w) \cos(wt) dw \quad (2.20)$$

usando el hecho de que $H_1 = 0$ para $w > 2w_n$.

A continuación hagamos $w = w_n - x$ en la primera integral de (2.20) y $w = w_n + x$ en la segunda integral.

$$h_1(t) = \frac{1}{\pi} \int_0^{w_n} H_1(w_n - x) \cos[(w_n - x)t] dw \\ + \frac{1}{\pi} \int_{w_n}^{2w_n} H_1(w_n + x) \cos[(w_n + x)t] dw \quad (2.21)$$

La variable muda x varía entre 0 y w_n en ambas integrales por lo que las dos pueden ser combinadas en una sola; y haciendo uso de la propiedad de simetría impar (2.17),

$$h_1(t) = \frac{1}{\pi} \int_0^{w_n} H_1(w_n - x) (\cos(w_n - x)t - \cos(w_n + x)t) dx \quad (2.22)$$

Usando la identidad

$$\cos(a-b) - \cos(a+b) = 2 \operatorname{sen} a \operatorname{sen} b$$

se obtiene el siguiente resultado:

$$h_1(t) = \frac{2}{\pi} \sin \omega_n t \int_0^{\omega_n} H_1(\omega_n - x) \sin x t \, dx \quad (2.23)$$

Nótese que, independientemente del valor de la integral (el cual dependerá de la característica escogida para H_1), el $\sin \omega_n t$ precedente garantiza que $h_1(t)$ sea 0 a intervalos espaciados $T = \pi/\omega_n$ seg. Pero al ser este el intervalo original de muestreo, la ecuación (2.23) proporciona 0 en todos los tiempos de muestreo T . Esta propiedad de $h(t)$ es por supuesto consecuencia de la simetría impar que se ha elegido para $H_1(\omega)$ LQD.

De esta forma, cualquier función que presente simetría impar alrededor de ω_n proporciona señalización libre de ISI para transmisión a intervalos T de muestreo. En la práctica se elige funciones con un ancho de banda menor al del coseno elevado, pero

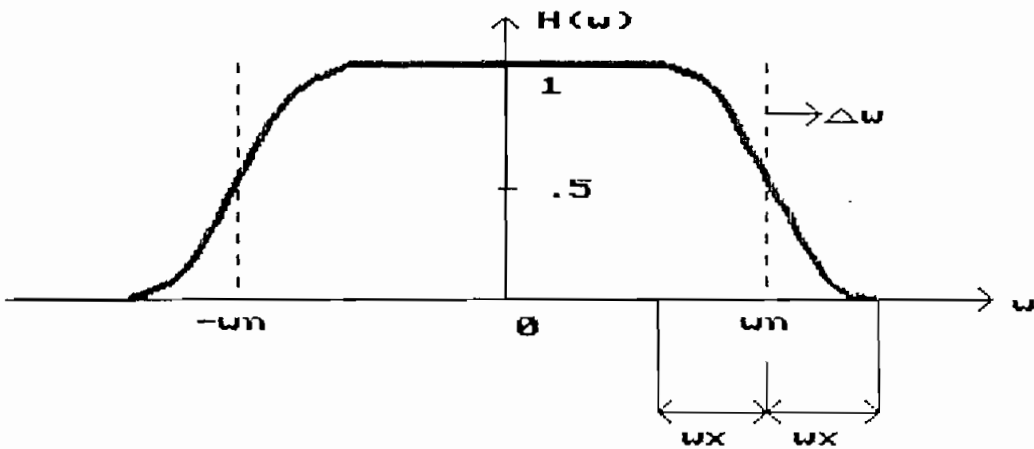


Figura 10 Espectro de caída sinusoidal

que conservan la insensibilidad ante la sincronización. El

se obtiene el siguiente resultado:

$$h_1(t) = \frac{2}{\pi} \sin \omega_n t \int_0^{\omega_n} H_1(\omega_n - x) \sin x t \, dx \quad (2.23)$$

Nótese que, independientemente del valor de la integral (el cual dependerá de la característica escogida para H_1), el $\sin \omega_n t$ precedente garantiza que $h_1(t)$ sea 0 a intervalos espaciados $T = \pi/\omega_n$ seg. Pero al ser este el intervalo original de muestreo, la ecuación (2.23) proporciona 0 en todos los tiempos de muestreo T . Esta propiedad de $h(t)$ es por supuesto consecuencia de la simetría impar que se ha elegido para $H_1(\omega)$ LQDD.

De esta forma, cualquier función que presente simetría impar alrededor de ω_n proporciona señalización libre de ISI para transmisión a intervalos T de muestreo. En la práctica se elige funciones con un ancho de banda menor al del coseno elevado, pero

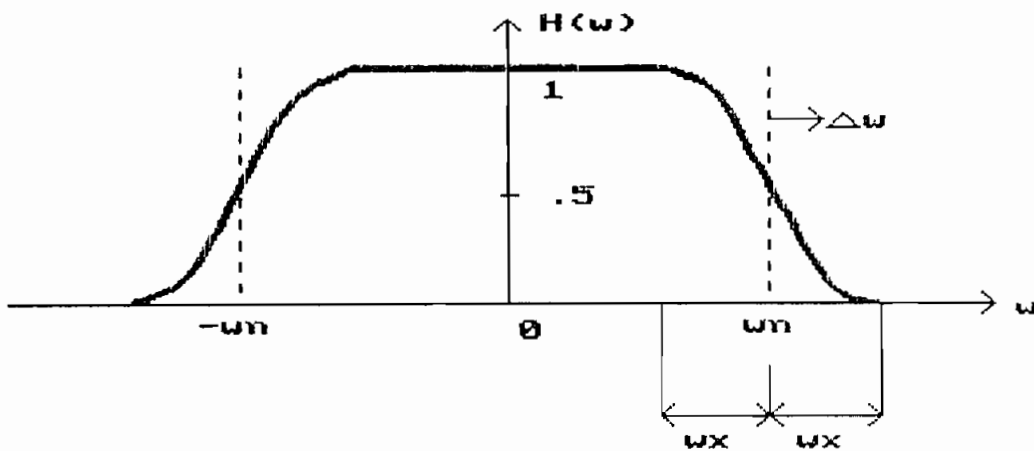


Figura 10 Espectro de caída senoidal

que conservan la insensibilidad ante la sincronización. El

espectro en este caso se denomina de caída senoidal, y se lo muestra en la fig.10.

La generalización para este tipo de pulsos se la obtiene de (2.10) como sigue:

$$H(w) = 1/2 (1 - \text{sen } \pi/2 \Delta w / w_n) \quad (2.10)$$

En general $w_x \neq w_n$ (w_x está definida en el gráfico 10), como se supuso en el filtro coseno elevado. Digamos por ejemplo que queremos que $H(w)$ sea 1 incluso hasta $w=0.9w_n$, lo que por cierto implica $w_x=.1w_n$.

Entonces, de (2.10)

$$1 = 1/2 (1 - \text{sen } (\pi/2 k (w-w_n) / w_n)) \quad (2.11)$$

$$1 = 1/2 (1 - \text{sen } (\pi/2 k (-.1w_n) / w_n)) \quad (2.12)$$

$$1 = 1/2 (1 + \text{sen } (\pi/2 k .1)) \quad (2.13)$$

donde k está por determinarse. La única forma de que se cumpla (2.13) es de que $k=1/.1$ ó lo que es lo mismo

$$k = 1/(w_x/w_n) \quad (2.14)$$

definamos $\alpha = 1/k$, de modo que:

$$\alpha = w_x/w_n = f_x/f_n \quad (2.15)$$

Este valor es conocido como factor de exceso de banda, y es el que establece el compromiso entre ancho de banda y sensibilidad a la sincronización. Alfa está entre 0, caso del filtro pasabajos ideal y 1, caso del coseno elevado.

En la simulación se usarán los siguientes valores, como recomienda el CCITT.

El CCITT recomienda usar un ancho de banda tal que la distorsión de propagación, de la cual hablaremos posteriormente, sea menor a la duración del símbolo. Permite, el organismo, que exista cierta distorsión en la transmisión a bajas velocidades, pero es más estricto en el caso de altas velocidades.

Velocidad de modulación [bits/s]	Exceso de banda alfa.	Ancho de banda [Hz.]
1200	.8	720 - 2880
1400	.6	680 - 2920
1600	.4	680 - 2920
1800	.25	675 - 2925
2000	.2	600 - 3000
2200	.15	535 - 3065
2400	.125	435 - 2965

Tabla 2. Exceso de ancho de banda alfa recomendado por el CCITT, tal que la distorsión de propagación sea menor a la duración del símbolo.

De esta manera, la generalización de (2.10), tomando el caso del espectro de caída senoidal, para los alfa establecidos, es:

$$H(w) = 1/2 (1 - \text{sen } \pi/2 \cdot 1/\text{alfa} \cdot \Delta w / w_n) \quad (2.16)$$

que para la codificación en el Programa en Lenguaje C hemos variado ligeramente, así:

$$H(w) = 1/2 (1 - \text{sen } (\pi/(2 \text{ alfa}) (f-f_n) / f_n)) \quad (2.17)$$

$$H(w) = 1/2 (1 - \text{sen } (\pi/(2f_n \cdot \text{alfa}) (f-f_n))) \quad (2.18)$$

$$H(w) = 1/2 (1 - \text{sen } (\pi T / \text{alfa} (f-f_n))) \quad (2.19)$$

que es la fórmula usada en el programa, en donde $T=1/2f_n$.

Para el intervalo entre 0 y $f_n(1+\alpha)$ queda entonces, con $\alpha = f_x/f_n$:

$$\begin{aligned}
 H(f) = & 1 && 0 < f < f_n(1-\alpha) \\
 & 1/2 (1 - \text{sen} (\pi T 1/\alpha (f-f_n))) && f_n(1-\alpha) < f < f_n(1+\alpha) \\
 & 0 && f > f_n(1+\alpha)
 \end{aligned}
 \tag{2.20}$$

De lo visto se puede colegir que el ancho de banda B está relacionado con el exceso de banda α a través de:

$$B = 1/2T (1+\alpha). \tag{2.21}$$

con lo que podemos hallar cuál es el ancho de banda B ($B=f_n$ [Hz]) que se requiere para poder transmitir a una velocidad de transmisión $1/T$ dada.

También, con B especificado, el número de pulsos por seg. que pueden transmitirse está dado por

$$1/T = 2B/(1+\alpha). \tag{2.22}$$

Por ej. supóngase que el ancho de banda permisible es de 2.4 KHz (para canales telefónicos). La máxima velocidad de pulsos en este canal es entonces de $2B=4800$ pulsos por seg. Cuando se usa un factor de exceso de banda, $\alpha = 0.25$, la cifra se reduce a 3840 pulsos por seg. Y si se usa el espectro del coseno elevado, la cifra se reduce a 2400 pulsos por seg.

En el caso de usar modulación, el ancho disponible del canal se divide para dos en el caso de transmisión con portadora ubicada en el centro de las características del canal. Por este motivo, se habla de que la máxima velocidad de transmisión con

portadora es de 2400 bits/s.

Este análisis, sin embargo, sólo toma en cuenta la distorsión de amplitud, ya que la distorsión de fase restringe la capacidad del canal telefónico a menos de 2.4 KHz, lo que hace necesario un procedimiento de ecualización.

2.4 Determinación de la Función de Transferencia de los Filtros de Transmisión y de Recepción.

A continuación analizaremos la influencia del ruido en la recepción de la señal, y se demostrará que la maximización de la relación señal a ruido conduce a la obtención de los filtros acoplados (matched filters). Esto permitirá el establecer las funciones de transferencia tanto del filtro de transmisión así como la del filtro de recepción. Se demostrará que estas funciones de transferencia que determinan la función de transferencia óptima del canal deben ser iguales entre sí e iguales a la raíz cuadrada del pulso de coseno elevado, lo que determina que el canal satisfaga el criterio de Nyquist y a la vez maximice la relación señal a ruido.

Para optimizar la relación señal a ruido, no sólomente la variación del ancho de banda del filtro, sino también de la forma de la característica del mismo pueden ajustarse para lograr tal propósito (Schwartz, ob. cit.).

Para demostrar que la variación de la característica optimiza la relación señal a ruido, considérese $f(t)$ aplicada a un filtro lineal con una función de transferencia $H(w)$. Definiendo $F(w)$ como la Transformada de Fourier de $f(t)$,

$$F(w) = \int_{-\infty}^{\infty} f(t) \exp(-jwt) dw \quad (2.23)$$

La señal de salida $s(t)$ está dada por la convolución de $f(t)$ y $h(t)$, lo que en el dominio de la frecuencia da $S(w)=F(w)H(w)$, por lo que tomando la Transformada Inversa de Fourier de $S(w)$,

$$s(t)=\frac{1}{2\pi} \int_{-\infty}^{\infty} F(w) H(w) \exp(jwt) dw \quad (2.24)$$

$$= \int_{-\infty}^{\infty} F(w) H(w) \exp(jwt) df, \quad w=2\pi f \quad (2.25)$$

La magnitud de $s(t)$ en el momento de muestreo " t_0 " es justamente la amplitud deseada de la señal de salida, sea $A=abs(s(t_0))$,

$$A=abs(s(t_0))=abs\left(\int_{-\infty}^{\infty} F(w)H(w)\exp(jw t_0)df\right) \quad (2.26)$$

El espectro de potencia del ruido blanco a la entrada del filtro se ha tomado como (ver Apéndice III),

$$G(f) = n_0/2 \quad (2.27)$$

El espectro de potencia a la salida del filtro es entonces (ver Apéndice III),

$$G_n(f) = n_0/2 \cdot abs(H(w))^2 \quad (2.28)$$

y la potencia promedio del ruido de salida (o voltaje de ruido cuadrático promedio a través de un resistor de 1 ohm) es

$$N = n_0/2 \int_{-\infty}^{\infty} \text{abs}(H(w))^2 df \quad (2.29)$$

con $\text{sqrt}(N)$ =ruido de salida rms en ausencia de señal.

Deseamos la elección de $H(w)$ tal que la relación $A/\text{sqrt}(N)$ sea máxima. Esto es lo mismo que maximizar el cuadrado de la relación, o A^2/N . Esta relación es justamente la relación de la potencia instantánea pico de la señal a $t=t_0$ respecto a la potencia promedio de ruido y la llamaremos relación de potencia pico de la señal al ruido ó SNR pico. Como la señal de entrada $f(t)$ se supone dada (señal NRZ), su contenido de energía

$$\int_{-\infty}^{\infty} f^2(t) dt \text{ es una constante.}^{13}$$

Llamando E a la energía, la siguiente identidad de Fourier es válida (Apéndice III, teorema de la energía),

$$E = \int_{-\infty}^{\infty} f^2(t) dt = \int_{-\infty}^{\infty} \text{abs}(F(w))^2 df \quad (2.30)$$

Dividiendo la relación SNR pico para la constante E no afectará la determinación de la relación máxima. Por lo tanto puede tomarse como un problema equivalente la maximización de la relación:

¹³ Si bien en la transmisión NRZ, los pulsos son randómicos, la energía contenida en el tren es definida, puesto que sabemos que la transformada de Fourier del tren NRZ es

$$2A^2T(\text{sen}(\pi fT)/(\pi fT))^2,$$

lo cual es la densidad espectral, que es independiente de la transmisión randómica. Si a esta densidad la integramos en el ancho de banda definido por el filtro de transmisión, el cual es definido de antemano, obtendremos un resultado que es independiente del tren enviado y que es constante.

$$A^2/(EN) = \frac{\int_{-\infty}^{\infty} F(w)H(w)\exp(jwto)df}{no/2 \int_{-\infty}^{\infty} F(w)^2 df \int_{-\infty}^{\infty} H(w)^2 df} \quad (2.31)$$

Esto se hace fácilmente por medio de la desigualdad de Schwarz, que relaciona la integral de los productos de funciones complejas (ver Apéndice III para la demostración):

$$\left| \int_{-\infty}^{\infty} X(w)Y(w) dw \right|^2 \leq \int_{-\infty}^{\infty} |X(w)|^2 dw \int_{-\infty}^{\infty} |Y(w)|^2 dw \quad (2.32)$$

La igualdad en la desigualdad (2.32) se satisface sólo cuando $Y(w) = K X^*(w)$, K es un número real.

Si llamamos

$$X(w) = F(w) \exp(jwto), \quad Y(w) = H(w),$$

la relación: $no A^2/(2EN)$ de (2.31) debe ser menor o igual a 1,

$$\left| \int_{-\infty}^{\infty} F(w)H(w)\exp(jwto)dw \right|^2 \leq \int_{-\infty}^{\infty} |F(w)|^2 dw \int_{-\infty}^{\infty} |H(w)|^2 dw \quad (2.33)$$

En particular, la relación es máxima cuando la igualdad se cumple o cuando

$$H(w) = K (F(w) \exp(jwto))^* = K F^*(w) \exp(-jwto) \quad (2.34)$$

Los filtros que poseen la característica de la ecuación

(2.34) se llaman filtros acoplados (matched filters).

A continuación veremos cómo se usa la ecuación (2.34) para determinar la función de transferencia de los filtros de transmisión y de recepción.

Para el efecto tenemos la fig.11, en la cual se muestra la cadena de filtros de Tx, Rx, teniendo a la entrada la señal $s(t)$.

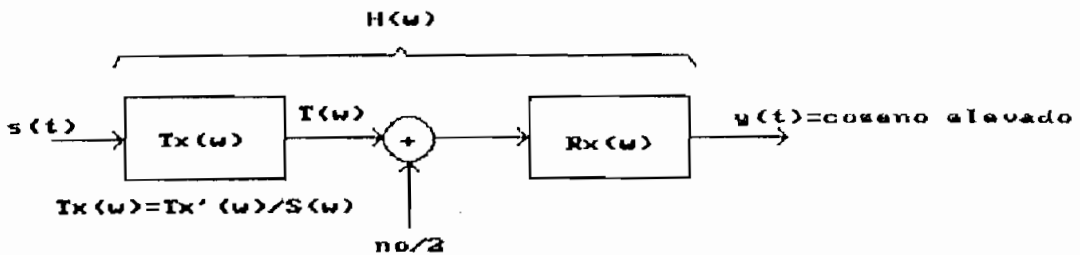


Figura 11 Determinación de las funciones de transferencia óptimas para los filtros de Tx y Rx.

Deseamos que a la salida $y(t)$ los pulsos que nos llegan sean tales que la ISI sea mínima. Esto se logra si $y(t)$ está conformada por pulsos de caída senoidal, los que ya fueron analizados, y cuya característica primordial era anular la ISI así como presentar baja sensibilidad a la sincronización (lo que como vimos depende del valor de alfa).

Por otro lado vemos que el ruido blanco está presente en el modelo únicamente a la entrada del filtro de Rx., lo que determina que tengamos libertad en la elección de la función de

transferencia del filtro de Tx.

Elegimos el filtro de Tx de tal forma que anule la función de transferencia particular que se use en la transmisión. Esto lo hacemos de modo que el análisis de señal a ruido no dependa del tipo de señal usada en la transmisión sino que dependa de la conformación de caída senoidal que asegura ISI nula. Sin embargo, al final veremos que la relación señal a ruido incluso no depende de esta forma de onda particular.

Sea T_x la FT del filtro de Tx, la libertad de elección a que hacemos alusión permite que:

$$T_x = T_x' / S \quad (2.35)$$

Es decir, a la salida del filtro de transmisión tenemos la señal T:

$$T = T_x' / S \quad S = T_x' \quad (2.36)$$

Para la elección de la función de transferencia del filtro de recepción, aplicamos la ecuación (2.34), que nos dice que la función de transferencia del filtro es igual al conjugado de la FT de la señal de entrada T. Por lo tanto, la señal deseada $y(t)$ se forma a partir de

$$Y = R_x T = H = \text{caída senoidal.} \quad (2.37)$$

con $T = T_x'$ de (2.36),

de (2.34)

$$R_x = K T_x'^* \exp(-j\omega t_0) \quad (2.38)$$

por lo que (con $t_0=0$, $K=1$)

$$Y = Tx'2 = H \quad (2.39)$$

De aquí resulta que la función de transferencia del filtro transmisor (y a través de (2.38) del receptor) es la raíz cuadrada de la función del canal H , la cual la hemos elegido como caída senoidal para anular la ISI. De (2.19):

$$R_x = \text{sqrt} (1/2 (1 - \text{sen} (\pi T 1/\text{alfa} (f-f_n)))) \quad (2.40)$$

que es la fórmula usada en el Programa en Lenguaje C para la FT del filtro de Recepción.

Para el filtro de recepción, puesto que estamos usando pulsos cuadrados, cuya transformada de Fourier es

$$S = \text{sen}(\pi fT)/(\pi fT) \quad (2.41)$$

de (2.35), tenemos

$$T_x = (\pi fT)/\text{sen}(\pi fT) \text{sqrt}(1/2 (1 - \text{sen}(\pi T 1/\text{alfa}(f-f_n)))) \quad (2.42)$$

que es la FT usada en el Programa.

Para finalizar veamos que la SNR no depende de la forma de onda caída senoidal.

Aplicando (2.34) a (2.31) se llega a

$$A^2/(EN) = 1/(n_o/2) \quad (2.43)$$

o

$$A^2/N = 2E/n_o \quad (2.44)$$

o

$$A/\sqrt{N} = \sqrt{2E/n_0} \quad (2.45)$$

La relación señal a ruido se ve que depende únicamente de la energía de la señal y de la densidad espectral del ruido blanco.

- ▶ *La dependencia de la SNR respecto de la forma de onda de la señal de entrada (en este caso la señal caída senoidal), ha sido*
- ▶ *eliminada por el uso del filtro acoplado.*

Dos señales de formas de onda diferentes pueden proporcionar la misma probabilidad de error en presencia de ruido blanco aditivo, siempre que tengan la misma energía y que estén filtradas por los apropiados filtros adaptivos para cada caso. *Es la energía de la señal la que proporciona finalmente la capacidad de detección en presencia de ruido.*

2.5 Recomendaciones del CCITT para limitar la distorsión de amplitud y de fase del canal de transmisión.

La distorsión de propagación o retardo de grupo es el parámetro más importante en la transmisión telefónica.¹⁴

Es causada por las diferentes componentes de frecuencia de la señal que recorren el canal a diferentes velocidades y llegan a diferente tiempo, originando un ensanchamiento del pulso, el cual genera la ISI analizada.

La principal fuente de distorsión de retardo es el efecto acumulativo de los varios filtros usados en sistemas FDM (Frequency Division Multiplex) y de aquí que este tipo de distorsión depende tanto del número de etapas repetidoras, como

¹⁴ Núñez. Tesis: Ecuación Digital. EPN. 1986.

de la longitud de la trayectoria.

En la fig.12. podemos observar la distorsión de Propagación así como la distorsión de Amplitud como recomienda el CCITT (Rec. M. 102, tomo IV, libro Naranja) para circuitos internacionales especializados.

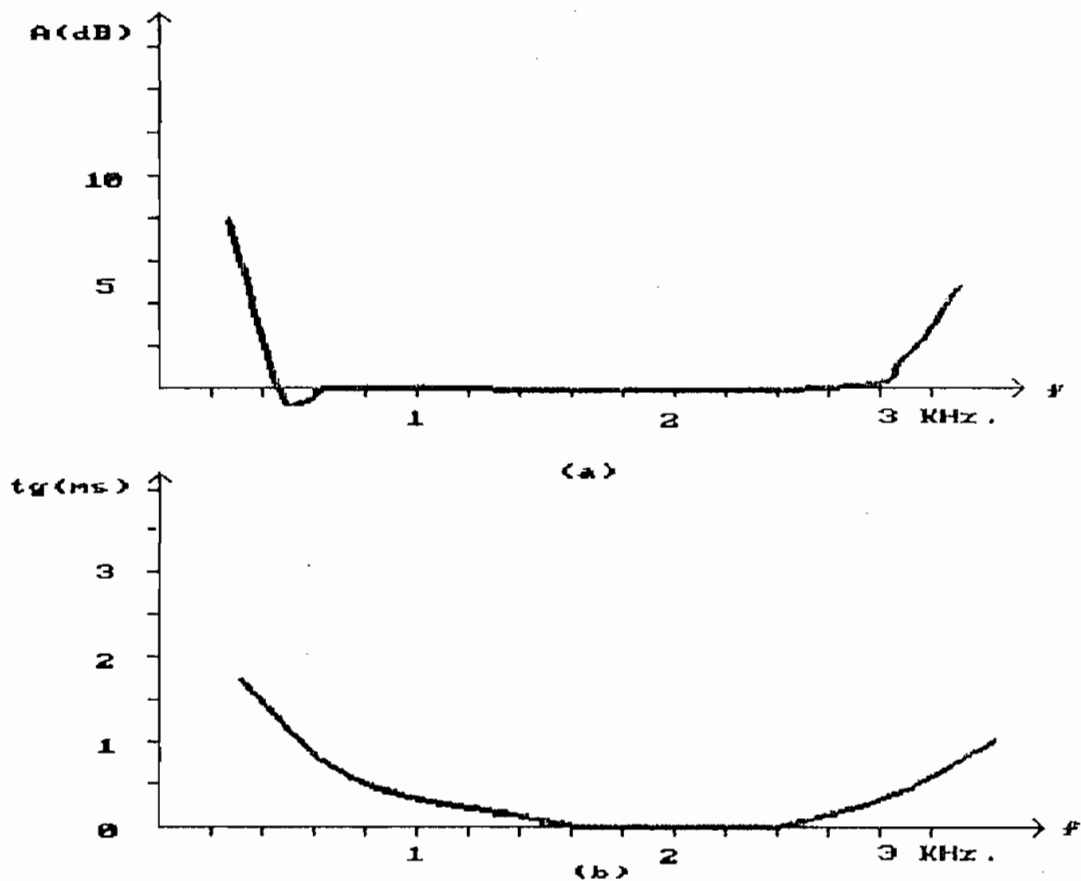


Figura 12 Distorsión de (a) Amplitud y de (b) Propagación en la Transmisión Digital

Como se demuestra en el capítulo 7, al seguir la recomendación del CCITT de usar una frecuencia de portadora de 1800 Hz, la principal fuente de distorsión es la distorsión de Propagación.

El canal usado, como recomienda el CCITT es el siguiente:

5. Para el rango 2700 - 3000 Hz.

$$A(f) = -.0001 f + 1.2274$$

$$t_g = (5.602747 \log_{10} f - 19.225) * .001 \text{ seg.}$$

6. Para el rango 3000 - 3200 Hz.

$$A(f) = -.0028125 f + 9.3548$$

$$t_g = (5.602747 \log_{10} f - 19.225) * .001 \text{ seg.}$$

Rangos que están graficados en la fig. 12.

Es este el modelo del canal que hemos implementado en el Programa en Lenguaje C como la subrutina canal.

2.6 Modelo de banda base para transmisión BPSK y explicación de las limitaciones que deben observarse en lo que respecta a amplitud y frecuencia en el caso de generalizar el modelo para una señal QPSK.

Puesto que el canal telefónico posee una configuración pasabanda (deja pasar frecuencias desde 300 a 3400 Hz), y como generalmente, el equipo de multiplexación en frecuencia para transmisión FDM precisa filtros que disminuyen aún más el ancho de banda útil del canal telefónico, los sistemas de transmisión de datos usan modulación para colocar el espectro de la señal en la banda usable; y usan demodulación para recuperar los datos.¹⁵

En la presente simulación se usa la modulación BPSK, la cual

¹⁵ Adams, P. Adaptive Filter in Telecommunications. En la obra citada de Grant and Cowan.

modula la fase de la onda portadora, haciéndola variar entre 0 y 180°. La señal modulada se puede expresar como:

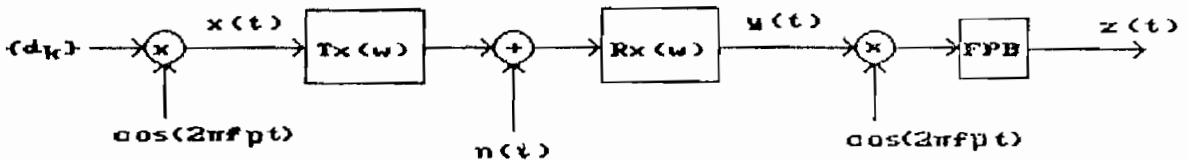
$$x(t) = \sum_{k=-\infty}^{\infty} d_k \cos(2\pi f_p t) \quad (2.47)$$

donde

$$d_k = +1, -1 \quad (2.48)$$

es la secuencia binaria NRZ que proporciona el cambio de fase mencionado a la portadora coseno, de frecuencia f_p .

El diagrama de bloques del modem BPSK se presenta en la figura 13.



donde: FPB=Filtro Pasa Bajos.

Figura 13 Diagrama de bloques del modem BPSK.

En la figura se puede observar que el filtrado de la señal se lleva a cabo luego de la modulación, lo cual sin embargo no siempre es lo más conveniente, puesto que si la portadora es de un valor elevado, digamos 1 MHz, y el ancho de banda es de 6 KHz, la implementación de un filtro pasabanda es más complicada que la de un filtro en banda base.

A continuación trataremos un modelo denominado análisis en banda base del sistema, el cual permite que un sistema con

portadora como es el caso del sistema BPSK, pueda ser pasado de la banda de paso (passband) a la banda base, lo cual simplifica notablemente el análisis. De esta forma, en lugar de diseñar complejos filtros en la banda de paso se diseñan filtros sencillos en banda base.

Dado un sistema en la banda de paso, si este presenta simetría alrededor de la frecuencia de portadora entonces es posible encontrar un sistema equivalente en banda base. (Feher, ob. cit.).

En la fig. 14. se muestra un esquema en banda base al cual ajustaremos el sistema en banda de paso BPSK.

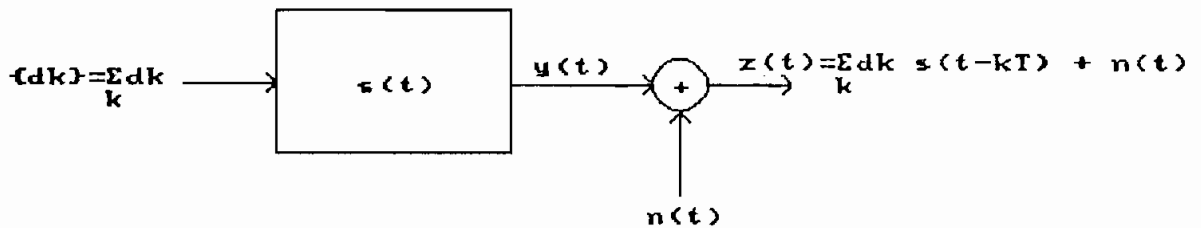


Figura 14 Sistema equivalente en Banda Base.

- En la figura se muestra dk , la información a transmitirse.

- $s(t)$ modela al canal de transmisión, es su respuesta impulsiva. En banda base, $s(t)$ representa el filtro de Tx y el filtro de Rx. En la banda de paso, $s(t)$ *representa además los sistemas de modulación y de demodulación.*

- El ruido aditivo $n(t)$ se añade a la señal filtrada por $s(t)$.

La fuente de información $\{dk\}$ tiene un ritmo de transmisión $f = 1/T$.

A partir de la figura 13 nos interesa encontrar las condiciones bajo las cuales un filtro en la banda de paso puede ser reemplazado por un filtro en banda base.

Supongamos que en lugar del filtro pasabanda de la fig. 13 tenemos un filtro pasabajos antes de la modulación, de modo que la señal que se transmite por el canal tiene la forma:

$$y(t) = [d(t) * s_L(t)] \cos(\omega_p t) \quad (2.49)$$

donde $*$ denota convolución y el subíndice L significa que es un filtro pasabajos (Low). La transformada de Fourier de (2.49) da:

$$Y(f) = [D(f) S_L(f)] * C(f) \quad (2.50)$$

donde $C(f)$ es la transformada del $\cos(\omega_p t)$.

La convolución de un espectro en banda base con una señal portadora sinusoidal resulta en un espectro de doble banda lateral (DSB) centrado alrededor de la portadora (Feher, ob. cit.):

$$Y(f) = 1/2 D(f-f_p)S_L(f-f_p) + 1/2 D(f+f_p)S_L(f+p). \quad (2.51)$$

Considerando, por otra parte, que el filtrado se realiza en la banda de paso (B),

$$y(t) = [d(t) \cos(\omega_p t)] * s_B(t) \quad (2.52)$$

lo que conlleva a:

$$\begin{aligned}
 Y(f) &= [D(f)*C(f)] S_B(f) \\
 &= [1/2 D(f-f_p) + 1/2 D(f+f_p)] S_B(f)
 \end{aligned}
 \tag{2.53}$$

se tiene que $Y(f)$ no varía al variar el tipo de filtrado si

$$S_B(f) = S_L(f-f_p) + S_L(f+f_p), \tag{2.54}$$

es decir si la configuración del filtro en la banda de paso es la misma que en la banda base.

Por esta razón, es posible analizar un sistema en la banda de paso mediante un sistema en banda base y esto además permite filtrar la señal antes de modularla, lo que por la ecuación (2.54) conduce a los mismos resultados.

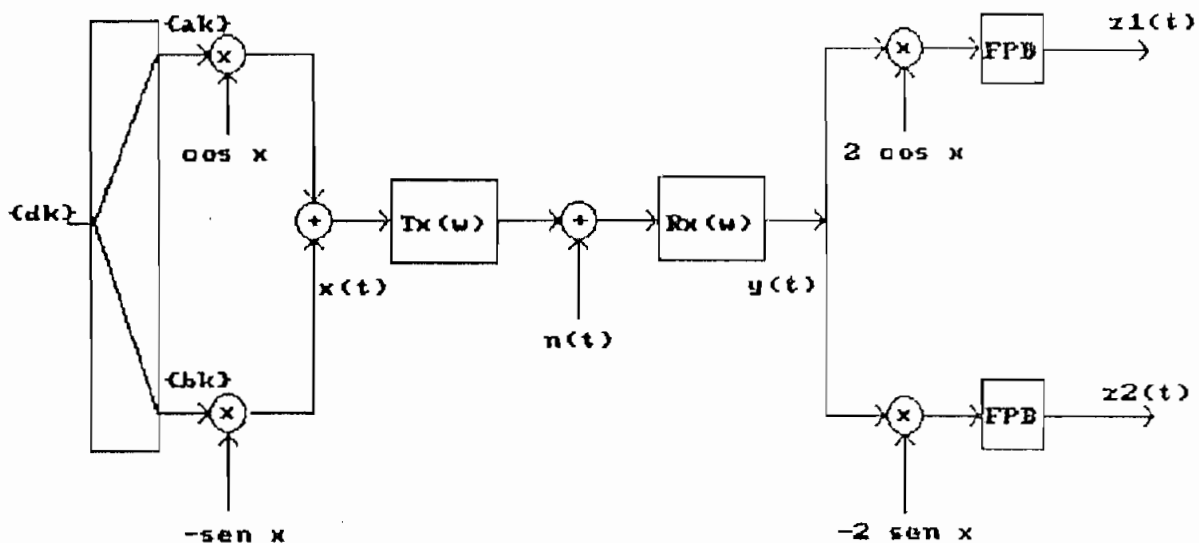
Es posible aplicar el análisis en banda base a la transmisión con portadoras en cuadratura QPSK.

En este caso se hace necesario considerar dos trenes, un tren NRZ a_k y un tren b_k , los cuales son obtenidos a partir de un tren d_k , por lo que presentan un período que es igual al doble del período original.

Se puede demostrar (Núñez, ob. cit.), que el producto de convolución de los dos trenes, a_k y b_k , con el filtro $s(t)$ en la banda de paso (que está constituido por el producto $T_x(w) R_x(w)$, -fig. 15) es equivalente a filtrar en banda base con dos filtros $s_1(t)$ y $s_2(t)$, tales que cumplan (2.54), con lo que el producto de convolución da :

$$\begin{aligned}
 y(t) &= [\sum a_k s_1(t-kT) - \sum b_k s_2(t-kT)] \cos(2\pi f_p t) \\
 &\quad + [\sum b_k s_1(t-kT) + \sum a_k s_2(t-kT)] \sin(2\pi f_p t) \\
 &\quad + n_o(t) \cos(2\pi f_p t) - n_e(t) \sin(2\pi f_p t)
 \end{aligned}
 \tag{2.55}$$

A la salida del demodulador, tenemos las señales $z_1(t)$ y



donde: $x=2\pi f_p t$, FPB=Filtro Pasa Bajos.

Figura 15 Esquema de Transmisión QPSK.

$z_2(t)$ que representan las partes real e imaginaria de la señal $z(t)$ (en donde el filtro pasabajos de recepción ha removido la portadora):

$$z_1(t) = \sum a_k s_1(t-kT) - \sum b_k s_2(t-kT) + n_c(t) \quad (2.56)$$

$$z_2(t) = \sum a_k s_2(t-kT) + \sum b_k s_1(t-kT) + n_s(t) \quad (2.57)$$

Con la siguiente notación

$$s(t) = s_1(t) + j s_2(t) \quad (2.58)$$

$$n(t) = n_c(t) + j n_s(t) \quad (2.59)$$

$$d_k = a_k + j b_k \quad (2.60)$$

$z(t)$ se puede escribir:

$$z(t) = \sum_k d_k s(t-kT) + n(t) \quad (2.61)$$

que no es más que la expresión matemática para el sistema de banda base que se muestra en la fig. 14.

Por esta razón podemos decir que el análisis realizado para la modulación BPSK puede ser extendido al análisis QPSK siempre y cuando se trate con señales complejas en todas las fórmulas que en el caso BPSK presentan valores reales.

Esto evidentemente complica el análisis por cuanto por cada producto real en modulación BPSK, en la modulación QPSK se tiene un producto complejo que equivale a 4 productos reales.

Debemos anotar además que la forma de las ecuaciones (2.56) y (2.57) revela una interacción entre los canales a_k y b_k . Esto nos indica que la distorsión en un sistema QPSK, a la vez que introduce interferencia intersímbolo en cada canal, introduce interferencia entre los canales, lo cual debe ser tomado en cuenta al modelar un canal QPSK.

Debemos señalar también que nuestro análisis BPSK ha sido llevado a cabo, entre otras, por las siguientes razones:

- A pesar de que se usa una computadora 80386 a 25 MHz (sin coprocesador), el tiempo de cada corrida dura alrededor de 15 a 20 minutos, lo que incluye las transformadas de Fourier, la generación de ruido gaussiano, el filtrado, etc. Este tiempo es de por sí elevado como para multiplicarlo por 4, con lo que estaríamos hablando de 1 hora y media aproximadamente en el caso de modular con QPSK.

- Por otro lado, los papers utilizados usan la modulación BPSK

para demostrar las propiedades de los algoritmos de ecualización.

Con ello, creemos que está justificado el análisis BPSK, pudiendo expandirlo a QPSK siempre que se apliquen las limitaciones que detallamos a continuación, en lo que respecta a frecuencia y potencia .

Se debe tomar en cuenta que un tren QPSK posee una frecuencia que es la mitad de la correspondiente al sistema BPSK. Además la potencia del tren QPSK es el doble de la potencia del tren BPSK, lo cual se ve de la ecuación dada por Feher (ob. citada) para el espectro de potencia de la señal BPSK:

$$2A^2T(\text{sen}(\pi fT)/(\pi fT))^2 \quad (2.62)$$

Tomando en cuenta que cada tren QPSK posee la mitad de la velocidad de la secuencia original, por lo que $T_q=2T$ (y los símbolos compuestos que mencionaremos luego tienen por tanto esta nueva velocidad), se tiene

$$\begin{aligned} \text{NRZ}_{\text{QPSK}}(w) &= 2A^2T_q(\text{sen}(\pi fT_q)/(\pi fT_q))^2 \\ &= 2A^22T(\text{sen}(\pi(f-f_p)2T)/(\pi(f-f_p)2T))^2 \\ &= 4A^2T(\text{sen}(2\pi(f-f_p)T)/(2\pi(f-f_p)T))^2 \end{aligned} \quad (2.63)$$

donde: A = amplitud de la señal a través de una $R=1\Omega$
(A=1 para NRZ).

f_p = frecuencia portadora.

T = 1/f = período de cada tren.

T_q = 1/ f_q = período del tren compuesto.

Por lo que (2.63) da

$$\text{NRZ}(w) = 4 T(\text{sen}(2\pi(f-f_p)T)/(2\pi(f-f_p)T))^2 \quad (2.64)$$

Y de ello se deduce que para transmisión QPSK, primero se

duplica la potencia respecto del caso BPSK, y segundo, los cruces por cero se producen para $f = f_p + f/2$, con lo que la frecuencia de corte del tren original de 1's y 0's se disminuye a la mitad al usar el tren QPSK de símbolos compuestos 00,01,10 y 11. Pero el producto Potencia Frecuencia permanece constante. Es decir, el alfabeto de símbolos enviados a partir de los originales 1 y 0 es 00,01,10 y 11, los cuales demandan el doble de potencia que los originales, pero a su vez requieren tan sólo la mitad de velocidad original, lo que permite ahorrar ancho de banda a expensas de la potencia (que no es más que el resultado del celebrado teorema de Shannon sobre la capacidad de un canal $C=B \log_2(1+P/(noB))$ donde P es la potencia de la señal, noB es la varianza del ruido blanco (Apéndice III:ruido blanco a través de un filtro pasabajos ideal) y B es el ancho de banda)

CAPITULO III.

CAPITULO III.

EL FILTRO TDL.

3.1 Estructura del filtro TDL.

En la figura 16. se muestra un Filtro TDL con 7 taps.

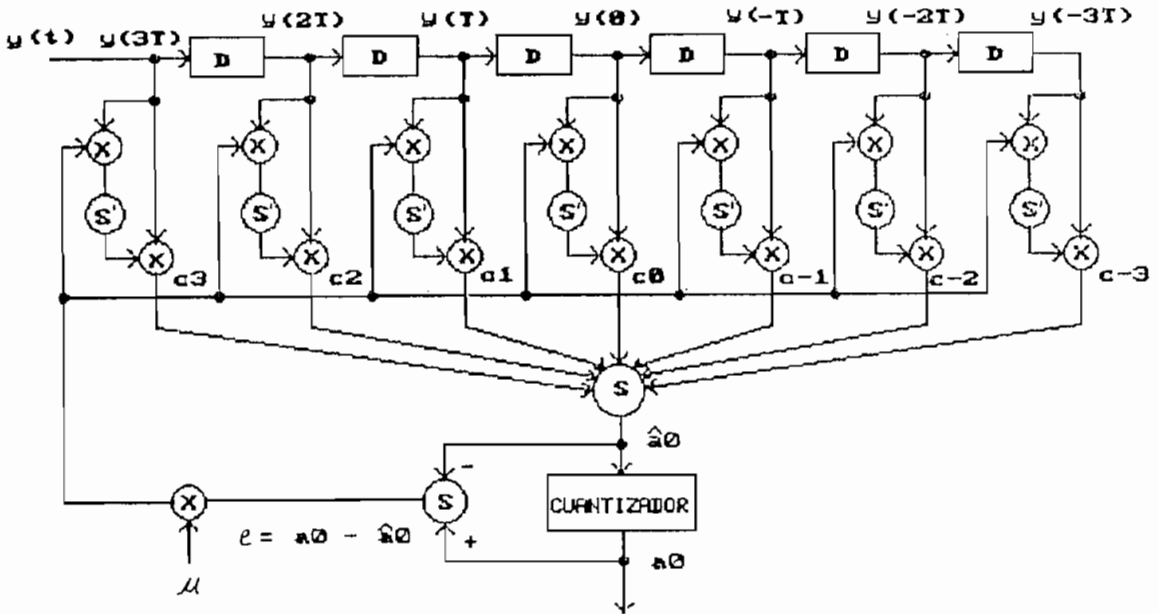


Figura 16 El Filtro Adaptivo Línea de Retardo con Derivaciones

Las siguientes operaciones se llevan a cabo sobre las $(2N+1)$ muestras almacenadas en la línea de retardos.

La señal $y(t)$ a la entrada del filtro `línea de retardos con derivación` (TDL) es muestreada sincrónicamente cada T seg. Debe tratarse de que el muestreo ocurra en el pico del pulso $y(t)$.

Cada muestra es alimentada al filtro TDL, el cual tiene $(2N+1)$ taps, de manera que la línea de retardo mantiene la información correspondiente a $(2N+1)T$ seg. El número de taps en el filtro `línea de retardos con derivación` es un parámetro de diseño elegido suficientemente grande para alcanzar la deseada

reducción de la ISI que resulta de la dispersión introducida por el canal limitado en banda¹⁶

La muestra almacenada en cada registro D (que retarda 1 unidad de tiempo a la señal) es multiplicada por la ganancia del tap correspondiente a dicho registro y se forma la suma de los productos de todos los taps para dar un estimado del símbolo de información deseado -referirse a la ecuación (2.8), que la repetimos a continuación.

$$\bar{a}_0 = \sum_{i=-N}^N c_i y(iT) \quad (2.8)$$

Continuando con la operación del TDL, tenemos que el estimado \bar{a}_0 es cuantizado al símbolo más cercano en el alfabeto de los símbolos transmitidos, asumiendo que la tasa de error es baja. Se producirá un error si el valor cuantizado de \bar{a}_0 no es y_0 .

Se observa además las operaciones de ajuste de las ganancias de los taps del TDL, que deben ser elegidas de modo de que el *Error Cuadrático Medio (MSE)* entre el símbolo deseado a_0 y el estimado \bar{a}_0 sea minimizado (Proakis and Miller, obra citada).

En la siguiente sección se derivan las ecuaciones de las ganancias de los taps que conducen al mínimo MSE.

¹⁶ Proakis and Miller. Adaptive Receiver For Digital Signaling. IEEE Transactions on Information Theory, reproducido en Lewis E. Franks, Baseband Communications, IEEE Press, 1971

3.2 El método de los Mínimos Cuadrados de Predicción Lineal y la determinación de las Ganancias Optimas de los taps del filtro TDL.

El filtro TDL descrito, forma un estimado del símbolo recibido en cada intervalo de señalización. Nos interesa el símbolo deseado a_0 . El MSE entre el símbolo deseado a_0 y el estimado \bar{a}_0 es:

$$\epsilon = E | e |^2 = E | a_0 - \bar{a}_0 |^2 \quad (3.1)$$

donde E da cuenta de la esperanza matemática o valor esperado.¹⁷

De (2.8) y (3.1):

$$\epsilon = E | a_0 - \sum_{k=-N}^N c_k y(kT) |^2 \quad (3.2)$$

El método del error medio cuadrático proporciona los coeficientes c_i óptimos derivando el error medio cuadrático ϵ con respecto a los coeficientes c_i .

$$\begin{aligned} \partial\epsilon/\partial c_i &= E \partial | a_0 - \sum_k c_k y(kT) |^2 / \partial c_i = 0 \\ &= - 2 E [| a_0 - \sum_k c_k y(kT) | y(iT)] = 0 \\ &= E [| a_0 - \sum_k c_k y(kT) | y(iT)] = 0 \end{aligned} \quad (3.3)$$

En la segunda línea de (3.3) hemos introducido la derivación

¹⁷ $E(x(kT)) = \sum_{k=-NT}^{NT} x(kT) f(kT),$

donde f =densidad de probabilidad.

al sumatorio, puesto que el índice del sumatorio es temporal y el de la derivada es ci .

Rearreglando (3.3) se obtiene:

$$E [|e| y(iT)] = 0 \quad (3.4)$$

Ecuación que nos dice que el error es ortogonal a las muestras de la señal, propiedad que usaremos profusamente en la derivación de ecuaciones de actualización de orden y tiempo en los Filtros Lattice (o filtro de Malla).

A partir de (3.3) tenemos:

$$E [a_0 y(iT) | - E [y(iT) \sum_k c_k y(kT)]] = 0 \quad (3.5)$$

De igual manera, si derivamos para $i=0, \dots, i-1, i+1, \dots, 2N+1$ obtendremos el número de ecuaciones necesario para formar un sistema único de ecuaciones:

$$E [a_0 \sum_i y(iT) | - E [\sum_i y(iT) \sum_k c_k y(kT)]] = 0 \quad (3.6)$$

Con la notación matricial-vectorial introducida en la ecuación (1.3), y para cualquier $t=n$,

$$E [a(n) Y^T(n) | = E [Y^T(n) C^{T_{opt}} Y(n)] \quad (3.7)$$

Asumiendo que el vector de coeficientes C^T es no correlacionado con el vector de la señal $Y(n)$, y que es óptimo por haber minimizado el error en el sentido del MSE,

$$E [a(n) Y^T(n) | = C^{T_{opt}} E [Y(n) Y^T(n)] \quad (3.8)$$

Los términos de expectación de (3.8) los definimos de la siguiente forma:

$P = E |a(n) Y(n)| =$ correlación cruzada entre la señal deseada al tiempo $t=n$ y el vector de señales estimadas.

$R = E |Y(n) Y^T(n)| =$ matriz de autocorrelación de las muestras de entrada $y(nT)$

Con lo que la ecuación (3.8) se puede escribir

$$P^T = C^{T\text{opt}} R \quad (3.9)$$

A esta ecuación se le reconoce generalmente como la ecuación de Wiener-Hopf, la cual da el set óptimo (en el sentido del MSE) de coeficientes $C^{T\text{opt}}$.

$$C^{T\text{opt}} = R^{-1} P^T \quad (3.10)$$

3.3 Análisis del algoritmo recursivo para la determinación de las Ganancias de los taps.

El set de $(2N+1)$ ecuaciones simultáneas para las ecuaciones de las ganancias de los taps que minimizan el MSE se puede resolver a partir de (3.10), invirtiendo la matriz de covarianza R . Sin embargo, la inversión directa de una matriz es un proceso muy laborioso, máxime que el canal está recibiendo datos continuamente y la solución de las ecuaciones implicadas debe ser continua.

Usaremos un proceso iterativo¹⁸ basado en el método del descenso más pronunciado para resolver (3.9).

Inicialmente uno empieza eligiendo valores para el conjunto

¹⁸ Proakis and Miller: Adaptive Receivers for Digital Signaling. Obra citada.

$\{C_k\}$ de ganancias de los taps. Esta elección inicial de las ganancias de los taps corresponde a algún punto j en la superficie cuadrática del MSE, en el espacio $(2N+1)$ dimensional de las ganancias de los taps¹⁹. El conjunto $(2N+1)$ de componentes $\partial\epsilon/\partial c_{kp}$ del gradiente es calculado en el punto j en la superficie del MSE, y cada ganancia de los taps es cambiada en la dirección contraria al correspondiente componente de gradiente, de modo de bajar por la superficie hiperparaboidal hacia el mínimo absoluto. *El cambio en la ganancia del k -ésimo tap es proporcional al tamaño del k -ésimo componente del gradiente.*

Denotemos $\{C_k^{(0)}\}$ a los valores iniciales de las ganancias de los taps. De esta forma, los sucesivos valores son obtenidos de acuerdo a la relación:

$$C_k^{(j+1)} = C_k^{(j)} - \mu/2 \partial\epsilon^{(j)}/\partial c_{kp} \quad (3.11)$$

de (3.3)

$$C_k^{(j+1)} = C_k^{(j)} + \mu E |e^{(j)} y(kT)| \quad (3.12)$$

con $k = -N, \dots, -1, 0, 1, \dots, N$

donde $C_k^{(j)}$ es el valor de la ganancia del k -ésimo tap en la j -ésima iteración, y lo mismo se estipula para el gradiente y para el error. $y(kT)$ es la muestra de señal más ruido almacenada en el k -ésimo tap del filtro TDL, y μ es un número positivo elegido suficientemente pequeño de modo de asegurar la

¹⁹ Las ganancias C^T de los taps generan una superficie hiperparabólica como se demuestra a continuación:

$$\epsilon_j = E |(a(j) - C_j^T Y(j))|^2$$

$$\epsilon_j = E |a^2(j) - 2 C_j^T P_j + C_j^T R_j C_j|$$

convergencia del procedimiento iterativo. El procedimiento continúa hasta que para algún orden j_0 , $C_k^{(j+1)} = C_k^{(j)}$ por lo que $E |e^{(j)} y(kT)| = 0$.

La dificultad con el método radica en la falta de conocimiento de los componentes del gradiente $\partial E^{(j)} / \partial c_k$ o su equivalente, los valores de las correlaciones cruzadas $E |e^{(j)} y(kT)|$, así:

$$\partial E^{(j)} / \partial c_k = \partial (E |a(n) - C^T Y(n)|^2) / \partial c_k \quad (3.13)$$

$$= 2 E (|a(n) - C^T Y(n)| Y(n)) \quad (3.14)$$

$$= 2 E (|e(n)| Y(n)) \quad (3.15)$$

Estas cantidades dependen de los valores muestreados $r(t-kT)$ de la respuesta impulsiva $r(t)$ y de los momentos de segundo orden entre el ruido y la secuencia $y(kT)$, lo cual no es conocido en el receptor. Para superar esta dificultad, uno puede usar estimados ruidosos de las correlaciones cruzadas, como sugiere Widrow²⁰. Así, las ganancias son ajustadas de acuerdo a la relación:

$$C_k^{(j+1)} = C_k^{(j)} + \mu [e^{(j)} y(kT)], \quad (3.16)$$

$$k = -N, \dots, -1, 0, 1, \dots, N$$

donde $[e^{(j)} y(kT)]$ es el producto entre el error en la j -ésima iteración y la muestra de señal $y(kT)$ en el k -ésimo tap en la j -ésima iteración. Cada iteración corresponde a la entrada de una nueva muestra en el filtro TDL. Por lo que la relación (3.16) es realizada una vez cada T segundos.

A continuación detallamos el resto de operaciones que

²⁰ Ferrara and Widrow: Time-Sequenced Adaptive Filter, IEEE Transactions on Circuits and Systems, June 1981.

realiza el TDL de la figura 16. Para cada nueva muestra de señal más ruido que entra al filtro TDL, se forma un estimado \bar{a}_0 del símbolo de información recibido mediante la multiplicación de las muestras $\{y(kT)\}$ almacenadas en la línea de retardos por las correspondientes ganancias de los taps $\{C_k\}$ y sumando los $(2N+1)$ productos resultantes. Se forma una señal de error "e" tomando la diferencia entre el verdadero símbolo transmitido a_0 y el estimado \bar{a}_0 . Esta señal de error es realimentada y multiplica a la señal $y(kT)$ en cada tap. Los productos así formados son escalados en el factor μ y añadidos a los valores presentes de las ganancias de los taps de acuerdo a la relación dada en (3.16).

De esta manera son obtenidos nuevos estimados de las ganancias óptimas de los taps. Estas operaciones son repetidas cada vez que una nueva muestra de señal más ruido entra en el filtro TDL.

Nota: El sumatorio s' de la figura 16. que tiene que ver con la actualización de los taps funciona de la siguiente manera: A pesar de que un sumatorio debe tener dos entradas, este tiene una, sin embargo, la otra está implícita y es interna. Internamente el sumatorio debe almacenar la ganancia del tap en la k -ésima iteración, de modo de que en la siguiente iteración, internamente forma la suma de la ganancia previa con el producto externo $\mu [e^{(j)} y(kT)]$ para formar la ganancia en curso:

$$C_k^{(j+1)} = C_k^{(j)} + \mu [e^{(j)} y(kT)].$$

Durante el período de entrenamiento, se envía una señal con información conocida por el receptor.

Durante el período de autoajuste, el procedimiento se basa en el concepto de deshacer la modulación en los pulsos que transmiten la información. Es decir, el estimado \bar{a}_0 es cuantizado

al símbolo transmitido más próximo. Llamemos a este estimado cuantizado a_{c0} . Se forma una señal de error entre el estimado cuantizado y el no cuantizado: $e = a_{c0} - \bar{a}_0$. En otras palabras, el símbolo decodificado es asumido como el símbolo transmitido correcto. Esto es posible si las variaciones del canal son lentas en comparación a la tasa de muestreo $1/T$ y a bajas tasas de error, con lo que un error ocasional tendrá un efecto despreciable en el comportamiento del sistema (Proakis and Miller, obra citada).

3.4 Algoritmo para el Filtro TDL.

El algoritmo básicamente es la implementación de la ecuación 3.16 para actualizar los coeficientes del filtro.

Se inicializa reseteando los contenidos de los registros de desplazamiento $ye_r[i]$ y encerando los coeficientes $h_r[i]$ para i de 1 al número de coeficientes del filtro menos 1 ($ca-1$). A continuación se ingresa las secuencias de 256 datos al filtro en forma secuencial, es decir, un dato a la vez, el cual se va desplazando en los registros o memorias del filtro.

Seguidamente se procede a calcular la salida del ecualizador, la cual es un sumatorio, como se muestra en la fig. 16.

Luego se cuantiza la salida a_r lo que da ad_r , el valor deseado de a_r , el cual es comparado con la señal almacenada $b_r[i]$ para calcular el error de transmisión.

En el período de entrenamiento calculamos los coeficientes h_r del filtro usando el error entre la salida del ecualizador y la señal almacenada; en tanto que en el período de autoajuste el error se forma entre la salida del ecualizador y la señal cuantizada ad_r .

Finalmente se procede a desplazar los datos de modo que se pierda el más antiguo e ingrese uno nuevo.

Todo lo afirmado anteriormente se resume en el siguiente algoritmo, el cual está expresado en lenguaje basic, como es costumbre hacerlo en la literatura consultada.

```

for i=1 to ca-1           `Inicialización
ye_r[i]=0
h_r[i]=0
next i

for k=1 to 256           `Empiezan las secuencias de 256 bits.
ye_r[ca]=y_r[k]         `Entrada de datos
a_r= Σi=1ca h_r[i]*ye_r[i]   `Salida del ecualizador
decide(a_r,ad_r)         `Cuantización
err_t_a=err_t_a+abs{(b_r[k]-ad_r)/2} `Error de transmisión

En el entrenamiento e_r=a_r-b_r[k] `Error de ecualización
En el autoajuste e_r=a_r-ad_r

for i=1 to ca
h_r[i]=h_r[i]-mu*(ye_r[i]*e_r)   `Actualización de los taps
next i

for i=2 to ca           `Desplazamiento en los registros
ye_r[i-1]=ye_r[i]
next i

next k                   `FIN DEL ALGORITMO DEL FILTRO TDL.

```

CAPITULO IV _

CAPITULO IV.

ANALISIS DEL AUMENTO DE LA VELOCIDAD DE CONVERGENCIA.

A continuación consideramos la elección del parámetro μ . Se demostrará que la convergencia del algoritmo recursivo TDL está garantizada para todos los valores de μ que satisfacen la desigualdad $0 < \mu < 2 / \lambda_{\max}$, donde λ_{\max} es el mayor valor propio de la matriz de covarianza R.

Para el efecto usaremos la teoría del Algebra Matricial en lo que respecta a la teoría de valores propios, de diagonalización, y de formas cuadráticas, todo lo cual se resume en el Apéndice IV.

Al final del desarrollo se demostrará que el tiempo de convergencia está determinado por la relación del máximo eigenvalor dividido para el mínimo eigenvalor de la matriz obtenida por diagonalización a partir de la de covarianza.

De esta forma nos introduciremos en la teoría de los algoritmos ortogonales, los cuales determinan que el máximo eigenvalor sea igual al mínimo y a todos los demás eigenvalores, de modo que el tiempo de convergencia se acelera notablemente.

4.1 Error cuadrático de exceso debido al uso de estimados ruidosos del gradiente.

Al usar el valor estimado (3.16) en lugar del valor exacto del gradiente (3.12), se comete un error adicional al que produce el método de los mínimos cuadrados. Al mínimo error residual lo llamaremos ϵ_{\min} , y es el error que resulta de asumir el valor exacto del gradiente.

Al mínimo error residual lo podemos encontrar de la siguiente manera. Reescribiendo la ecuación (3.2)

$$\epsilon_{\min} = E | a(n) - C^{T\text{opt}} Y(n) |^2 \quad (4.1)$$

$$\text{con } e = a(n) - C^{T\text{opt}} Y(n) \quad (4.1.a)$$

$$\epsilon_{\min} = E | e |^2 = E | e (a(n) - C^{T\text{opt}} Y(n)) | \quad (4.2)$$

$$= E | e a(n) | - C^{T\text{opt}} E | e Y(n) | \quad 21$$

Y usando la ecuación (3.4) tenemos

$$\epsilon_{\min} = E | e a(n) | \quad (4.3)$$

$$= E | a^2(n) | - C^{T\text{opt}} E | a(n) Y(n) |$$

$$\epsilon_{\min} = E | a^2(n) | - C^{T\text{opt}} P \quad (4.4)$$

Ecuación que proporciona el error mínimo en el caso exacto.

En el caso de la aproximación del gradiente por un estimado ruidoso, podemos escribir el error de la siguiente manera²²:

$$e = a(n) - C^T Y(n) \quad (4.5)$$

$$= a(n) - C^{T\text{opt}} Y(n) - (C^T - C^{T\text{opt}}) Y(n) \quad (4.6)$$

²¹ En la anterior ecuación hemos usado la propiedad siguiente de la esperanza matemática respecto de las constantes multiplicativas:

$$E (c x(n)) = c E (x(n))$$

²² Friendlander, B. Adaptive Algorithms for Finite Impulse Response Filters. En la obra citada de Grant and Cowan.

Por lo que usando 4.1.a

$$\epsilon = E(e^2) = E(\epsilon_{\min}^2) + E[(C^T - C^{\text{Topt}})Y(n)Y^T(n)(C - C_{\text{opt}})] \quad (4.7) \\ + 2 E(\epsilon_{\min} Y^T(n) (C - C_{\text{opt}}))$$

y usando la propiedad de que el error de predicción es no correlacionado con los datos pasados, ecuación (3.4) tenemos:

$$\epsilon = \epsilon_{\min} + E[(C^T - C^{\text{Topt}})Y(n)Y^T(n)(C - C_{\text{opt}})] \quad (4.8)$$

$$\epsilon = \epsilon_{\min} + (C^T - C^{\text{Topt}})E(Y(n)Y^T(n))(C - C_{\text{opt}}) \quad (4.9)$$

$$\epsilon = \epsilon_{\min} + (C^T - C^{\text{Topt}})R(n)(C - C_{\text{opt}}) \quad (4.10)$$

$$\epsilon = \epsilon_{\min} + (C - C_{\text{opt}})^T R(n) (C - C_{\text{opt}}) \quad (4.11)$$

que es la ecuación buscada.

4.2 Propiedades de convergencia del algoritmo recursivo TDL.

Defínase ²³ V como la diferencia entre C y C_{opt}, luego (4.11) viene dada por:

$$\epsilon = \epsilon_{\min} + V^T R(n) V \quad (4.12)$$

La diferenciación de (4.12) da una relación útil para el gradiente, la cual usaremos posteriormente:

$$\nabla = 2 R V \quad (4.12b)$$

puesto que ϵ_{\min} es independiente de C.

²³ Widrow, et al. Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter. Proceedings of The IEEE, Aug. 1976.

La matriz de covarianza R , es una matriz cuadrática de energía, por lo que puede ser representada como (ver Apéndice IV, ec. A4.33):

$$R = QGQ^T \quad (4.13)$$

donde Q es la matriz diagonalizante ortogonal ($Q^{-1}=Q^T$) de que se habla en el Apéndice IV y G es la matriz diagonal $G=\text{diag}(\lambda_1, \dots, \lambda_n)$ de los eigenvalores de A . Del Apéndice IV sabemos que las dos matrices G y R , relacionadas a través de (4.13), representan la misma transformación lineal en distintas bases V' y V respectivamente, las cuales están relacionadas a través de la matriz ortogonal Q de la siguiente manera:

$$V' = Q^T V \quad (4.14)$$

Observación: para normalizar la notación usada en el apéndice, según la cual la expresión anterior sería $V' = VQ$, aclaremos que ello depende de la forma en que se define un vector. En el presente capítulo los vectores se definen como una matriz $n \times 1$, en tanto que en el apéndice como una matriz $1 \times n$. Por otro lado en el apéndice, la matriz Q se forma trasponiendo los vectores fila, en tanto que aquí se formará trasponiendo los vectores columna.

Por lo que la ecuación (4.12) puede ser reexpresada como:

$$\epsilon = \epsilon_{\text{min}} + V^T Q G Q^T V \quad (4.15)$$

de (4.14),

$$V'^T = V^T Q$$

con lo que de (4.15)

$$\epsilon = \epsilon_{\text{min}} + V'^T G V' \quad (4.16)$$

donde las coordenadas primas son los ejes principales de la superficie cuadrática.

La transformación (4.14) es aplicable a los taps:

$$C'(j) = Q^{-1}C(j) \quad (4.17)$$

(Del Apéndice IV sabemos para una matriz diagonal se cumple $Q^{-1} = Q^T$). En dicho apéndice se ve también que el diagonalizar una forma cuadrática elimina los términos en xy (si hablamos de los ejes coordenados x =abscisa, y =ordenada), es decir que anula la interrelación y facilita el tratamiento de la forma cuadrática (una elipse en el ejemplo del final del Apéndice IV). Esto se efectúa por intermedio de la diagonalización, ya que entonces desaparecen los términos a_{ij} , para $i \neq j$.

De esta forma, en la ecuación se puede analizar la convergencia de los taps C_i , que en este caso tienen asociada la coordenada V que desempeña el papel de la abscisa, con mucha facilidad, ya que al pasar a la nueva coordenada V' estamos en un sistema caracterizado por una matriz diagonal, lo que facilita el análisis de convergencia como veremos a continuación.

De la ecuación (3.11) sabemos que los taps tienen un cambio proporcional al gradiente

$$C_k(j+1) = C_k(j) - \mu/2 \partial \epsilon(j) / \partial C_{kP} \quad (3.11)$$

ó

$$C_k(j+1) = C_k(j) - \mu/2 \nabla(j) \quad (4.18)$$

con (4.12.b) obtenemos:

$$C_k(j+1) = C_k(j) - \mu R V \quad (4.19)$$

y con (4.13):

$$C_k^{(j+1)} = C_k^{(j)} - \mu QGQ^T V \quad (4.20)$$

restando a los dos miembros C_{opt} y multiplicando por Q^{-1} tenemos:

$$Q^{-1}(C_k^{(j+1)} - C_{opt}) = Q^{-1}(C_k^{(j)} - C_{opt}) - \mu GQ^T V \quad (4.21)$$

4.3 La Ecuación Homogénea Desacoplada que controla la Convergencia.

Usando la definición de V y la transformación $V' = Q^{-1}V$ tenemos, de (4.21),

$$V'(j+1) = V'(j) - \mu G V'(j) \quad (4.22)$$

$$V'(j+1) - (I - \mu G) V'(j) = 0 \quad (4.23)$$

La ecuación (4.23) es una ecuación de diferencias homogénea, la cual es desacoplada, es decir que el coeficiente $c_0'(j+1)$ solo depende del coeficiente $c_0'(j)$ y no de $c_m'(j)$, $m=1,2,\dots,n$.

Esta ecuación tiene una solución geométrica simple en las coordenadas primas:

$$V'(j) = (I - \mu G)^j V'(0) \quad (4.24)$$

donde $V'(0) = Q^{-1}(C(j) - C_{opt}) = C'(j) - C_{opt}'$ por (4.17).

Por la condición de desacoplamiento, cada tap con función $c(j)$ puede ser analizado por separado tomando el correspondiente eigenvalor de la matriz diagonal G , así, de (4.24):

$$c_k'(j) = (1 - \mu \lambda_j)^j c_k'(0) \quad (4.25)$$

por lo que el k-ésimo tap convergirá si:

$$(1 - \mu \lambda_j) < 1 \quad (4.26)$$

En general, los taps pueden ser positivos o negativos, por lo que la condición general es (abs = valor absoluto de)

$$\text{abs}(1 - \mu \lambda_j) < 1 \quad (4.27)$$

de donde

$$-1 < 1 - \mu \lambda_j < 1$$

restando 1 a cada término de la desigualdad

$$-2 < -\mu \lambda_j < 0$$

despejando μ

$$2/\lambda_j > \mu > 0$$

4.4 Condiciones que debe cumplir el coeficiente de convergencia μ .

Si en forma general usamos $\lambda_{\text{máx}}$ en lugar de λ_j , aseguramos que se cumpla la condición sobre μ con la siguiente desigualdad

$$2/\lambda_{\text{máx}} > \mu > 0 \quad (4.28)$$

LQGD.

donde $\lambda_{\text{máx}}$ es el máximo eigenvalor de la matriz de covarianza R.

Para analizar la convergencia del algoritmo, que es lo fundamental de este capítulo, aproximemos la ecuación de diferencias (4.23) por la ecuación diferencial:

$$\partial V' / \partial j + \mu G V' = 0 \quad (4.29)$$

o en forma equivalente,

$$\partial c' / \partial j + \mu \lambda_j c' = 0 \quad (4.30)$$

cuya solución es

$$c'(j) = \exp [(-\mu \lambda_j)j] = \exp (-j/\tau) \quad (4.31)$$

donde $\tau = 1 / \mu \lambda_j$, y la tasa con la que decrece $c'(j)$ es

$$c'(j+1)/c'(j) = \exp(-(j+1)/\tau)/\exp(-j/\tau) = \exp (-1/\tau) \quad (4.32)$$

que para el peor caso (respecto del tiempo de muestreo, que es conveniente definirlo como 1, como referencia), es decir grandes τ , (4.32) se puede aproximar por

$$c'(j+1)/c'(j) = \text{converge} = \exp (-1/\tau) \approx 1 - 1/\tau \quad (4.33)$$

y de (4.23)

$$\text{converge} = 1 - \mu \lambda_j = 1 - 1/\tau$$

por lo que el tiempo de convergencia τ , con la asunción hecha de grandes τ se convierte en

$$\tau = 1 / \mu \lambda_{\text{mín}} \quad (4.34)$$

con (4.28) llegamos a

$$\tau = \lambda_{\text{máx}} / 2\lambda_{\text{mín}} \quad (4.35)$$

por lo que hemos obtenido el siguiente resultado fundamental:

- ▶ La velocidad de convergencia del algoritmo TDL que usa el método
- ▶ del gradiente de descenso más pronunciado es directamente
- ▶ proporcional a la relación del máximo eigenvalor de la matriz de
- ▶ covarianza dividido para el mínimo eigenvalor. De esto se
- ▶ concluye que para mejorar la velocidad de convergencia del
- ▶ algoritmo se debe tratar de que todos los eigenvalores de la
- ▶ matriz R sean iguales, lo cual se puede lograr mediante la
- ▶ ortogonalización de las señales de entrada a los taps del
- ▶ ecualizador. 24

4.5 Aumento de la velocidad de convergencia del algoritmo mediante la ortogonalización de las señales de entrada al ecualizador.

Es pues esta la razón que determina que el Filtro de Malla, que será tratado en el capítulo V, presente mayor velocidad de convergencia que el Filtro TDL. En el Filtro de Malla se procede a mejorar la velocidad de convergencia del algoritmo de inversión de la matriz de covarianza, ortogonalizando la señal que entra al ecualizador, lo que como veremos posteriormente se puede lograr con el algoritmo del Filtro de Malla, ya que este algoritmo define al error reverso $r(n,i)$ como la combinación lineal de $y(n,i+1), \dots, y(n,i)$, tal que

$$E(r(n,i)r(m,i)) = \text{potencia de la señal para } m=n \\ = 0 \text{ para } m \neq n.$$

Lo que básicamente implica que la señal de error es una señal ortogonal formada a partir de la señal de entrada al ecualizador.

Este hecho fue fácilmente comprobado en la simulación, en

²⁴ Qureshi, S. Adaptive Equalization. Proceedings of the IEEE. Sept. 1985

donde se comprobó la ortogonalidad de las señales formando la suma de los productos una vez que convergió el algoritmo del Filtro de Malla y dividiendo para el número de iteraciones considerado. El resultado, trabajando con los datos de la matriz B (cuya obtención se detalla en la siguiente página) dio $E(r_1r_2) \approx .06 \approx 0$, y $E(r_1,r_1) \approx E(r_2,r_2) \approx 0.68 \approx 1$, lo que comprueba la ortogonalidad de las señales, así:

$$E(r_1r_2) = [1.015 * .717 + .956 * .94 - .489 * .83 + 1.277 * .533 + .655 * 1.227 - .4746 * .703 - .52 * .542 - .719 * .6 - .867 * .797 - .559 * .745] / 9 = .06$$

$$E(r_1r_1) = [1.014^2 + .957^2 + .489^2 + 1.277^2 + .655^2 + .475^2 + .52^2 + .719^2 + .867^2] / 9 = .67$$

$$E(r_2r_2) = [.717^2 + .94^2 + .83^2 + .533^2 + 1.23^2 + .7^2 + .542^2 + .6^2 + .797^2] / 9 = .69$$

Puesto que la matriz de covarianza se forma con el valor esperado $E(r(n,i)r(n,j))$, esta matriz tiene una matriz diagonal asociada (ver Apéndice IV) para la cual los eigenvalores de la diagonal principal son iguales entre sí e iguales a los valores de potencia de la señal.

La última afirmación se sustenta en la simulación realizada. Para efectos de demostrar que eigenvalores iguales determinan ortogonalidad en las señales de entrada, hemos usado el Programa realizado en Lenguaje C, el cual se lista en el Apéndice II.

Corrimos el programa con los siguientes datos:

```
semilla = 50;
velocidad de modulación = 1200 bits/s;
```



```

exceso de ancho de banda = .25;
portadora = 1800 Hz;
# coeficientes del Filtro de Malla = 2;

```

Hemos usado el procedimiento detallado en el Apéndice IV para hallar la matriz de covarianza R. En el caso presente, los vectores r[i] reemplazan a los datos de entrada y[i] usados en dicho apéndice. Estos vectores r[i] fueron obtenidos de la tercera secuencia de 256 bits (paso 3 de 4) para una corrida de 1024 bits. Específicamente a partir de b_r[166], donde este vector es el indicador de la posición en la secuencia de 256 bits usado en el programa y que se muestra en pantalla cuando dicho programa es corrido.

Los 9 vectores r[i] (con 8 se obtuvo el resultado deseado, pero se eligió 9 para mejorar aún más los resultados) para un Filtro de Malla de 2 coeficientes son:

B = [b1 b2 b3 b4 b5 b6 b7 b8 b9]

$$= \begin{vmatrix} 1.014 & .957 & -.489 & -1.277 & -.655 & .475 & -.52 & .719 & -.867 \\ .717 & .94 & .83 & -.533 & -1.23 & -.7 & .542 & -.6 & .797 \end{vmatrix}$$

donde b1 = $\begin{vmatrix} r[1,166] \\ r[2,166] \end{vmatrix}$,

b2 = $\begin{vmatrix} r[1,167] \\ r[2,167] \end{vmatrix}$, etc.

Es decir, hemos tomado los errores reversos de orden 1 y 2 para cada uno de los tiempos que van desde i=166 hasta i=174, en total 9 vectores.

A continuación hemos obtenido la matriz de covarianza R como

$$R = B B^T = \begin{vmatrix} 6 & .96 \\ .96 & 5.7 \end{vmatrix}$$

pero sabemos que la matriz R es un promedio de las muestras, y como hemos tomado 9, normalizando tenemos:

$$R = \begin{vmatrix} .7 & .1 \\ .1 & .63 \end{vmatrix}$$

- ▶ que es lo que sustenta la afirmación de que si uno ortogonaliza
- ▶ los datos de entrada, la matriz resultante presenta unos
- ▶ eigenvalores en su matriz diagonal tales que son aproximadamente
- ▶ iguales (.7 ≈ .63 ≈ 1; .1 ≈ 0), por lo que de (4.35) la
- ▶ convergencia es más rápida que en el caso del Filtro TDL.

4.6 Determinación del valor óptimo de μ .

A partir de (4.16), regresando a la notación de sumatorios, tenemos²⁵:

$$\epsilon = \epsilon_{\min} + \sum_{j=0}^n \lambda_j |C_j' - C_j'^{\text{opt}}|^2 \quad (4.36)$$

El incremento en el MSE (error cuadrático medio) debido a las fluctuaciones randómicas de los taps alrededor de sus valores óptimos está dada por el segundo término de (4.36). El valor promedio de este error residual, denotado como

²⁵ Proakis and Miller, An Adaptive Receiver for Digital Signaling Through Channels With Intersymbol Interference. En la obra de L. Franks: Baseband Communications, IEEE Press. 1975

$$\epsilon_r = \sum_{j=0}^n \lambda_j E|C_j' - C_j^{\text{opt}}|^2 \quad (4.37)$$

es definido como el "error medio cuadrático de exceso" que resulta del uso de estimados ruidosos en el algoritmo recursivo.

A continuación consideramos la determinación de ϵ_r , la cual se demostrará que viene dada por

$$\epsilon_r = \mu (n+1) \epsilon_{\text{min}} r_p/2 \quad (4.38)$$

donde r_p es el valor promedio de la energía en la diagonal de la matriz de covarianza R . Nótese que el ϵ_r se incrementa con el aumento del número de taps y de μ .

Para la simulación hemos encontrado que un valor que asegura adecuada convergencia del algoritmo es $\mu=0.01$, lo que puede ser deducido a partir de (4.38). Tomando μ de modo que ϵ_r es reducido a un valor muy pequeño comparado con ϵ_{min} , digamos la décima parte, se tiene

$$\mu = 0.2 / \{(n+1) r_p\} \quad (4.39)$$

Con $n=20$ coeficientes y el promedio de la energía r_p en la diagonal de la matriz de covarianza igual a 1 (para transmisión NRZ normalizada: +1,-1), se tiene $\mu = 0.01$.

4.7 Deducción de la expresión para el error residual ϵ_r .

En la presente sección obtendremos una expresión para $E|C_j' - C_j^{\text{opt}}|^2$, el cual es usado en (4.37).

Empecemos suponiendo que el tomar un estimado ruidoso para

y $E(e_j y_j) \approx e_j y_j$ determina que en (4.23) la ecuación homogénea tenga un término de ruido aditivo r^j que es causante de que C^j fluctúe en el estado estable alrededor de C^{opt} :

$$V^j(j+1) = (I - \mu G) V^j(j) + r^j \quad (4.40)$$

o en forma equivalente,

$$v^j(j+1) = (1 - \mu \lambda_j) v^j(j) + r^j \quad (4.41)$$

Asumiendo que el análisis se centra alrededor de r^j , de modo de que

$$E(v^j(j+1)) \approx E(v^j(j)) \quad (4.41.b)$$

Asumiendo también que el ruido r^j es blanco, e.d.:

$$\begin{aligned} E(r^j(k,j) r^j(l,j)) &= E(r^j(k,j))^2 \text{ si } k=l \\ &= 0 \text{ en otro caso.} \end{aligned} \quad (4.42)$$

Tomando el cuadrado de ambos miembros de (4.41)

$$v^j(j+1)^2 = (1 - \mu \lambda_j)^2 v^j(j)^2 + 2 (1 - \mu \lambda_j) v^j(j) r^j + r^j{}^2 \quad (4.43)$$

Aplicando el valor esperado a ambos miembros y tomando en cuenta que el ruido blanco es independiente de los coeficientes se llega a:

$$E(v^j(j+1)^2) = (1 - \mu \lambda_j)^2 E(v^j(j)^2) + E(r^j{}^2) \quad (4.44)$$

por lo que usando (4.41.b),

$$(1 - (1 - \mu \lambda_j)^2) E(v^j(j)^2) = E(r^j{}^2) \quad (4.45)$$

$$E(v'(j)^2) = E(cj' - cj'_{opt})^2 = E(r'j^2) / (1 - (1 - \mu\lambda_j)^2) \quad (4.46)$$

De lo cual vemos que hallar el valor esperado de $E(cj' - cj'_{opt})^2$ se ha reducido a hallar el valor esperado de $E(r'j^2)$.

Como hemos mencionado anteriormente, el ruido aditivo r_j deviene del uso de estimados ruidosos de $E(e_j y_j)$, lo que determina que los coeficientes C_j fluctúen alrededor de sus valores óptimos debido al ruido (de promedio cero = $E(e_j y_j)$):

$$r_j = \mu e_j y_j \quad (4.47)$$

que con la matriz de transformación de base, Q^T , da $r'j$

$$r'j = Q^T r_j \quad (4.48)$$

La matriz de covarianza de $r'j$, Gr' , es:

$$Gr' = E(r'j r'j^T) \quad (4.49)$$

$$Gr' = Q^{-1} E(r_j r_j^T) Q \quad (4.50)$$

de (4.47)

$$Gr' = \mu^2 Q^{-1} E(e_j^2 y_j y_j^T) Q \quad (4.51)$$

En este punto hacemos otra asunción concerniente a la estadística del ruido, intuitivamente parece razonable que conforme el número de muestras y_j aumenta, el error y el cuadrado del error tienden a ser no correlacionados con y_j y con los productos $y(k,j)y(l,j)$, lo que básicamente es demostrado en la simulación, en donde la expresión para μ funciona adecuadamente.

Por tanto:

$$E(e_j^2 y_j y_j^T) = E(e_j)^2 E(y_j y_j^T) = \epsilon_{\min} E(y_j y_j^T) \quad (4.53)$$

donde $E(e_j)^2 = \epsilon_{\min}$ es consistente con la suposición de que los taps casi han convergido a los valores óptimos de modo que el MSE es mínimo.

Esta asunción nos permite expresar la matriz de covarianza Gr' como

$$Gr' = \mu^2 \epsilon_{\min} Q^{-1} E(y_j y_j^T) Q \quad (4.54)$$

$$Gr' = \mu^2 \epsilon_{\min} Q^{-1} R Q \quad (4.55)$$

$$Gr' = \mu^2 \epsilon_{\min} G \quad (4.56)$$

donde hemos usado la definición de la matriz diagonal G semejante a R .

Por tanto

$$E(r'j^2) = \mu^2 \epsilon_{\min} \lambda_j, \quad j=0, \dots, n \quad (4.57)$$

La sustitución de este resultado en (4.46) da

$$E(v'(j)^2) = E(cj' - cj'_{\text{opt}})^2 = \mu^2 \epsilon_{\min} \lambda_j / (1 - (1 - \mu \lambda_j)^2) \quad (4.58)$$

para $j=0, \dots, n$.

Sustituyendo (4.58) en (4.37) da

$$\epsilon_r = \mu^2 \epsilon_{\min} \sum_{j=0}^n \lambda_j^2 / (1 - (1 - \mu \lambda_j)^2) \quad (4.59)$$

Para simplificar el resultado, hagamos la asunción de que

$\mu \lambda_j \ll 1$, de modo que (4.59) se convierte en

$$\epsilon_r = \frac{\mu}{2} \epsilon_{\min} \sum_{j=0}^n \lambda_j \quad (4.60)$$

Del Apéndice IV sabemos que la suma de los eigenvalores de la matriz diagonal es la traza de la matriz y que esta traza se caracteriza porque no es necesario obtener los eigenvalores, sino que también es igual a la suma de los elementos de la diagonal de la matriz de covarianza no diagonalizada.

$$\epsilon_r = \mu/2 \epsilon_{\min} \text{Tr } R \quad (4.61)$$

con $\text{Tr } R = R(0) = r_{00} + r_{11} + \dots + r_{nn}$

$$\epsilon_r = \mu/2 \epsilon_{\min} (n+1) r_p \quad (4.62)$$

donde $r_p = r$ promedio, que para transmisión NRZ es 1.

Que es el resultado mostrado en (4.38), LQD.

4.8 Análisis del Ruido Gaussiano.

En el análisis del capítulo 2 se mencionó que la incidencia del ruido en la señal puede ser analizado agregando un término de voltaje de ruido $\Omega(t)$ a la ecuación 2.6, que la repetimos a continuación:

$$y(t) = \sum_{i=-N}^N a_i r(t-iT) + \Omega(t) \quad (2.6)$$

Para la simulación nos interesa encontrar una expresión que relaciona la potencia del ruido Gaussiano σ^2 , con la relación energía por bitio transmitido a la densidad del ruido (E_b/N_o).

Sea E_b = energía por bit transmitido. A partir de la ecuación (A3.3.19) del apéndice 3, que repetimos a continuación, se establece que la densidad espectral del ruido blanco es:

$$G_n(f) = \text{constante} = N_o/2 \quad (A3.3.19)$$

En la práctica es más conveniente medir la potencia promedio del tren que la energía de los bits, sin embargo existe la siguiente relación entre potencia y energía de los bits.²⁸

$$E_b = P T = P (1/f) \quad (4.63)$$

Usando (A.3.3.18a),

$$N = \sigma^2 = N_o * \text{Ancho de banda del ruido} = N_o 2 f_n \quad (4.64)$$

Para la modulación BPSK, el ancho de banda del ruido es el correspondiente al de una señal de doble banda lateral puesto que la modulación duplica el ancho de banda, que en banda base es $B = f_n$.

$$E_b/N_o = P T/[N/(2f_n)] = P/f * 2 f_n/N = P/N \quad (4.65)$$

puesto que $f = 2 f_n$.

De donde resulta que el voltaje de ruido a ser introducido es:

$$\sigma = \text{sqrt}(N) = \text{sqrt} (P/(\text{SNR})) \quad (4.66)$$

²⁸ Feher, K. Digital Communications. Pg. 158.

en donde $SNR = E_b/N_0 =$ relación señal a ruido.

Puesto que trabajamos con señales de +1 y -1, la potencia promedio a través de una resistencia normalizada de un ohm es 1 watt. Por esta razón, a partir de (4.66) tenemos

$$\sigma = \text{sqrt}(N) = \text{sqrt} (1/(SNR)) \quad (4.67)$$

que es la ecuación usada en la simulación para introducir un voltaje de ruido σ .

Con el objeto de tener una referencia del SNR a usar, tenemos que Gitlin y Magee, en su documento "Self-Orthogonalizing Adaptive Equalization, IEEE Transactions On Communications, July 1977", establecen una relación de señal a ruido (SNR) de 31 dB, lo que determina, si usamos señales de 1mW de potencia, que el ruido sea del orden de 0.001 mW.

Honig, en "Echo Cancellation of Voiceband Data Signals", establece que en transmisión telefónica la SNR es del orden de 40 dB, lo que corresponde a un nivel de ruido de 0.0001 mW.

Y la recomendación M.1025 del CCITT, libro rojo, establece que el límite provisional para circuitos arrendados para distancias superiores a 10000 km es de -38dBmOp, que corresponde a 36 dB. Por esta razón, usaremos un valor de 36 dB en las corridas del programa.

Nota: La unidad dBmOp tiene que ver con el ruido sofométrico, el cual no es más que el ruido medido por un voltímetro rms que tiene un filtro ecualizador que sopesa las diferentes frecuencias presentes en el ruido de acuerdo a la sensibilidad del oído humano. Es decir, el oído no presenta igual sensibilidad para las diferentes frecuencias y es más sensible entre los 800 y 1000 Hz, presentando atenuación para frecuencias fuera de este rango. De esta manera, la forma de calibrar el aparato sofométrico consiste en enviar una frecuencia e ir aumentando la potencia de ésta hasta que un par de personas que se ocupan de la prueba,

encuentren que el efecto de molestia a alcanzado un cierto nivel predeterminado.

A continuación se envía una frecuencia de 1000 Hz y se mide la potencia necesaria para generar el mismo nivel de ruido que con la frecuencia enviada anteriormente.

Obviamente, dada la baja sensibilidad del oído a frecuencias muy graves o muy agudas, los niveles de potencia requeridos por la frecuencia de 1000 Hz serán menores que los correspondientes a las frecuencias mencionadas.

A continuación se configura el filtro ecualizador de tal forma que atenúe las diferentes frecuencias presentes en el ruido de la misma manera que lo hace el oído humano.

Agreguemos finalmente que la relación entre dBmOp y dB es:
 $\text{dBmOp} + 90.8 = 88.3 + \text{dB}$.

Esto resulta del hecho de que la medida en dB no considera la ponderación sofométrica y es por tanto 2.5 dB mayor que con la ponderación.

CAPITULO V.

CAPITULO V.

El Filtro de Malla.

En este capítulo analizaremos el Filtro de Malla, motivados entre otras cosas por las buenas propiedades presentadas por su contraparte analógica.

Por ejemplo, el filtro análogo de la fig.17. presenta una relativa insensitividad de la respuesta de frecuencia frente a ligeras perturbaciones de los elementos circuitales alrededor de sus valores nominales.

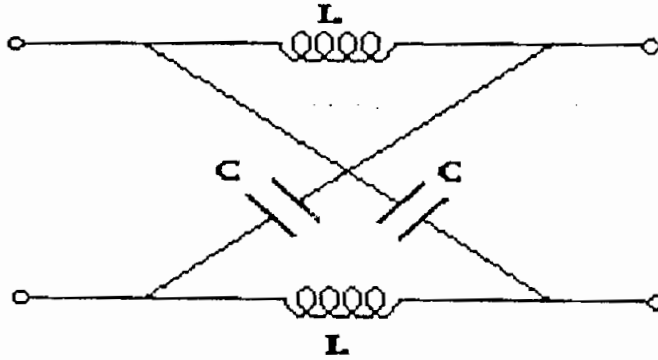


Figura 17 Implementación analógica del Filtro de Malla.

Desarrollando una configuración de filtro digital que es semejante a la estructura análoga de malla, el filtro digital hereda muchas de las mismas propiedades²⁷.

En la misma referencia anterior se establece que la función

²⁷ Turner. Recursive Least-Squares and Lattice Filter. en la obra citada de Cowan and Grant.

de transferencia de la etapa analógica de la fig. 17. presenta polos y ceros simultáneamente; sin embargo, esta forma simétrica puede ser modificada en el caso digital a una forma asimétrica con el objeto de tener el mínimo número de multiplicadores y se la muestra en la parte (b) de la fig. 18.

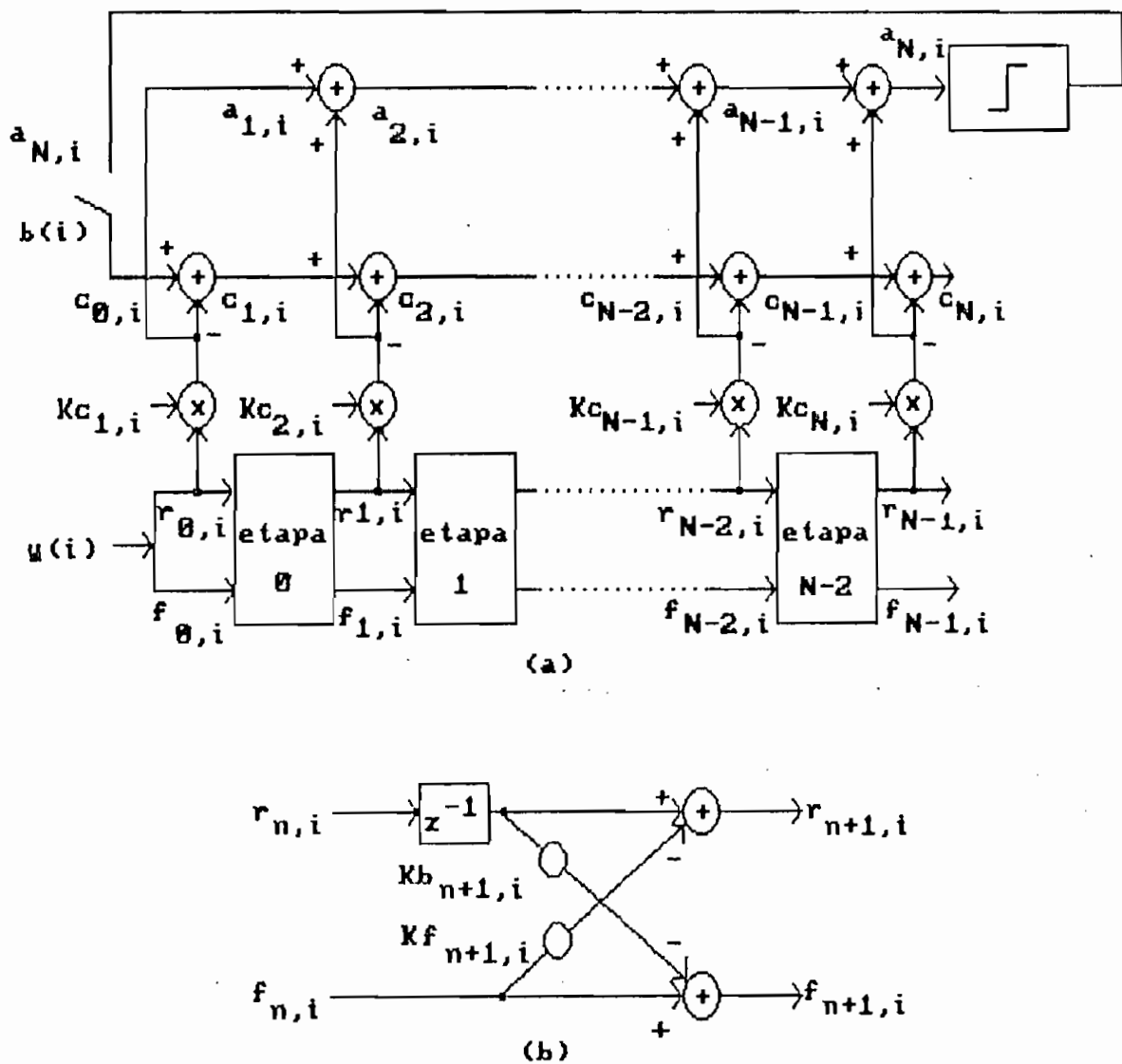


Figura 18 El Filtro de Malla: (a) Estructura general, (b) Etapa n-ésima.

5.1 Propagación de señales en el filtro.

En la figura 18.b. se pueden observar que se propagan 2 señales, $f(t)$ y $r(t)$, a las cuales posteriormente llamaremos error directo y reverso.

Estas señales interactúan entre sí de la siguiente manera, la cual puede ser deducida prestamente de la misma figura:

$$f(n+1,t) = f(n,t) - k(n+1) r(n,t-1) \quad (5.1)$$

$$r(n+1,t) = r(n,t-1) - k(n+1) f(n,t) \quad (5.2)$$

En este par de ecuaciones, la dependencia de f y r respecto del orden n y del tiempo t se expresa como $f(n,t)$ y $r(n,t)$.

Los coeficiente $k(n)$ son conocidos como coeficientes de reflexión²⁸.

5.2 Propiedades de ortogonalización del Filtro de Malla.

El problema de ecualización que estamos tratando puede ser visualizado como una regresión lineal múltiple en donde los coeficientes se ajustan usando el criterio del mínimo error cuadrático.

²⁸ Los coeficientes k deben su nombre al hecho de que el modelo a partir del cual pueden deducirse "físicamente" es un modelo de la cavidad bucal (Turner, ob.cit.). Este modelo considera dicha cavidad como formada de secciones cilíndricas de igual longitud pero diferente diámetro. A través de estas secciones se propaga el sonido, el cual posee una velocidad v y una onda de presión sonora p . Como se demuestra en Turner, la transmisión del sonido obedece a la ecuación de onda, la cual determina que en los extremos de las secciones de diferente área exista una reflexión de la onda sonora debido al cambio abrupto de medio. En Turner se encuentra que la reflexión de la onda viene dada por una relación de áreas y esta relación es la que define a los coeficientes k , que por esta razón se denominan coeficientes de reflexión. En la contraparte eléctrica no se habla de relación de áreas sino de relación de energías de error.

La regresión lineal se usa en todos los casos en donde se tiene un número grande de observaciones y en los cuales se quiere predecir el valor que adquirirá una cierta variable en función de otras.

Básicamente entonces, lo que se tiene es un sistema de ecuaciones con más datos que incógnitas, el cual es irresoluble en forma directa, y es por ello que se busca el mejor estimado de la variable que nos interesa.

Lo que hemos afirmado se puede establecer matemáticamente diciendo que la muestra recibida $x(t)$ puede ser aproximada por un sumatorio de las muestras anteriores, en donde existirá un cierto error que llamaremos $f(n,t)$.

El parámetro n es necesario puesto que denota el orden o número de términos que consideraremos en la aproximación:

$$f(n,t) = y(t) + a_1 y(t-1) + \dots + a_n y(t-n) \quad (5.3)$$

Recordemos del capítulo 3, ec (3.4), que el error es ortogonal a la señal de entrada cuando se aplica el método de los mínimos cuadrados para hallar los coeficientes del filtro, en este caso los a_i . Para el caso del filtro de malla y el error directo f , tenemos:

$$E(f(n,t) y(t-j)) = 0, \quad 1 \leq j \leq n \quad (5.4)$$

Además, en el caso del filtro de Malla es necesario definir un error inverso r , el cual predice $y(t-n-1)$ a partir de las mismas muestras $y(t-1), \dots, y(t-n)$, donde $y(t-n-1)$ es el valor de $y(t)$ que abandona el filtro, ya que este sólo tiene n taps.

$$r(n,t-1) = y(t-n-1) + c_1 y(t-n) + \dots + c_n y(t-1) \quad (5.5)$$

con

$$E(r(n,t-1) y(t-j)) = 0, \quad 1 \leq j \leq n+1 \quad (5.6)$$

Por qué es que se precisa de este error reverso se explica seguidamente.

Incrementando el orden de predicción a $n+1$, $f(n+1,t)$ no puede ser predicho sino con el auxilio de $y(t-n-1)$, el cual ya salió del filtro.

Sin embargo, la mayor parte de la información de $y(t-n-1)$ está contenida en $y(t-1)$, ..., $y(t-n)$. El error reverso representa entonces la información en la muestra $y(t-n-1)$, como se puede ver de (5.5).

Se ve de esta manera que el aumentar el orden del filtro implica el cálculo de f y de r , lo cual se puede efectuar recursivamente mediante (5.1) y (5.2) que las repetimos a continuación para el caso general, en donde los coeficientes de reflexión son distintos.

$$f(n+1,t) = f(n,t) - kf(n+1) r(n,t-1) \quad (5.7)$$

$$r(n+1,t) = r(n,t-1) - kb(n+1) f(n,t) \quad (5.8)$$

Las ecuaciones para el cálculo de los coeficientes de reflexión, conjuntamente con las 2 anteriores (lo que conducirá a (5.16) y (5.17)), explican la propiedad de ortogonalización del Filtro de Malla que detallaremos posteriormente.

El cálculo procede de manera similar a todas las determinaciones de coeficientes usando el criterio LMS (least mean squares).

5.3 Cálculo de los coeficientes de reflexión.

Tenemos que el error de orden $n+1$ es ortogonal a las

muestras:

$$E(f(n+1,t) y(t-j)) = 0, \quad 1 \leq j \leq n+1 \quad (5.9)$$

donde sabemos que la condición anterior aplicada a $y(t-n-1)$ no es satisfecha directamente por $f(n+1,t)$ (debido a que se precisa de $y(t-n-1)$), sino en forma indirecta usando el error reverseo y la ecuación (5.7), por tanto:

$$E(f(n+1,t) y(t-n-1)) = 0 \quad (5.10)$$

$$0 = E(f(n,t)y(t-n-1)) - kf(n+1) E(r(n,t-1)y(t-n-1)) \quad (5.11)$$

Para j entre 1 y n , se cumple (5.4), por lo que (5.11) no se altera si la reescribimos:

$$0 = E(f(n,t)r(n,t-1)) - kf(n+1) E(r(n,t-1)r(n,t-1)) \quad (5.12)$$

de donde:

$$kf(n+1) = E(f(n,t)r(n,t-1)) / E(r(n,t-1))^2 \quad (5.13)$$

Y a partir de

$$E(r(n+1,t) y(t-n-1)) = 0 \quad (5.14)$$

con un procedimiento similar al anterior se llega a:

$$kb(n+1) = E(f(n,t)r(n,t-1)) / E(f(n,t))^2 \quad (5.15)$$

Juntando las ecuaciones (5.7) y (5.13) tenemos las actualizaciones de f y r :

$$f(n+1,t) = f(n,t) - E(f(n,t)r(n,t-1)) / E(r(n,t-1))^2 r(n,t-1) \quad (5.16)$$

$$r(n+1,t)=r(n,t-1)-E(f(n,t)r(n,t-1)) / E(f(n,t))^2 f(n,t) \quad (5.17)$$

que nos dicen que los errores directo e inverso se obtienen a partir de un proceso de ortogonalización de Gram-Schmidt sobre las muestras de entrada al ecualizador.

Para explicar el párrafo anterior introduzcamos las ecuaciones del proceso de Gram-Schmidt.

5.4 El proceso de ortogonalización de Gram-Schmidt.

Teorema de Ortogonalización: Sea x_1, x_2, \dots, x_k una sucesión finita de elementos de un espacio euclídeo (que tiene producto interno) V . Existe una sucesión correspondiente de elementos y_1, y_2, \dots, y_k de V que tiene la propiedad de que el elemento y_k es ortogonal a todo elemento del subespacio $L(y_1, y_2, \dots, y_{k-1})$.²⁹

Demostración: Construyamos los elementos y_1, y_2, \dots, y_k por inducción. Para iniciar el proceso tomamos $y_1=x_1$. Supongamos que hemos construido y_1, \dots, y_r . Definamos y_{r+1} :

$$y_{r+1}=x_{r+1} - \sum_{i=1}^r a_i y_i \quad (5.18)$$

donde los escalares a_i deben determinarse. Para $j \leq r$, el producto interno de y_{r+1} con y_j viene dado por

$$(y_{r+1}, y_j) = (x_{r+1}, y_j) - \sum_{i=1}^r a_i (y_i, y_j) = (x_{r+1}, y_j) - a_j (y_j, y_j) \quad (5.19)$$

puesto que $(y_i, y_j)=0$ si $i \neq j$.

²⁹ Apostol, Calculus vol.2. Ed. Reverté 1980.

Si y_j es no nulo, podemos hacer y_{r+1} ortogonal a y_j tomando

$$a_j = (x_{r+1}, y_j) / (y_j, y_j). \quad (5.20)$$

Ej:

Sean $x_1 = (1, 1, 1)$, $x_2 = (-1, 0, -1)$ y $x_3 = (-1, 2, 3)$ la base S de V .
Hallar un conjunto $T = (y_1, y_2, y_3)$ que forme una base ortogonal de V .

Nota: Un conjunto de vectores $S = (x_1, x_2, \dots, x_k)$ de un espacio vectorial V se denomina base de V si S genera a V y es linealmente independiente.

Por ej., x_1, x_2, x_3 generan cualquier vector de 3 dimensiones. Para demostrar ello tomemos un vector en 3 d cualquiera, (a, b, c) y debemos hallar 3 constantes a_1, a_2, a_3 tales que: $(a, b, c) = a_1(1, 1, 1) + a_2(-1, 0, -1) + a_3(-1, 2, 3)$, lo que conlleva que:

$$a_1 - a_2 - a_3 = a$$

$$a_1 + 0 - 2a_3 = b$$

$$a_1 - a_2 + 3a_3 = c, \text{ que resolviendo da:}$$

$$a_1 = b - c/2 + a/2$$

$$a_2 = b - a/4 - 3c/4$$

$$a_3 = c/4 - a/4$$

Es claro que dados a, b, c , siempre se pueden formar a_1, a_2, a_3 , por lo que se concluye que la base S genera a V .

Para demostrar que S es linealmente independiente formamos:
 $a_1(1, 1, 1) + a_2(-1, 0, -1) + a_3(-1, 2, 3) = 0$, lo que es cierto si:

$$a_1 - a_2 - a_3 = 0$$

$$a_1 + 0 - 2a_3 = 0$$

$$a_1 - a_2 + 3a_3 = 0.$$

La única solución de este sistema de ecuaciones es: $a_1=a_2=a_3=0$, lo cual implica que S es linealmente independiente. Luego S es una base de V .

Regresando a obtener la base ortogonal, aplicamos el proceso de Gram-Schmidt a los vectores y_1, y_2, y_3 .

$$\text{Sea } y_1=x_1=(1,1,1)$$

$$y_2=x_2-(x_2,y_1)/(y_1,y_1) \quad y_1 = (-1,0,-1)+2/3 (1,1,1)=(-1/3,2/3,-1/3)$$

$$y_3=x_3-(x_3,y_1)/(y_1,y_1) y_1-(x_3,y_2)/(y_2,y_2) \quad y_2 = (-1,2,3) - 4/3 (1,1,1) - 2/3 \cdot 9/6 (-1/3,2/3,-1/3) = (-2,0,2)$$

Si eliminamos las fracciones de cada y_i , el conjunto resultante es también una base ortogonal (puesto que en la deducción igualamos (5.19) a 0):

$T=\{ (1,1,1), (-1,2,-1), (-2,0,2) \}$ es la base buscada, la cual se puede verificar fácilmente que es ortogonal.

Regresando al Filtro de Malla, mediante la simulación (ver acápite 4.5) hemos demostrado

$$E(r(n,t-1)r(j,t-1)) = \begin{cases} s & j=n \\ 0 & 1 \leq j < n \end{cases} \quad (5.21)$$

lo cual expresa que la señal de error r es una señal ortogonal, ya que su producto interno es diferente de cero sólo cuando se efectúa el producto consigo misma y es cero cuando se lo efectúa con otras señales.

Igual sucede con el error directo, para el cual

$$E(f(n,t)f(j,t)) = \begin{cases} s & j=n \\ 0 & 1 \leq j < n \end{cases} \quad (5.22)$$

A partir de (5.21) y de (5.5) que la reproducimos a continuación

$$r(n,t-1) = y(t-n-1) + c_1 y(t-n) + \dots + c_n y(t-1) \quad (5.5)$$

tenemos

$$r(n,t-1) = y(t-n-1) + \sum_{j=1}^n c_j y(t-j) \quad (5.23)$$

ecuación que la reconocemos como muy similar a (5.18), por lo que concluimos que la señal de error reverso proviene de un proceso de ortogonalización de Gram-Schmidt y que c_i viene dada por:

$$c_i = E(y(t)r(n,t-1)) / E(r(n,t-1)r(n,t-1))$$

que la podemos reescribir como

$$c_i = E(f(n,t)r(n,t-1)) / E(r(n,t-1)r(n,t-1)) \quad (5.24)$$

debido a (5.6).

- ▶ La última ecuación la reconocemos como el coeficiente de reflexión directo $k_f(n+1)$, habiendo de esta forma ligado el
- ▶ proceso de actualización de los coeficientes de reflexión, así
- ▶ como la de los errores directo y reverso a un proceso de
- ▶ ortogonalización de Gram-Schmidt.

Finalmente remarquemos el hecho de que se usará el error reverso y no el directo para la obtención de la salida del ecualizador debido a que el error directo involucra a la señal $y(t)$, la cual es necesario nuevamente tomar en cuenta en el algoritmo de malla, de modo que sería considerada duplicadamente.

El error inverso no contempla $y(t)$, de modo que es ideal para la aplicación.

Agreguemos que los errores en la etapa m no son función de los coeficientes de reflexión en etapas subsiguientes, de modo de que el filtro es totalmente recursivo en el orden.

CAPITULO VI

CAPITULO VI .

LA PREDICCIÓN LINEAL Y LAS ECUACIONES DE YULE-WALKER.

En este capítulo usaremos la teoría de la Predicción Lineal para la deducción del algoritmo del Filtro de Malla, el cual es más largo que el correspondiente al Filtro TDL.

La metodología a seguir es la siguiente:

- Primero consideremos que en el análisis a seguir existen dos líneas bien definidas.

La una tiene que ver con la predicción lineal que proporciona el valor de la muestra $y(t)$ basándose en los valores anteriores de dicha muestra.

Esto nos conducirá al establecimiento de las ecuaciones de Yule-Walker, que es una forma equivalente del teorema de ortogonalidad entre el error y los datos.

Estas ecuaciones nos permitirán obtener en forma eficiente la inversión de la matriz de covarianza, al hacer uso de dos propiedades intrínsecas de dicha matriz, tanto en el orden como en el tiempo.

Puesto que invertir la matriz es equivalente a actualizar los coeficientes implicados en la definición de error directo e inverso, lo primero será tratar de obtener tales actualizaciones.

La otra línea de análisis tiene que ver con el uso de la configuración del Filtro de Malla para ecualización, lo que conduce a obtener el mejor estimado de la salida del ecualizador, no precisamente mediante predicción lineal sino con un procedimiento modificado.

Este procedimiento contempla no sólo las muestras anteriores para hallar la actual sino también considera la entrada presente al filtro, de modo que aquí no hablaremos de las ecuaciones de Yule-Walker sino de los denominados procesos conjuntos.

De esta forma podemos visualizar al Filtro de Malla como de dos entidades independientes. Una que nos permite ortogonalizar las muestras de la señal de entrada, y otra que se reduce al filtro TDL usado anteriormente pero con entradas ortogonales.

Esto se ve de la figura 18., que la repetimos a continuación por facilidad de referencia.

En ella llamamos $k_c(n)$ a los coeficientes de la parte "tipo TDL", y hemos dibujado en bloques a la parte que configura propiamente al Filtro de Malla.

En esta figura podemos apreciar entre otras cosas el lazo de realimentación que será usado en el período de autoajuste, el cual precisa de un elemento de decisión para cuantizar la salida del ecualizador.

Vemos que los bits recibidos alimentan a la parte del Filtro de Malla que denominaremos "parte tipo predicción lineal", en tanto que los bits almacenados (en el período de entrenamiento) ó los bits cuantizados (en el período de autoajuste) alimentan a la "parte tipo TDL".

Podemos ver además que cada etapa del Filtro de Malla presenta un elemento de retardo sólo en el camino del error reverso, lo cual constituye la asimetría de la cual se habló al principio del capítulo 5, que determina un mínimo número de multiplicaciones.

El resto de señales las analizaremos en el numeral 6.6.

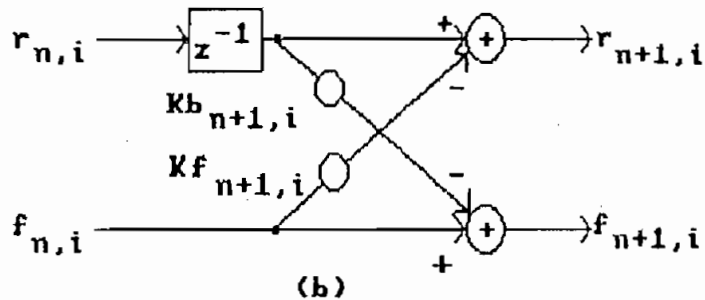
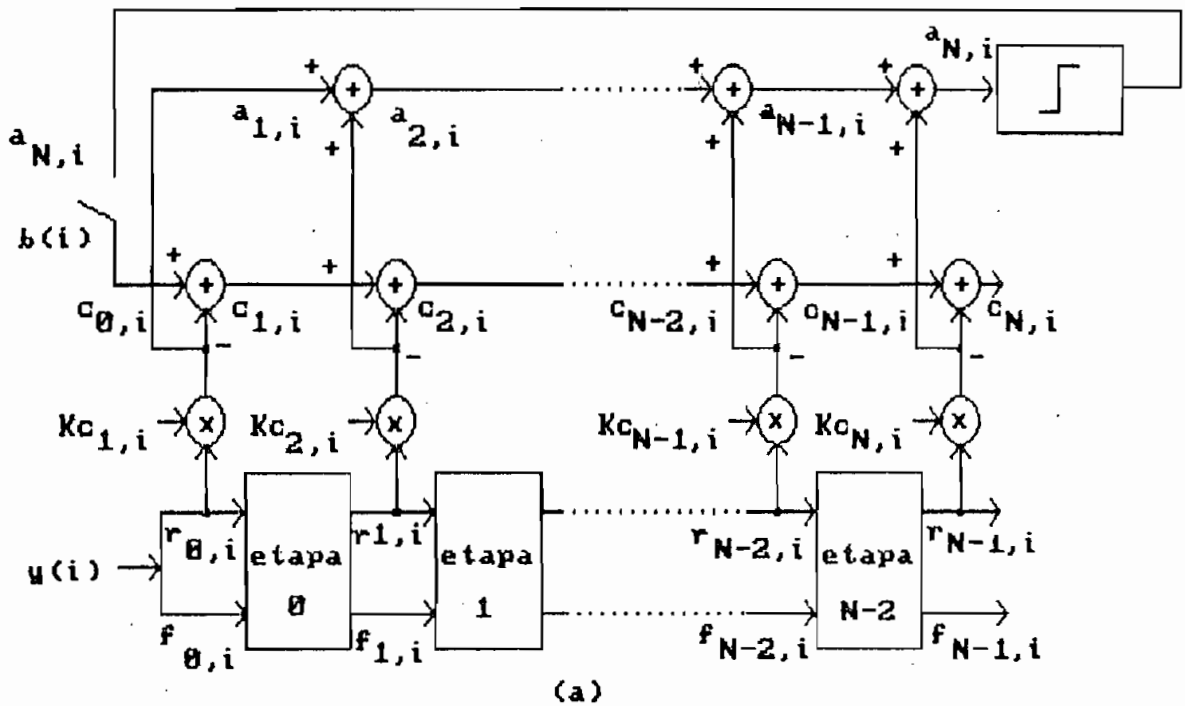


Figura 18 El Filtro de Malla: (a) Estructura general, (b) Etapa n-ésima.

- Establecido el problema como la inversión de la matriz de covarianza usando las ecuaciones de Yule-Walker, veremos el procedimiento de Levinson-Durbin, el cual permite efectuar la actualización de los coeficientes del error directo e inverso usando las propiedades de recursión en el orden y en el tiempo que presenta la matriz de covarianza.

Una vez que lleguemos a este punto introduciremos una matriz D que toma en cuenta las nuevas muestras introducidas.

Demostraremos que esta matriz D proviene de aplicar el lema de inversión de matrices a la matriz que actualiza la matriz de covarianza.

Finalmente procederemos a tratar la parte tipo TDL, para ver la mejor manera de obtener la salida del ecualizador, al tiempo que analizaremos la obtención del error en el período de autoajuste.

6.1 Las Ecuaciones de Yule-Walker:

Asumamos que $y(i)$ proviene de un proceso de autoregresión de orden n .

Si deseamos predecir $y(i)$ en base a las n muestras previas³⁰

$$\hat{y}(i) = -a_1y(i-1) - \dots - a_ny(i-n) \quad (6.1)$$

debemos calcular la correlación del error con $y(i)$ y con las muestras previas, y por el principio de ortogonalidad:

$$E[(y(i)-\hat{y}(i))y(i-j)] = 0 \quad 1 \leq j \leq n \quad (6.2)$$

Efectuando un cambio de variables, $k=i-j$, tenemos:

$$E[(y(k+j)-\hat{y}(k+j))y(k)] = 0 \quad , \quad 1 \leq j \leq n \quad (6.3)$$

$$E(y(k+j)y(k)) - E(\hat{y}(k+j)y(k)) = 0, \quad 1 \leq j \leq n \quad (6.4)$$

³⁰ Kay and Marple, Spectrum Analysis -A Modern Perspective. Proceedings of the IEEE, Nov. 1981.

Definiendo la covarianza $R(j) = E(y(k+j)y(k))$, tenemos (usando [6.1]):

$$R(j) = -E\left(\sum_{l=1}^n a_l y(k+j-l)y(k)\right) = -\sum_{l=1}^n a_l R(j-l) \quad 1 \leq j \leq n \quad (6.5)$$

Adicionalmente necesitamos el valor que la covarianza adquiere para $j=0$,

$$R(j) = s - \sum_{l=1}^n a_l R(-l), \quad j=0 \quad (6.6)$$

donde s es la potencia del elemento "y" cuando la covarianza o correlación se realiza para el elemento "y" consigo mismo ($j=0$).

De esta forma tenemos las ecuaciones de Yule-Walker, las cuales las juntamos y las mostramos a continuación.

$$R(j) = \begin{cases} -\sum_{l=1}^n a_l R(j-l) & 1 \leq j \leq n \\ -\sum_{l=1}^n a_l R(-l) + s & j=0 \end{cases} \quad (6.7)$$

Ecuaciones que las escribimos en detalle a continuación para un orden $n=3$:

$$\begin{aligned} R(0) &= -a_1R(-1) - a_2R(-2) - a_3R(-3) + s \\ R(1) &= -a_1R(0) - a_2R(-1) - a_3R(-2) \\ R(2) &= -a_1R(1) - a_2R(0) - a_3R(-1) \\ R(3) &= -a_1R(2) - a_2R(1) - a_3R(0) \end{aligned}$$

Las cuales en forma matricial dan:

$$\begin{bmatrix} R(0) & R(-1) & R(-2) & R(-3) \\ R(1) & R(0) & R(-1) & R(-2) \\ R(2) & R(1) & R(0) & R(-1) \\ R(3) & R(2) & R(1) & R(0) \end{bmatrix} * \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} s \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.8)$$

Debe notarse que en lo posterior usaremos la forma preventaneada de (6.8), es decir asumiremos que antes de la transmisión, todos los datos son cero, con lo que la matriz R resulta:

$$\begin{bmatrix} R(0) & 0 & 0 & 0 \\ R(1) & R(0) & 0 & 0 \\ R(2) & R(1) & R(0) & 0 \\ R(3) & R(2) & R(1) & R(0) \end{bmatrix} * \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} s \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.9)$$

que en forma general da:

$$R(n,i) * A(n,i) = \begin{bmatrix} s(n,i) \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (6.10)$$

Recordemos que la matriz R se puede obtener a partir de:

$$R = E |Y(i) Y^T(i)|$$

donde $Y^T(i) = [y(i), y(i-1), \dots, y(i-n+1)]$

es el vector de las muestras de entrada $y(nT)$.

A partir de estas definiciones, y de que la esperanza matemática puede aproximarse por el promedio para un número

suficiente de muestras³¹, tenemos que (temporalmente llamaremos A a la matriz de covarianza):

$$A(n, i+1) = 1/(i+1) \sum_{j=1}^{i+1} Y(j)Y^T(j) \quad (6.11)$$

de donde

$$A(n, i+1) = 1/(i+1) \sum_{j=1}^i Y(j)Y^T(j) + 1/(i+1) Y(i+1)Y^T(i+1) \quad (6.12)$$

con

$$A(n, i) = 1/(i) \sum_{j=1}^i Y(j)Y^T(j) \quad (6.13)$$

se tiene

$$A(n, i+1) = i/(i+1) A(n, i) + Y(i+1)Y^T(i+1) \quad (6.14)$$

con un cambio de variables, $R(n, i+1) = (i+1) A(n, i+1)$

$$R(n, i+1) = R(n, i) + Y(i+1)Y^T(i+1) \quad (6.15)$$

Con lo cual hemos obtenido la primera de las dos propiedades estructurales de la matriz de covarianza que son explotadas con el objeto de obtener las actualizaciones, del tiempo en este caso.

³¹ Gittlin and Magee: Self-Orthogonalizing Adaptive Equalization Algorithms. IEEE Transactions on Communications, July 1977.

Para la actualización del orden se hace uso de la propiedad de que la matriz de covarianza de orden $n+1$ contiene a la matriz de covarianza de orden n .

Para la demostración he tomado las matrices de covarianza de orden 1 y 2 al tiempo 3. He desarrollado paso a paso las matrices porque creo que es esta la única forma de verdaderamente comprender lo que sucede.

$$\text{Partiendo de } R(n,i) = \sum_{j=1}^i Y_j Y_j^T \quad (6.15.a)$$

con $Y_j^T = [y_j, y_{j-1}, \dots, y_{j-n}]$

tenemos:

$$R(1,3) = \sum_{j=1}^3 Y_j Y_j^T = Y_1 Y_1^T + Y_2 Y_2^T + Y_3 Y_3^T =$$

$$= \begin{vmatrix} y_1 \\ y_0 \end{vmatrix} (y_1 \ y_0) + \begin{vmatrix} y_2 \\ y_1 \end{vmatrix} (y_2 \ y_1) + \begin{vmatrix} y_3 \\ y_2 \end{vmatrix} (y_3 \ y_2) \quad (6.15.a1)$$

$$= \begin{vmatrix} y_1 y_1 + y_2 y_2 + y_3 y_3 & y_1 y_0 + y_2 y_1 + y_3 y_2 \\ y_0 y_1 + y_1 y_2 + y_2 y_3 & y_0 y_0 + y_1 y_1 + y_2 y_2 \end{vmatrix}$$

y,

$$R(3,3) = \sum_{j=1}^3 Y_j Y_j^T = Y_1 Y_1^T + Y_2 Y_2^T + Y_3 Y_3^T =$$

$$= \begin{vmatrix} y_1 \\ y_0 \\ y_{-1} \end{vmatrix} (y_1 \ y_0 \ y_{-1}) + \begin{vmatrix} y_2 \\ y_1 \\ y_0 \end{vmatrix} (y_2 \ y_1 \ y_0) + \begin{vmatrix} y_3 \\ y_2 \\ y_1 \end{vmatrix} (y_3 \ y_2 \ y_1)$$

$$= \begin{vmatrix} y_1y_1+y_2y_2+y_3y_3 & y_1y_0+y_2y_1+y_3y_2 & y_1y_{-1}+y_2y_0+y_3y_1 \\ y_0y_1+y_1y_2+y_2y_3 & y_0y_0+y_1y_1+y_2y_2 & y_0y_{-1}+y_1y_0+y_2y_1 \\ y_{-1}y_1+y_0y_2+y_1y_3 & y_{-1}y_0+y_0y_1+y_1y_2 & y_{-1}y_{-1}+y_0y_0+y_1y_1 \end{vmatrix} \quad (6.15b)$$

Se ve claramente de estas últimas líneas que la matriz de covarianza de orden 2 contiene como submatriz a la matriz de orden 1. Esto lo establecemos de la siguiente manera:

La segunda propiedad estructural de la matriz de covarianza se refiere a la actualización en el orden y establece:

$$R(n+1,i) = \begin{vmatrix} R(n,i) & * \\ * & * \end{vmatrix} = \begin{vmatrix} * & * \\ * & R(n,i-1) \end{vmatrix} \quad (6.16)$$

Donde los asteriscos denotan elementos que no son usados en la deducción del algoritmo.

La tercera parte de (6.16) es fácil de notar de (6.15b), ya que si $i=1$, $y_1y_1+y_2y_2+y_3y_3$, por lo que al tiempo $i=0$ tenemos $y_0y_0+y_1y_1+y_2y_2$.

A continuación se hará uso de los vectores de coeficientes de los errores directo y reverso A y C, donde:

$$A(n,i)^T = [1 \ a(1,i) \dots a(n,i)] \quad (6.17)$$

$$C(n,i)^T = [c(n,i) \dots c(1,i) \ 1] \quad (6.18)$$

Los cuales son los coeficientes cuya actualización significa llevar a cabo la inversión de la matriz de covarianza.

Estos coeficientes son parte del Filtro de Malla propiamente dicho, y como tales, parte de un proceso de autoregresión dado por las ecuaciones de Yule-Walker (6.7), así:

$$R(n,i) [A(n,i) \ C(n,i) \ D(n,i)] = \begin{vmatrix} sf(n,i) & 0 & y(i) \\ 0 & 0 & y(i-1) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 & sb(n,i) & y(i-n+1) \end{vmatrix} \quad (6.19)$$

Donde sf y sb son las energías del error directo y reverso, y el vector D toma en cuenta las nuevas muestras.

6.2 El vector D ó coeficiente de Kalman y su escalar asociado, γ .

Puesto que D es crucial en el desarrollo posterior, considero que es importante efectuar su deducción basado en los lineamientos dados en la referencia³².

D es llamado también coeficiente de Kalman puesto que apareció por primera vez en la deducción del algoritmo de Kalman.

Definimos la matriz inversa de la matriz de covarianza como:

$$P(n,i) = R(n,i)^{-1}, \quad (6.20)$$

Usaremos el lema de la inversión de una matriz, el cual es³³:

$$[A+BCD]^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} \quad (6.21)$$

Aplicando el lema a (6.14) que es

$$R(n,i+1) = R(n,i) + Y(i+1)Y^T(i+1)$$

³² S.T.Alexander, A Derivation of the Complex Fast Kalman Algorithm, IEEE Transactions on Acoustics, Speech, and Signal Processing. December 1984.

³³ Friedlander Benjamin, Adaptive Algorithms for Finite Impulse Response Filters, en la obra citada de Cowan and Grant.

tenemos:

$$P(n, i+1) = P(n, i) - P(n, i)Y(i+1)\{(Y^T(i+1))^{-1} + P(n, i)Y(i+1)\}^{-1}P(n, i) \quad (6.22)$$

$$= P(n, i) - \frac{P(n, i)Y(i+1)P(n, i)}{1 + P(n, i)Y(i+1)Y^T(i+1)} \quad (6.23)$$

$$= P(n, i) - \frac{P(n, i)Y(i+1)Y^T(i+1)P(n, i)}{1 + Y^T(i+1)P(n, i)Y(i+1)} \quad (6.24)$$

Definiendo:

$$D(i+1) = \frac{P(n, i)Y(i+1)}{1 + Y^T(i+1)P(n, i)Y(i+1)} \quad (6.25)$$

$$P(n, i+1) = P(n, i) - D(i+1)Y^T(i+1)P(n, i) \quad (6.26)$$

Usando la sugerencia dada por S.T.Alexander, de postmultiplicar (6.26) en ambos miembros por $Y(i+1)$, y usando (6.25), tenemos:

$$P(n, i+1)Y(i+1) = P(n, i)Y(i+1) - D(i+1)Y^T(i+1)P(n, i)Y(i+1) \quad (6.27)$$

$$= D(i+1)(1 + Y^T(i+1)P(n, i)Y(i+1)) - D(i+1)Y^T(i+1)P(n, i)Y(i+1) \quad (6.28)$$

lo que da:

$$D(i+1) = P(n, i+1)Y(i+1) \quad (6.29)$$

$$D(i) = R(n, i)^{-1} Y(i) \quad (6.30)$$

que es la relación buscada para D.

Adicionalmente definamos el escalar gama, g:

$$g(n, i) = Y^T(i) D(n, i) = Y^T(i) R(n, i)^{-1} Y(i) \quad (6.31)$$

6.3 Ecuaciones para actualización del orden de A, C, del error directo y reverso, de la energía y de los coeficiente de reflexión.

Hemos llegado así al punto en donde haremos uso del algoritmo de Levinson-Durbin para la eficiente inversión de la matriz de covarianza.

Siguiendo el desarrollo usual de las ecuaciones recursivas, asumiendo que los vectores de coeficientes $A(n, i)$ y $C(n, i)$ son conocidos, están por determinarse los predictores de orden $n+1$.

A partir de (6.10), asumimos que:

$$R(n+1, i) * A(n+1, i) = \begin{bmatrix} s(n+1, i) \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (6.32)$$

Puesto que $A(n+1, i)$ va a ser formada a partir de $A(n, i)$ y $C(n, i-1)$, estos vectores de orden n son aumentados con un cero al final para dar un orden $n+1$.

El $A(n, i)$ aumentado satisface la siguiente ecuación, excepto por la última entrada:

$$R(n+1,i) \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} = \begin{vmatrix} R(n,i) & * \\ * & * \end{vmatrix} \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} = [sf(n,i) \ 0 \dots 0 \ \alpha(n+1,i)] \quad (6.33)$$

$$\alpha(n+1,i) = [\text{última columna de } R(n+1,i)] \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} \quad (6.34)$$

Similarmente, aumentando $C(n,i-1)$ con un cero al principio satisface las ecuaciones normales excepto por la primera entrada.

$$R(n+1,i) \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} = \begin{vmatrix} * & * \\ * & R(n,i-1) \end{vmatrix} \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} \quad (6.35)$$

$$= [\Gamma(n+1,i) \ 0 \dots 0 \ sb(n,i-1)]$$

$$\Gamma(n+1,i) = [\text{primera columna de } R(n+1,i)] \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} \quad (6.36)$$

Debido a que (6.32) exige que la última entrada sea nula, debemos combinar adecuadamente las ecuaciones (6.33) y (6.35) de modo de cancelar los términos α y Γ .

Previamente usaremos el hecho de que:

$\alpha(n+1,i) = \Gamma(n+1,i)$, lo cual implica que las energías directa y reversa sean iguales, es decir:

$$sf = sb.$$

Este hecho fue experimentalmente comprobado en la simulación, donde la igualdad se cumplió hasta el orden de las milésimas.

El hecho se halla consignado en Turner (ob.cit.), en donde se establece que cuando la señal $y(t)$ es estacionaria, con función de autocorrelación conocida, las energías de predicción del error directo y reverso son idénticas en cada etapa.

Para continuar, conforme señala Turner, (sólo señala), y como nosotros demostraremos, la cancelación de α y Γ se produce como sigue.

Multiplicando (6.35) por $\alpha(n+1,i)/sb(n,i-1)$ y al resultado substrayéndolo de (6.33):

$$R(n+1,i) \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} \begin{matrix} \alpha(n+1,i)/sb(n,i-1) \\ \alpha(n+1,i)/sb(n,i-1) \\ [\Gamma(n+1,i)0..0 \text{ sf}(n,i)] \end{matrix} \quad (6.37)$$

restándolo de (6.33) que reproducimos a continuación,

$$R(n+1,i) \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} = \begin{vmatrix} R(n,i) & * \\ * & * \end{vmatrix} \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} = [\text{sf}(n,i) \ 0..0 \ \alpha(n+1,i)] \quad (6.33)$$

se tiene:

$$\begin{aligned} R(n+1,i) \left(\begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} - \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} \alpha(n+1,i)/sb(n,i-1) \right) &= \\ &= \begin{vmatrix} \text{sf}(n,i) - \alpha(n+1,i)\Gamma(n+1,i)/sb(n,i-1) \\ 0 \\ \vdots \\ \alpha(n+1,i) - \alpha(n+1,i) \end{vmatrix} \\ &= \begin{vmatrix} \text{sf}(n,i) - \alpha^2(n+1,i)/sb(n,i-1) \\ 0 \\ \vdots \\ 0 \end{vmatrix} \quad (6.38) \end{aligned}$$

Ecuaciones con las cuales hemos logrado cancelar α y Γ , a la vez hemos obtenido las actualizaciones de la matriz A así como de $\text{sf}(n,i)$ como vemos a continuación. De (6.38) se ve que el coeficiente que multiplica a la matriz de covarianza es el verdadero $A(n+1,i)$, puesto que hemos logrado cancelar la última entrada.

$$A(n+1,i) = \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} - \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} p(n+1,i)/sb(n,i-1) \quad (6.39)$$

en donde hemos renombrado a α como p , puesto que con ese nombre la usamos en el programa en lenguaje C.

La actualización de sf se ve del segundo miembro de (6.38) y es

$$sf(n+1,i) = sf(n,i) - p^2(n+1,i)/sb(n,i-1) \quad (6.40)$$

Multiplicando (6.33) por $\Gamma(n+1,i)/sf(n,i)$ y al resultado substrayéndolo de (6.35) se obtiene la actualización de C y de sb :

$$C(n+1,i) = \begin{vmatrix} 0 \\ C(n,i-1) \end{vmatrix} - \begin{vmatrix} A(n,i) \\ 0 \end{vmatrix} p(n+1,i)/sf(n,i) \quad (6.41)$$

$$sb(n+1,i) = sb(n,i-1) - p^2(n+1,i)/sf(n,i) \quad (6.42)$$

Cuando las ecuaciones de actualización de A y C son premultiplicadas por $Y^T(i)$, resultan las ecuaciones de malla:

$$Y^T A(n+1,i) = Y^T A(n,i) - Y^T C(n,i-1) p(n+1,i)/sb(n,i-1) \quad (6.43)$$

definiendo,

$$\begin{aligned} A(n,i)^T &= [1 \ a(1,i) \dots a(n,i)] \\ C(n,i)^T &= [c(n,i) \dots c(1,i) \ 1] \\ f(n,i) &= A^T(n,i)Y(i) \text{ y} \\ r(n,i) &= C^T(n,i)Y(i) \text{ , se tiene} \end{aligned} \quad (6.43a)$$

$$f(n+1,i) = f(n,i) - kb(n+1,i)r(n,i-1) \quad (6.44)$$

que es la ecuación de actualización del error directo y donde:

$$kb(n+1,i) = p(n+1,i)/sb(n,i-1) \quad (6.45)$$

es la ecuación de actualización del coeficiente de reflexión reverso.

De idéntica forma se llega a:

$$r(n+1,i) = r(n,i-1) - kf(n+1,i)f(n,i) \quad (6.46)$$

y a:

$$kf(n+1,i) = p(n+1,i) / sf(n,i) \quad (6.47)$$

que actualizan el error reverso y el coeficiente directo de reflexión.

6.4 Ecuaciones de actualización del orden de D y gama.

El razonamiento que sigue es más bien intuitivo pero conlleva a los resultados mostrados en Turner (que da unas instrucciones en extremo simplificadas de cómo actualizar D).

Dejemos establecida la ecuación de recurrencia de D que será usada más tarde:

$$R(n+1,i) \begin{vmatrix} 0 \\ D(n-1,i) \end{vmatrix} = \begin{vmatrix} R(n,i) & * \\ * & * \end{vmatrix} \begin{vmatrix} 0 \\ D(n-1,i) \end{vmatrix} = [* y(i) \dots y(i-n)]^T \quad (6.48)$$

Para actualizar $D(n,i)$, primero tratamos de hallar, para el orden siguiente, cuál es el último elemento en D, por lo que nos interesa el último elemento de $D(n+1,i)$, que simbolizaremos: último elemento $D(n+1,i)$.

A partir de (6.30),

$$\text{último elemento } D(n+1,i) = \text{último elemento } (R(n+1,i)^{-1}Y(i))$$

si imaginamos que el vector C afecta tanto a Y como a R de forma tal de formar los conocidos productos:

$CTY = b = \text{error reverso (6.43a)}$; y

$RC = [0 \dots sb(n+1,i)]$, (6.19); y puesto que R "está en el denominador",

= último elemento $(b(n+1,i)/[0 \dots sb(n+1,i)])$

= último elemento $([0 \dots b(n+1,i)]/sb(n+1,i))$

= $b(n+1,i)/sb(n+1,i)$

Puesto que el último elemento de $C(n+1,i)$ es 1, y como el último elemento de $D(n+1,i)$ ya fue determinado, la ecuación de recursión en el orden de $D(n+1,i)$ es:

$$D(n+1,i) = \begin{vmatrix} D(n,i) \\ 0 \end{vmatrix} + C(n+1,i) b(n+1,i)/sb(n+1,i) \quad (6.49)$$

La ecuación de actualización del orden de gama se determina premultiplicando (6.49) por $Y(i)^T$ y de (6.31) y (6.43a).

$$g(n+1,i) = g(n,i) + b^2(n+1,i)/sb(n+1,i) \quad (6.50)$$

6.5 Ecuaciones de actualización del tiempo.

Para proseguir explotaremos la propiedad de actualización en el tiempo de la matriz de covarianza, dada por (6.15). Postmultiplicando (6.15) por $A(n,i)$:

$$R(n+1,i)A(n,i) = R(n,i)A(n,i) + Y(i+1)Y(i+1)^T A(n,i) \quad (6.51)$$

Para reescribir (6.51) hagamos las siguientes asumpciones para el estado estable, es decir cuando el filtro ya ha alcanzado la convergencia:

- $Y(i+1) = Y(i)$, es decir que el mejor estimado no varía al tomar

una nueva muestra, puesto que el algoritmo ya convergió; y

- $R(n,i)A(n,i) = R(n+1,i)A(n+1,i)$, lo que implica que el valor de la potencia de los bits recibidos es una constante independiente de la etapa en que nos encontremos. Este hecho fue comprobado experimentalmente en la simulación.

De modo que la ecuación (6.51) puede ser reescrita como:

$$R(n+1,i)A(n,i) = R(n+1,i)A(n+1,i) + Y(i+1)Y^T(i)A(n,i) \quad (6.52)$$

Premultiplicando ambos miembros de (6.52) por $R(n+1,i)^{-1}$,

$$A(n,i) = A(n+1,i) + R(n+1,i)^{-1} \begin{bmatrix} y(i+1) \\ \vdots \\ y(i-n+1) \end{bmatrix} * [y(i) \dots y(i-n)] A(n,i) \quad (6.53)$$

usando (6.48),

$$A(n+1,i) = A(n,i) - \begin{bmatrix} 0 \\ D(n-1,i) \end{bmatrix} Y(i)^T A(n,i) \quad (6.54)$$

premultiplicando (6.54) por $Y(i+1)^T$ y usando la definición de *gama* (6.31):

$$Y(i+1)^T A(n+1,i) = Y(i)^T A(n,i) - Y(i)^T \begin{bmatrix} 0 \\ D(n-1,i) \end{bmatrix} Y(i)^T A(n,i) \quad (6.55)$$

$$f(n,i+1) = Y(i)^T A(n,i) - g(n-1,i) Y(i)^T A(n,i) \quad (6.56)$$

de donde

$$Y(i)^T A(n,i) = f(n,i+1) / (1 - g(n-1,i)) \quad (6.57)$$

con lo que de (6.54) obtenemos la actualización en el tiempo para $A(n,i+1)$:

$$A(n,i+1) = A(n,i) - \left| \begin{array}{c} 0 \\ D(n-1,i) \end{array} \right| f(n,i+1)/(1-g(n-1,i)) \quad (6.58)$$

Para la actualización de la energía reversa, establezcamos los siguientes lineamientos en estado estable:

-la energía acumulada es igual en todas las etapas:

$$1) R(n+1,i)A(n+1,i) = R(n,i)A(n,i)$$

-los coeficientes difieren de una etapa a otra:

$$2) A(n+1,i) \neq A(n,i)$$

-la covarianza de orden superior contiene a la de orden inferior:

$$3) R(n+1,i) \neq R(n,i)$$

-la energía para un orden dado cambia en el transcurso del tiempo:

$$4) R(n,i+1)A(n,i+1) \neq R(n,i)A(n,i)$$

-la matriz de covarianza al tiempo $i+1$ contiene a la matriz del tiempo anterior:

$$5) R(n,i+1) \neq R(n,i)$$

-cuando han convergido, los coeficientes no cambian en el tiempo:

$$6) A(n,i+1) = A(n,i)$$

-la predicción al tiempo $i+1$ da esencialmente el mismo valor que al tiempo anterior:

$$7) Y(i+1) = Y(i).$$

De los cuales 1,2,4 y 6 fueron comprobados experimentalmente en la simulación.

Con ellos podemos actualizar $sb(n,i)$ de la siguiente manera:

$$\text{sea } sb(n, i+1) \triangleq A^T(n, i+1)R(n, i+1)A(n, i+1) \quad (6.59)$$

Usando (6.15) para la actualización en el tiempo de R, y (6.58),

$$\begin{aligned} R(n, i+1)A(n, i+1) = & R(n, i)A(n, i) - R(n, i) \begin{vmatrix} 0 \\ D(n-1, i) \end{vmatrix} Y^T(i)A(n, i) + \\ & + Y(i+1)Y^T(i+1)A(n, i) - Y(i+1)Y^T(i+1) \begin{vmatrix} 0 \\ D(n-1, i) \end{vmatrix} Y^T(i)A(n, i) \end{aligned} \quad (6.60)$$

Despejemos $\begin{vmatrix} 0 \\ D(n-1, i) \end{vmatrix} Y^T(i)$ de (6.58)

obtenemos:

$$\begin{vmatrix} 0 \\ D(n-1, i) \end{vmatrix} Y^T(i) = (1 - A(n, i+1)/A(n, i)) \quad (6.61)$$

por lo que (6.60) da

$$\begin{aligned} R(n, i+1)A(n, i+1) = & R(n, i)A(n, i) - R(n, i)(1 - A(n, i+1)/A(n, i))A(n, i) + \\ & + Y(i+1)Y^T(i+1)A(n, i) - Y(i+1)Y^T(i+1)(1 - A(n, i+1)/A(n, i))A(n, i), \end{aligned}$$

desarrollando y simplificando términos semejantes se tiene

$$R(n, i+1)A(n, i+1) = R(n, i)A(n, i+1) + Y(i+1)Y^T(i+1)A(n, i+1) \quad (6.62)$$

por lo que

$$\begin{aligned} A(n, i+1)R(n, i+1)A(n, i+1) = & A(n, i+1)R(n, i)A(n, i+1) + \\ & + A(n, i+1)Y(i+1)Y^T(i+1)A(n, i+1) \end{aligned} \quad (6.63)$$

en base al lineamiento 2), 6) y al 7) dados,

$$\begin{aligned} A(n, i+1)R(n, i+1)A(n, i+1) = & A(n, i)R(n, i)A(n, i) + \\ & + A(n, i+1)Y(i+1)Y^T(i)A(n, i) \end{aligned} \quad (6.64)$$

y usando la definición de error directo (6.43a), la de energía de error directo (6.59) y la fórmula (6.57) llegamos a

$$sf(n,i+1) = sf(n,i) + f^2(n,i+1)/(1-g(n-1,i)) \quad (6.65)$$

que es la fórmula de actualización de la energía del error directo.

Aplicando similares técnicas, las recursiones para $C(n,i)$ y para $sb(n,i)$ pueden ser obtenidas, lo que resulta en:

$$C(n,i+1) = C(n,i) - \left| \begin{matrix} D(n-1,i+1) \\ 0 \end{matrix} \right| r(n,i+1)/(1-g(n-1,i+1)) \quad (6.66)$$

$$sb(n,i+1) = sb(n,i) + r^2(n,i+1)/(1-g(n-1,i+1)) \quad (6.67)$$

Para actualizar los coeficientes de reflexión recursivamente, se precisa la actualización de p .

$$[0 \ C(n,i)] R(n+1,i+1) \left| \begin{matrix} A(n,i+1) \\ 0 \end{matrix} \right| = p(n+1,i+1) \quad (6.68)$$

Usando la actualización de la covarianza (6.15), el predictor A actualizado a través de las ecuación (6.58) y C , se obtiene la actualización de p :

$$p(n+1,i+1) = p(n+1,i) + r(n,i)f(n,i+1)/(1-g(n-1,i)) \quad (6.69)$$

Es importante notar que el desarrollo de las ecuaciones de actualización en el tiempo permite que se acumule la influencia de todos los datos previos en los parámetros estimados en curso.

Si el proceso que se estima contiene atributos que son variables en el tiempo, es deseable sopesar con mayor importancia a las recientes observaciones. Se puede incluir un factor exponencial de olvido w en las covarianzas acumuladas. Los

valores típicos van desde 0.8 a 1 (1 corresponde a contemplar todos los datos pasados). El algoritmo que presentamos subsecuentemente contempla el factor de olvido w .

Debemos anotar que en la simulación se varió el factor de olvido desde 0.3 a 1, observándose una enorme influencia sobre la convergencia de los coeficientes, la cual será comentada a su debido tiempo.

6.6 El filtro Lattice para procesos conjuntos, específico para la ecualización.

En los acápites anteriores se ha deducido en algún detalle el algoritmo para la actualización de los parámetros de la parte "tipo predicción lineal" del Filtro de Malla, que es la parte que se encarga de ortogonalizar la señal de entrada.

El proceso de la ecualización requiere sin embargo, no sólo de la parte mencionada, sino también de la parte "tipo TDL". Esto es así debido a que el proceso total requiere el comparar la señal ortogonal con la señal de referencia almacenada ó con la señal de referencia realimentada (en el período de autoajuste).

Este tipo de procesos que requieren de dos señales son conocidos en la literatura como procesos conjuntos.

Para el efecto debe definirse un nuevo error de predicción que incluye muestras de ambos procesos. Al error lo hemos llamado $c(n,i)$, lo cual se ve de inmediato de la fig. 18. Los dos procesos mencionados son, de la fig.18, $y(i)$ y $b(i)$ ($a_N(i)$ en el proceso de autoajuste). De esta forma, el error de predicción de N -ésimo orden $c(N,i)$ es el error que deviene de estimar b_i a partir de $\{y_i, y_{i-1}, \dots, y_{i-N}\}$, donde $g(i,N)$ son los coeficientes de predicción obtenidos al minimizar la suma de los errores cuadrados.

$$c(N,i) = b_i - \sum_{n=0}^{N-1} g(n+1) y(n,i) \quad (6.70)$$

Sin embargo, como demostramos en el acápite 4.4, que "para mejorar la velocidad de convergencia del algoritmo se debe tratar de que todos los eigenvalores de la matriz R sean iguales, lo cual se puede lograr mediante la ortogonalización de las señales de entrada a los taps del ecualizador", en lugar de los $y(i)$ usaremos la señal ortogonal $r(i)$, lo que aumenta la velocidad de convergencia, así:

$$c(N,i) = b_i - \sum_{n=0}^{N-1} K^e(n+1) r(n,i) \quad (6.71)$$

Los coeficientes $K^e(n)$ que minimizan $c(N,i)^2$ son, siguiendo el mismo proceso desarrollado en el acápite 5.3,

$$K^e(n+1) = E(c(n,i)r(n,i-1)) / E(r(n,i-1))^2 \quad (6.72)$$

Además, se tiene de (6.71),

$$c(n+1,i) = c(n,i) - K^e(n+1) r(n,i-1) \quad (6.73)$$

El coeficiente p^e asociado a K^e se obtiene a partir de (6.69),

$$p^e(n+1,i+1) = p^e(n+1,i) + c(n,i+1)r(n,i+1)/(1-g(n-1,i)) \quad (6.74)$$

en la cual tanto c como r son tomados al mismo tiempo, lo cual difiere de (6.69) puesto que r se obtiene de f a partir de un elemento de retardo, lo que se ve de la fig. 18.

De la fig.18. se ve que el error del orden 0 no es más que

la señal de entrada, por definición.

Finalmente, de la misma figura se ve que la señal de realimentación en el período de autoajuste $a_n(i)$ se obtiene como un sumatorio a partir de las salidas de los K^c , lo cual es la aplicación incremental de (6.71):

$$a_{n+1}(i) = a_n(i) + K^c(n+1,i)r(n,i) \quad (6.75)$$

Esta es en definitiva la obtención de las ecuaciones de actualización para el proceso conjunto denominado Filtro de Malla.

6.7 Algoritmo del Filtro de Malla.

El algoritmo consiste de la implementación de las ecuaciones de actualización del orden y del tiempo deducidas anteriormente.

Se inicializa reseteando las siguientes variables:

```
ra[n]      ^error reverso anterior (corresponde a r[n,i-1])
ga[n]      ^gama anterior
pa[n+1]    ^coeficiente "p" anterior de la parte tipo "predicción
           ^lineal"
pca[n+1]   ^coeficiente "p" anterior de la parte tipo "TDL",
e inicializando al parámetro covarianza reversa anterior a un
valor pequeño que asegure inversibilidad de la matriz de
covarianza,
sba[n]=eps
y la covarianza directa anterior de orden 0,
sfa[0]=eps
```

Seguidamente se inicia el procesamiento de las secuencias de 256 datos..

Como se ve de la fig. 18, el error directo y reverso de

orden cero son por definición el valor de la entrada a la parte tipo "predicción lineal", por lo que esto se especifica en el inicio de la secuencia de 256 bits. También se da los valores de partida de la energía directa y la energía reversa de orden cero, $sf[0]$ y $sb[0]$ (de la definición de matriz de covarianza de orden 0). Se inicializa γ de orden 0 a cero y se resetea la salida del ecualizador de orden 0 (valores de inicio de la iteración).

Se debe hacer notorio que la inicialización del error $c[0]$ depende de si estamos en el entrenamiento o si estamos en el autoajuste. En el entrenamiento, de la fig.18, se ve que $c[0]$ no es más que la señal almacenada $b_r[i]$, mientras que en el autoajuste es igual a la salida del ecualizador. Esto determina que con el objeto de inicializar $c[0]$ en el autoajuste debemos calcular previamente $a[ca]$, que es la salida del ecualizador.

A continuación se calcula iterativamente los ρ (de correlación parcial, Turner ob. cit.) $\rho[n+1]$ de la parte tipo "predicción lineal", con ellos se calcula los coeficientes de reflexión reversos $kb[n+1]$ y directos $kf[n+1]$, los errores directo $f[n+1]$ y reverso $r[n+1]$, la covarianza directa $sf[n+1]$ e inversa $sb[n+1]$ y la variable probabilística γ , $g[n+1]$.

Procedemos a calcular los ρ $\rho[n+1]$ de la parte "tipo TDL" usando el valor hallado de $c[0]$ en el entrenamiento (autoajuste), a partir de los cuales se halla los coeficientes de reflexión $kc[n+1]$, la salida del filtro $a[n+1]$ y el error $c[n+1]$.

Para el cálculo del error $c[n+1]$ tenemos en el entrenamiento que $c[n+1]=b_r[i]-a[n+1]$, es decir se resta el valor almacenado menos la salida del ecualizador, mientras que en el autoajuste se resta la salida realimentada y cuantizada ad_r menos $a[n+1]$.

Luego se efectúa un desplazamiento en los registros, y finalmente se calcula el valor del error de transmisión. Todo lo

afirmado anteriormente se resume en el siguiente algoritmo, el cual está expresado en lenguaje basic, como es constumbre hacerlo en la literatura de telecomunicaciones.

```

for n=0 to ca-1           `Inicialización
ra[n]=0
ga[n]=0
pa[n+1]=0
pca[n+1]=0
sba[n]=eps
next n
sfa[0]=eps

for i=1 to 256           `Secuencias de 256 bits

a[0]=g[0]=0
f[0]=r[0]=y_r[i]
sf[0]=sb[0]=w*sfa[0]+y_r[i]*y_r[i]
sfa[0]=sf[0]

                                `Inicialización del error c[0]
En el entrenamiento       c[0]=b_r[i]
En el autoajuste

for n=0 to ca-1
p[n+1]=w*pa[n+1]+f[n]*ra[n]/(1-ga[n])
kb[n+1]=p[n+1]/sba[n]
kf[n+1]=p[n+1]/sf[n]
f[n+1]=f[n]-kb[n+1]*ra[n]
r[n+1]=ra[n]-kf[n+1]*f[n]
a[n+1]=a[n]+kc[n+1]*r[n]
next n
aca=a[ca]
decide(aca,ad_r)
c[0]=ad_r

for n=0 to ca-1           `Inversión de la matriz de covarianza
p[n+1]=w*pa[n+1]+f[n]*ra[n]/(1-ga[n])
kb[n+1]=p[n+1]/sba[n]
kf[n+1]=p[n+1]/sf[n]
f[n+1]=f[n]-kb[n+1]*ra[n]
r[n+1]=ra[n]-kf[n+1]*f[n]
sf[n+1]=sf[n]-p[n+1]*kb[n+1]
sb[n+1]=sba[n]-p[n+1]*kf[n+1]
g[n+1]=g[n]+r[n]*r[n]/sb[n]

                                `Cálculo del error c[n+1]
En el entrenamiento       pc[n+1]=w*pca[n+1]+c[n]*r[n]/(1-g[n])
                                kc[n+1]=pc[n+1]/sb[n]

```

```

a[n+1]=a[n]+kc[n+1]*r[n]
c[n+1]=b_r[i]-a[n+1]

```

En el autoajuste

```

pc[n+1]=w*pca[n+1]+c[n]*r[n]/(1-g[n])
kc[n+1]=pc[n+1]/sb[n]
a[n+1]=a[n]+kc[n+1]*r[n]
aca=a[n+1]
decide(aca,ad_r)
c[n+1]=ad_r-a[n+1]

```

```

ra[n]=r[n]           `Desplazamiento en los registros.
sba[n]=sb[n]
pa[n+1]=p[n+1]
pca[n+1]=pc[n+1]  ga[n]=g[n]

```

next n

```

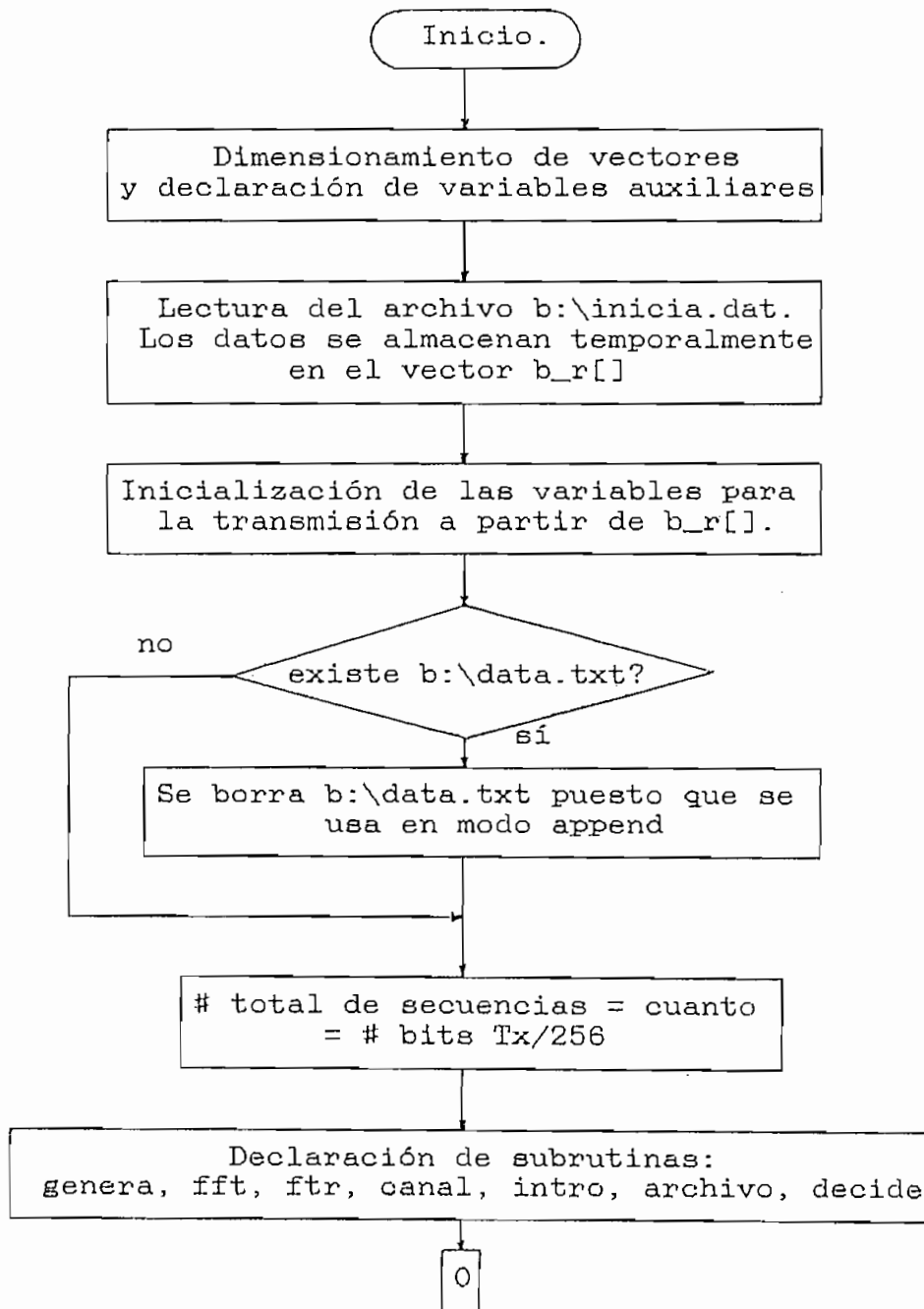
aca=a[ca]
decide(aca,ad_r)   `Cálculo del Error de Transmisión
err_t_a=err_t_a+abs{(b_r[i]-ad_r)/2}

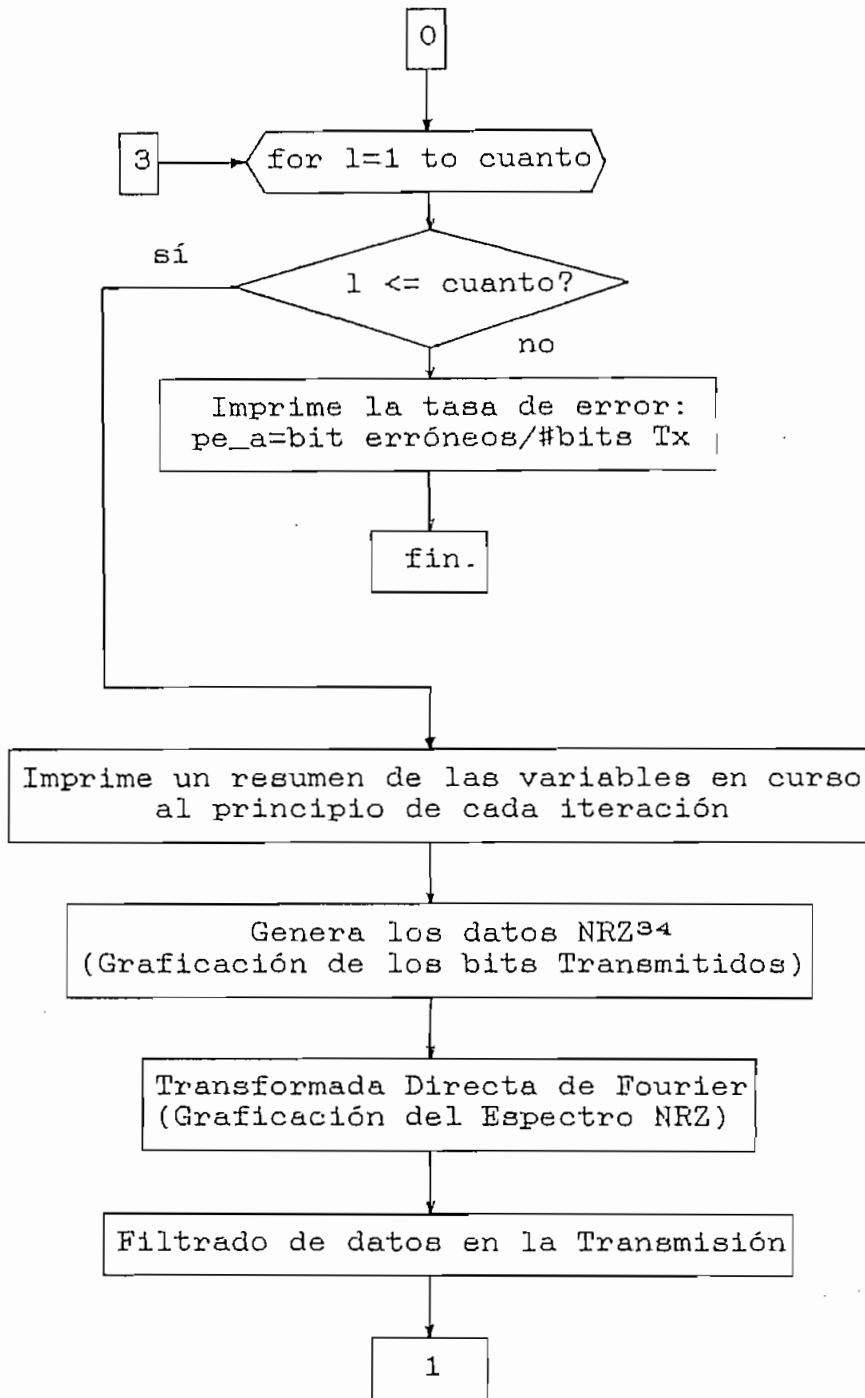
```

next i `FIN DEL ALGORITMO DEL FILTRO DE MALLA.

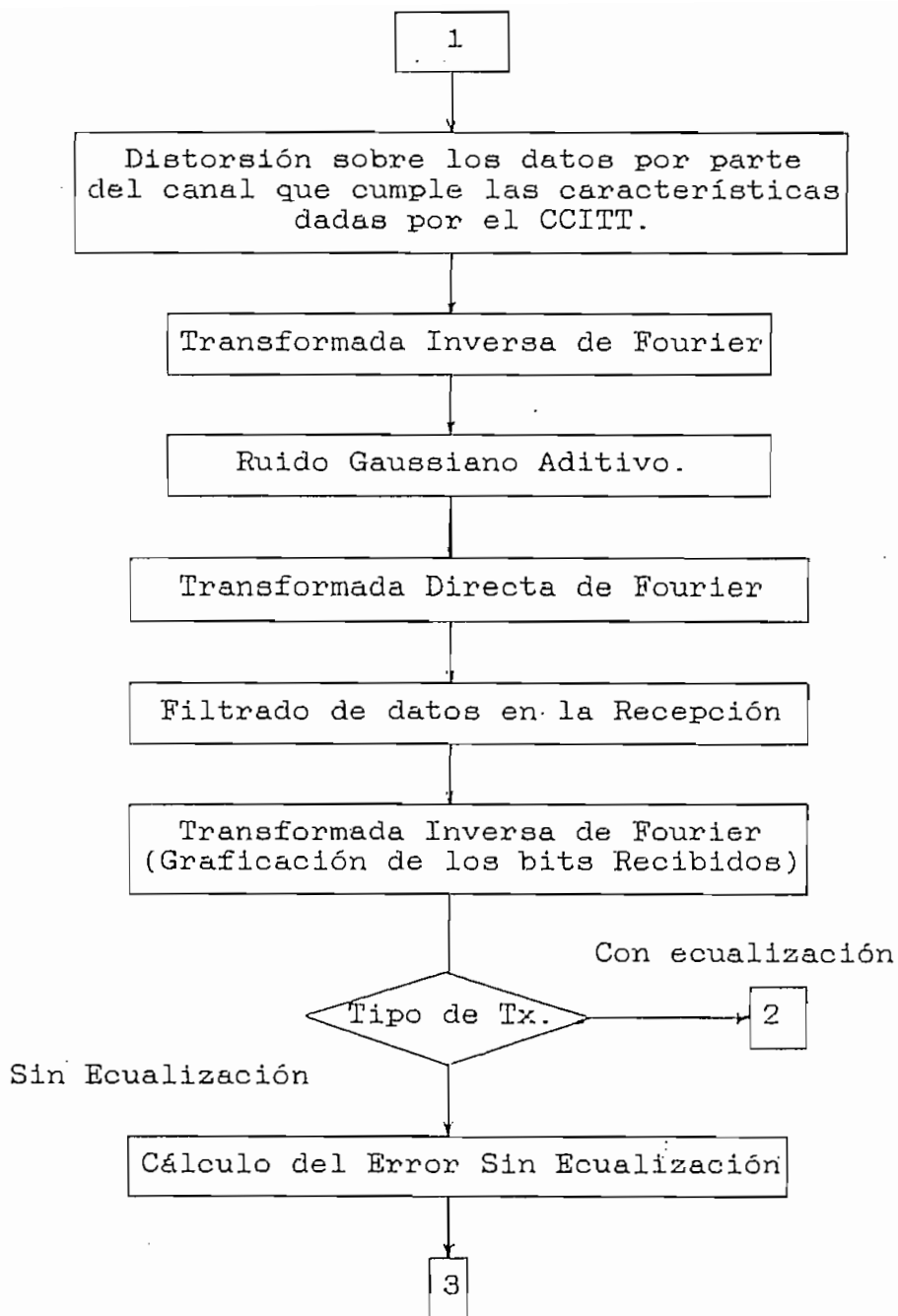
6.8 Diagrama de Flujo General del Programa.

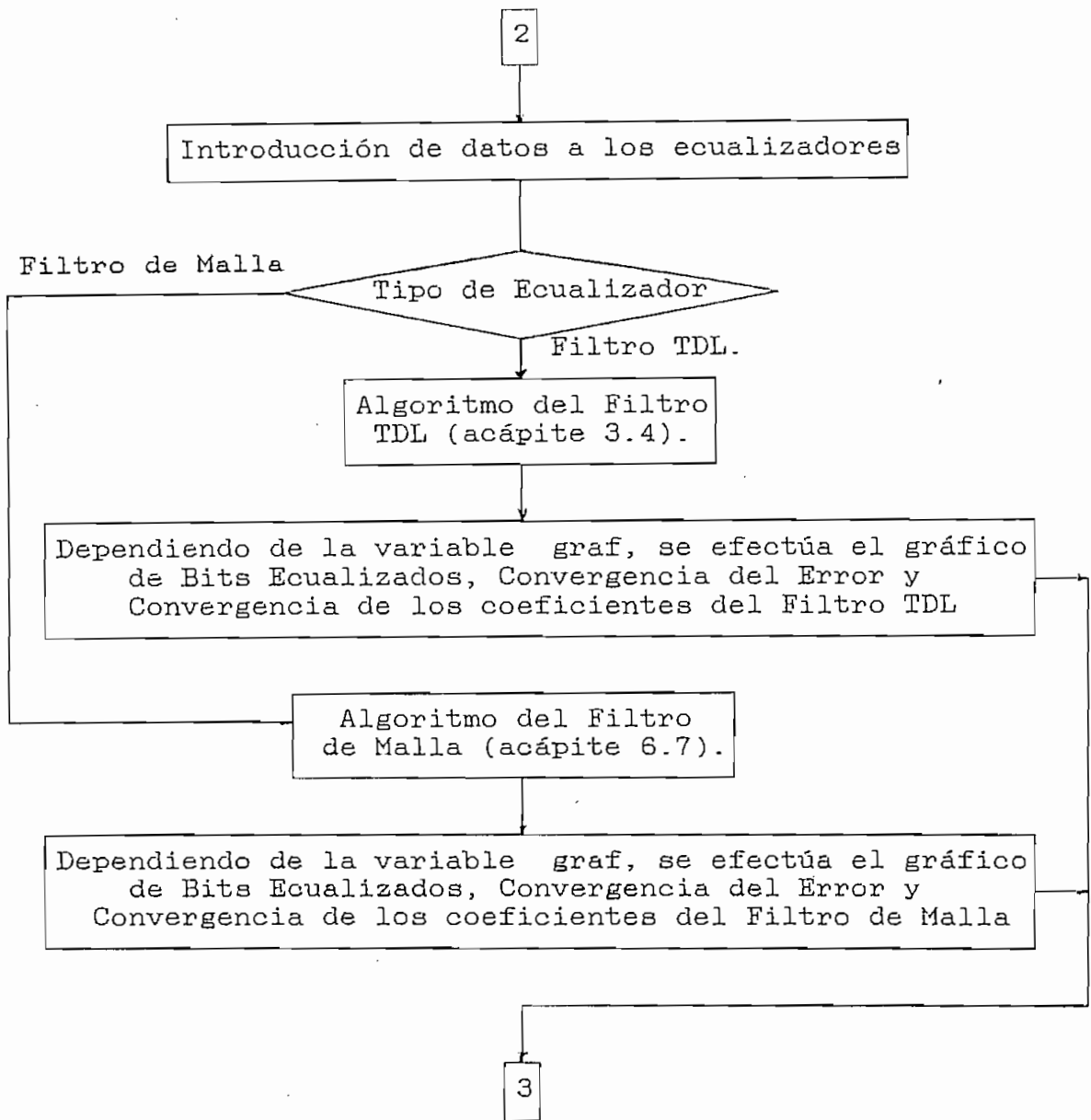
El diagrama general del programa, que contempla transmisión sin ecualización y con ecualización (Filtro TDL y Filtro de Malla), se presenta a continuación:





³⁴ La graficación en cada ítem es opcional, por lo que se la muestra entre paréntesis.





CAPITULO VII.

CAPITULO VII

RESULTADOS DE LA SIMULACION PARA EL CANAL DE TRANSMISION.

A continuación procedemos a sumarizar los resultados obtenidos en la simulación realizada.

7.1 Simulación de la transmisión a través de un canal con distorsión.

Como primer paso hacia la simulación se ha elegido el modelo matemático del canal, el cual fue ya detallado en el capítulo 2.

A continuación se precisa generar el tren aleatorio de datos codificados en NRZ de valores +1, -1.

Las secuencias deben ser aleatorias para minimizar la probabilidad de largas secuencias de 0s (-1) o 1s (+1) que determinan la pérdida de sincronismo en una transmisión real, así como para homogeneizar el espectro de potencia.

Esto se logra con la subrutina genera, la cual hace uso a su vez de la subrutina randu la que genera valores aleatorios con distribución uniforme.

Los datos aleatorios comprenden secuencias de 256 bits, el cual es un valor de compromiso entre velocidad para la simulación y suficientes valores para obtener un espectro más o menos denso (para el caso de la computadora 80386 a 25 MHz. usada). En el caso de un supercomputador, como el usado en Proakis (ob. citada.) se habla de secuencias de 20000 a 30000 bits.

A continuación, se muestrea cada bit 8 veces en un período, y el resultado obtenido se muestra en la fig.19.

Se debe anotar que si bien la figura 19. presenta el primer cruce por 0 en 256, que corresponde al vector complejo $y[256]$, lo que en el dominio de la frecuencia representa una frecuencia de $f = 256$ Hz, la misma figura puede ser usada para representar cualquier otra velocidad de transmisión. Por ejemplo, si usamos una portadora de 1800 Hz (recomendada por el CCITT para transmisión por modems a 1200 y 2400 bits/s, recomendación que tiene su fundamento como veremos posteriormente) y una velocidad de transmisión de 1200 bits/s, estamos hablando de un primer cruce por 0 en 3000 Hz.

Para poder almacenar este cero en el vector $y[256]$, debemos transformar 3000 Hz a la posición 256. Para ello simplemente se resta la portadora y al resultado se divide para delta; de donde resulta que $\text{delta} = 1200/256 = 4.68 = \text{baud} / 256$, donde delta y baud son variables usadas en el programa y que significan respectivamente paso unitario y velocidad de transmisión.

7.2 Espectro de frecuencia de la señal NRZ transmitida.

El espectro de potencia de la señal NRZ (Feher, ob. cit), es la función

$$2A^2T(\text{sen}(\pi fT)/(\pi fT))^2 \quad (7.2.1)$$

cuyo primer cero corresponde a la mencionada velocidad de transmisión, ya que

$$\text{sen}(\pi fT) = 0 \text{ para } \pi fT = \pi,$$

lo que da que $f=1/T$, que es la velocidad de transmisión.

La fig.19. nos muestra el espectro de *voltage* de la señal NRZ el cual es aproximadamente la función $\text{sen } x / x$, es decir la

raíz cuadrada de (7.2.1) -asumiendo una resistencia de 1 ohm. Presenta ceros en múltiplos de 256 y va de 1 a 2048 en la frecuencia, que no es más que el resultado de 8 muestras por bit * 256 bits, que es lo que asumimos previamente. Es de aquí que surge la necesidad de disponer de un vector de dimensión = 2048.

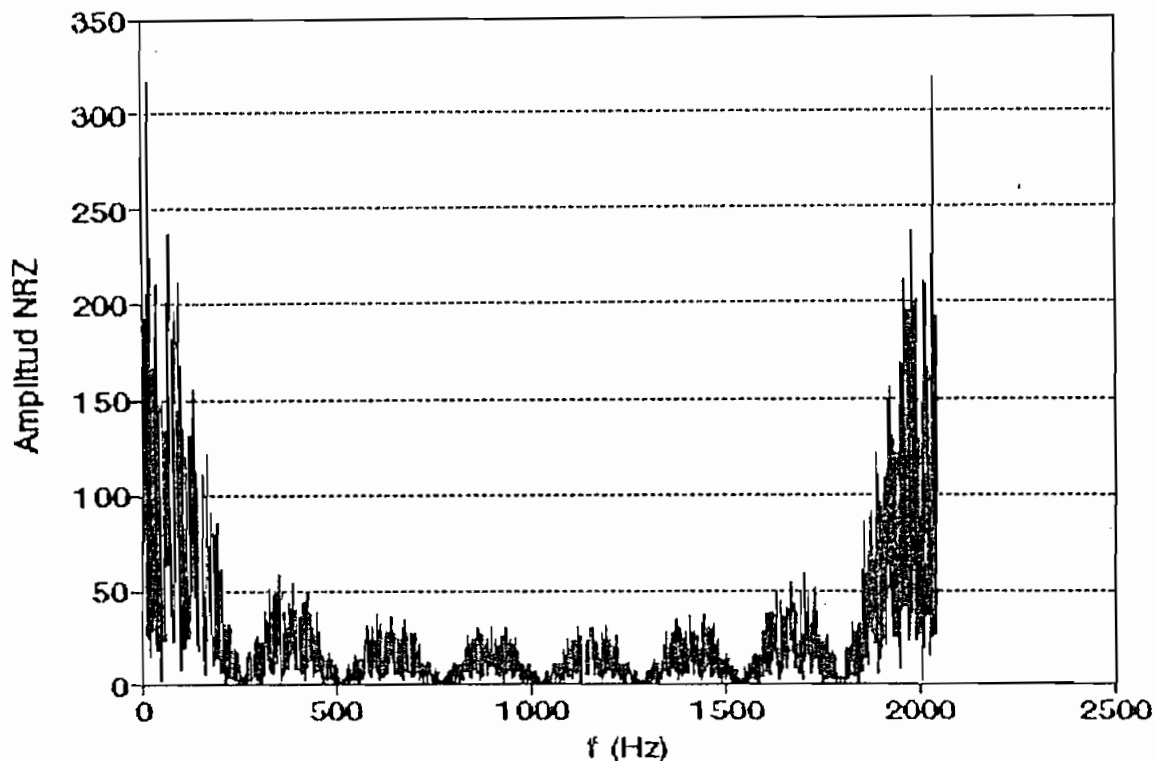


Figura 19 Amplitud del Espectro de la señal NRZ versus frecuencia.

En los gráficos que siguen hemos seguido la convención: eje y= eje vertical= amplitud; eje t = eje horizontal= número de iteraciones (número de bits) de una corrida de la simulación.

Con la transformada de Fourier del tren NRZ almacenada, procedemos a formar la convolución del tren con el filtro de transmisión para lo cual se usa la rutina ftr.

Como se ve en el capítulo 2, la condición de filtro acoplado y de filtro que cumple el criterio de Nyquist de ISI nula, determina que la función de transferencia del filtro sea el

producto $x/\text{sen}(x) \sqrt{H(f)}$, donde $H(f)$ es la transformada de la función coseno elevado.

Formamos el producto en la frecuencia y usamos la subrutina canal, la cual implementa las características de distorsión de amplitud y de fase para un canal telefónico, como recomienda el CCITT (Núñez, ob. cit.).

Seguidamente procedemos a tomar la transformada inversa de Fourier de modo de volver al dominio del tiempo con el objeto de añadir el ruido gaussiano de $\text{SNR} = 36 \text{ dB}$. Un dato útil de niveles de ruido en línea telefónica consiste en asumir 10000 pWp0 para una conversación libre de ruido³⁵, 100000 pWp0 para una conversación con ruido y 1e6 pWp0 cuando el ruido es excesivo. Los 100000 pWp0 conducen a -40 dBmOp , que está cerca al valor de -38 dBmOp recomendado por el CCITT. Este valor de -38 dBmOp conduce a 36 dB , lo que para la potencia normalizada de 1 W , da el valor de $.00025 \text{ W}$ para la potencia del ruido y esto a su vez da un voltaje de ruido de $.015 \text{ V}$ sobre una resistencia de 1 ohm , el mismo que lo inyectamos al canal. Hecho esto volvemos al dominio de la frecuencia.

En la recepción formamos la convolución con el filtro receptor, el cual está acoplado al filtro transmisor de modo de maximizar la relación señal a ruido.

Luego de hecho esto, la transmisión está completa y sólo resta tomar la transformada inversa de Fourier del tren recibido.

Efectuamos esta operación con la misma subrutina `fft`, luego de lo cual enviamos los datos al archivo `b:\data.txt` mediante la subrutina `archivo`.

³⁵ Posada, Omar. Introducción a los Sistemas de Microondas. Agosto de 1969. Dept. de Publicaciones. Facultad de Ing. Eléctrica.

7.3 Bits recibidos para diferentes velocidades de transmisión.

Hemos graficado los datos en QPRO y se los muestra en la fig. 20.

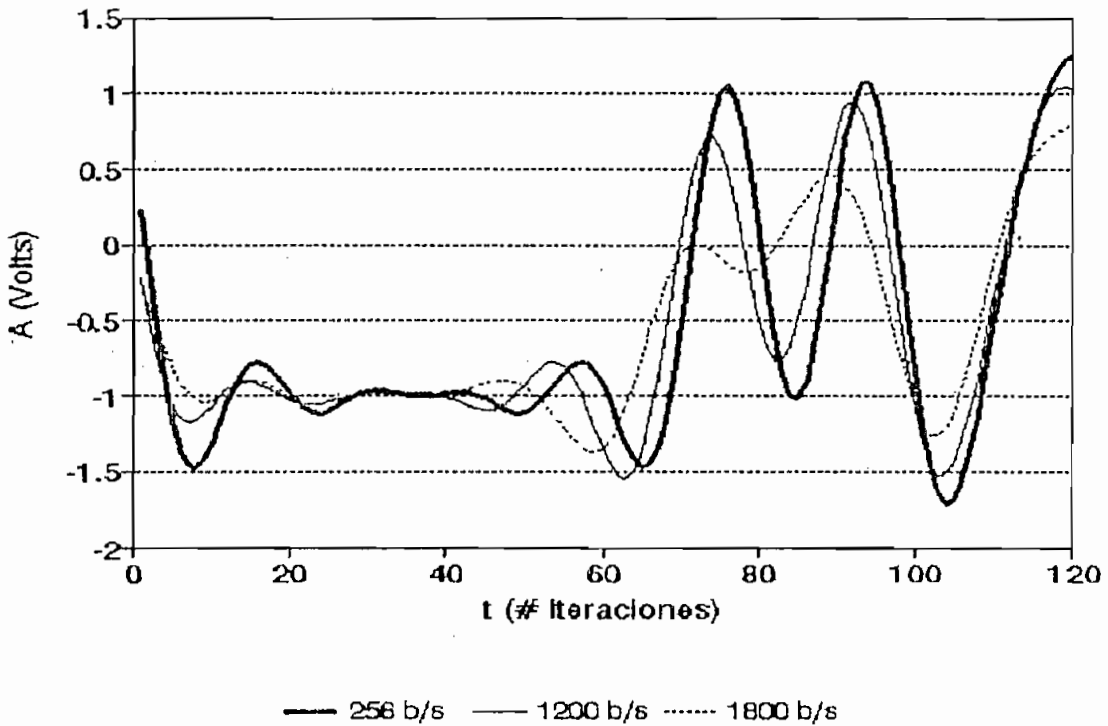


Figura 20 Pulsos recibidos para 3 diferentes velocidades de modulación. Eje A=amplitud de los bits NRZ (+1,-1), Eje t=número de iteraciones.

Esta figura amerita un comentario detallado.

La figura fue obtenida enviando una secuencia de 256 bits, usando tres velocidades de transmisión. Se usó un exceso de banda $\alpha = 0.25$, una semilla para generación de números aleatorios $sem = 50$, instante de decisión $= 4$, una relación señal a ruido $SNR = 36$ dB y fue realizado en la modalidad Transmisión Sin Ecuilibración.

En la figura se muestran los primeros 15 datos de la secuencia de 256 bits: -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1,

1, -1, -1, 1.

Cada dato tiene una duración de 8 unidades, ya que hemos normalizado una de las 8 muestras por pulso a la unidad.

Para una velocidad de transmisión baja, de 256 bits/s (línea gruesa), se puede apreciar que los datos recibidos son prácticamente los datos enviados, excepto por la forma adquirida al pasar por los filtros de Tx, Rx, y en menor proporción por la conferida por el canal $H(f)$.

Para una velocidad de transmisión de 1200 bits/s (línea delgada), se observa que la señal recibida está notablemente distorsionada respecto a la señal enviada. En este caso la contribución mayor a la distorsión está dada por las características del canal $H(f)$.

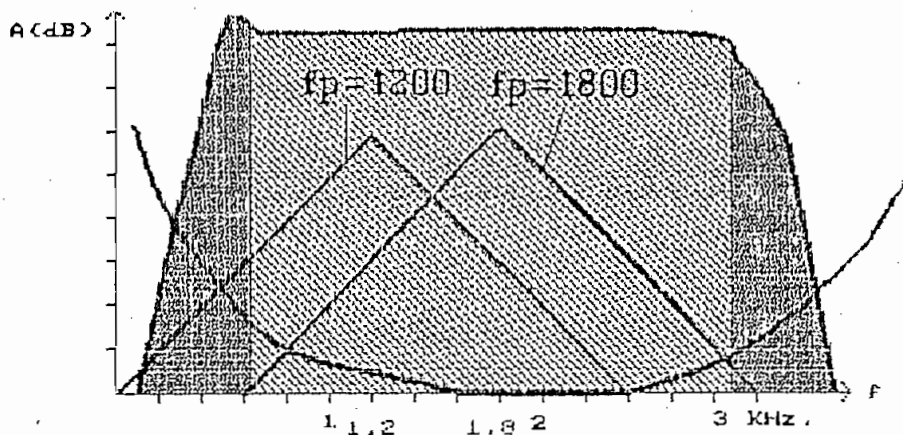
Para una velocidad de transmisión de 1800 bits/s (línea punteada), se observa que la señal recibida está terriblemente distorsionada respecto a la señal enviada. Al igual que en el caso anterior, la distorsión está dada por las características del canal $H(f)$.

7.4 Explicación de la distorsión observada usando las curvas de distorsión del CCITT.

Podemos explicar la distorsión cualitativamente con la fig.21.

En dicha figura hemos aproximado el espectro NRZ, el cual ha sido previamente filtrado con el filtro de Tx, con un triángulo.

El filtro de Tx. determina que la frecuencia de corte esté en $f_n(1+\alpha)$, pero para el análisis que sigue hemos tomado el caso ideal $\alpha=0$, por lo que el corte está justamente en la f_n .





La zona  representa la parte casi lineal del canal, que es donde existe la menor distorsión. En ella se halla el tren fp1800, el cual es el recomendado por el CCITT. La zona  presenta alta distorsión puesto que está en las características NO LINEALES del canal. Allí está el tren fp1200, lo cual explica la distorsión encontrada.

Figura 21 Análisis de la distorsión observada usando la función de transferencia del canal dada por el CCITT.

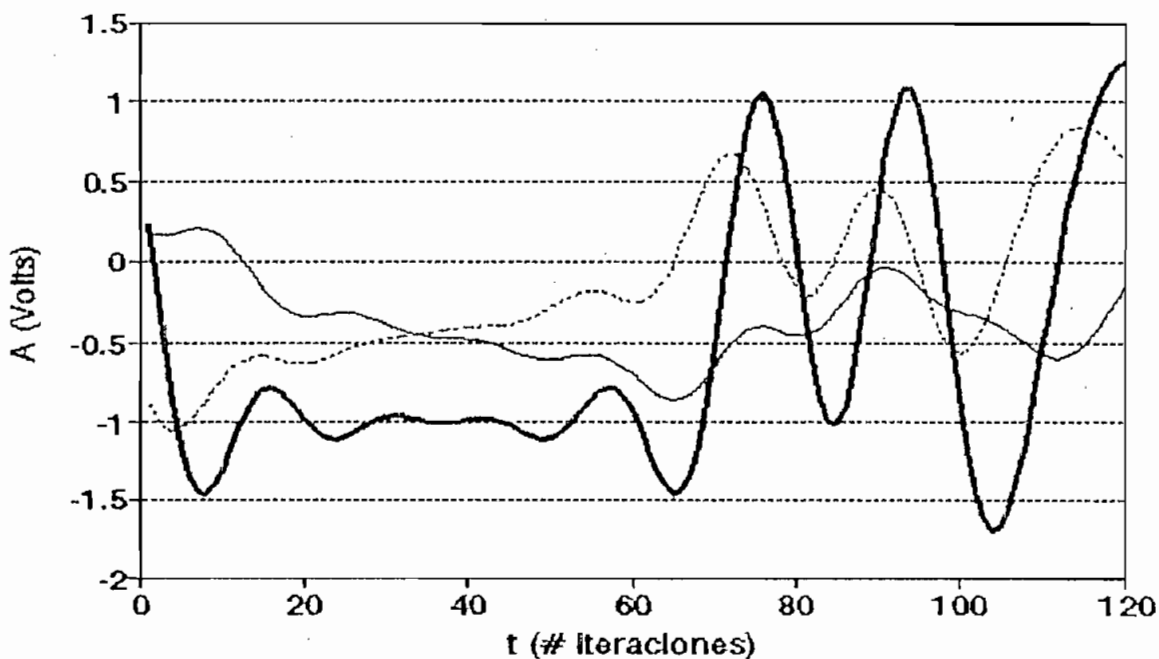
En el caso de una frecuencia baja de transmisión, el triángulo presenta la frecuencia de corte en la zona donde las características de amplitud y fase del canal son casi lineales.

Sin embargo, conforme se aumenta la velocidad de transmisión, entramos cada vez más en la zona no lineal de las características, lo que resulta en la terrible distorsión anotada.

Es por esta razón que la velocidad de transmisión en línea sin ecualización, está limitada a 1200 bits/s para el caso del espectro coseno elevado, que es justamente el caso del gráfico, en donde se ve que para esta velocidad estamos dentro de las características lineales del canal.

Y es aquí donde se aplica el procedimiento de ecualización para contrarrestar principalmente la distorsión de fase introducida por el canal.

De la misma figura se puede mostrar el por qué de la recomendación del CCITT para una frecuencia de portadora de 1800 Hz.



— 2800 Hz — 1800 Hz 1200 Hz

Figura 22 Recepción de los bits para varias frecuencias portadoras. A=amplitud de los bits NRZ; t=# de iteraciones.

La explicación es como en el caso de la distorsión antes analizada: al establecer la portadora en el centro de las características del canal, la distorsión es mínima, en tanto que la distorsión aumenta con el corrimiento hacia cualquiera de los dos extremos del espectro. En la figura 22. se muestran los bits recibidos para 3 diferentes frecuencias portadoras. La figura se llevó a cabo con una semilla de 50, decisión = 4, SNR = 36 dB, y una velocidad de modulación baja de baud = 256 b/s, de modo que el principal factor de distorsión esté constituido por la

variación de la frecuencia portadora.

Nota: En la modulación BPSK cada símbolo enviado corresponde a un bit, sin embargo usaremos el término baud que implica el envío de uno o más bits para dar cuenta de la velocidad de modulación.

CAPITULO VIII.

CAPITULO VIII -

RESULTADOS DE LA SIMULACION PARA TRANSMISION CON ECUALIZACION Y ANALISIS COMPARATIVO.

8.1 Resultados de la Ecualización con el Filtro TDL.

Los resultados obtenidos contemplaron la siguiente inicialización de las variables:

```
semilla, sem=50;
velocidad de modulación, baud=1400 bits/s;
exceso de ancho de banda, alfa=0.6;36
frecuencia de portadora, fp=1800;
# de bits tratados, bits=1024;
período de aprendizaje de 1024 bits;37
# de coeficientes, ca=20;
instante de decisión, decis=4;38
coeficiente de convergencia, mu=0.01.
relación señal a ruido, sn=36 dB.
```

Los resultados, que muestran la tremenda mejoría lograda al usar ecualización, se los muestra en la figura 23; fueron

³⁶ Como recomienda el CCITT para transmisión a 1400 b/s, se ha tomado un exceso de banda alfa de .6

³⁷ El período de aprendizaje puede ser cambiado modificando la variable aprende. En la presente corrida aprende=4, por lo que el aprendizaje comprende toda la secuencia para secuencias de hasta 1024 bits. Para secuencias mayores entramos en el período de autoajuste, en donde ya no se actualizan los coeficientes de los filtros sino que se usa los obtenidos en el aprendizaje. En este período el error no se calcula con los valores verdaderos almacenados sino con el valor esperado que previamente ha sido cuantizado.

³⁸ El instante de decisión influye de manera determinante en los resultados. En nuestro caso tenemos 8 muestras por cada pulso y del capítulo 7, al graficar los bits recibidos se ve que la menor distorsión se produce en el centro del pulso, lo que corresponde al bit #4.

obtenidos en la cuarta secuencia de 256 bits de los 1024 bits enviados, desde la iteración 1800 a la 2048 (recordemos que el vector complejo es de dimensión $2048 = 256 * 8$ muestras/bit). En la figura, los pulsos NRZ están normalizados a +1 y -1, lo que determina la amplitud en el eje vertical. Para referencia hemos graficado los bits transmitidos con línea continua delgada.

A continuación, con línea punteada gruesa mostramos los bits a la entrada del ecualizador TDL. En ellos se nota la enorme distorsión introducida por el canal telefónico. Finalmente, con línea gruesa continua se muestra a los bits ecualizados previos a la cuantización.

Hemos usado la velocidad de modulación de 1400 bits/s como un compromiso entre ser capaces de demostrar las propiedades de convergencia de los algoritmos de ecualización, y poderlo hacer para un número adecuado de iteraciones (1024).

Debe notarse que los pulsos recibidos presentan un cierto retraso, lo que físicamente se debe a la extensión del cable telefónico. Esto fue implementado en la simulación definiendo una fase negativa, lo que en el dominio del tiempo se traduce en un retardo.

En la fig. 23. es posible notar que para el bit 14, correspondiente al tiempo 1912-1916, existe error, puesto que con la cuantización, el bit recibido daría +1 debiendo dar -1. Además, para los bits 1, 4 y 13 el nivel de decisión difiere apenas de la referencia 0, por lo que tienen una alta probabilidad de presentar error.

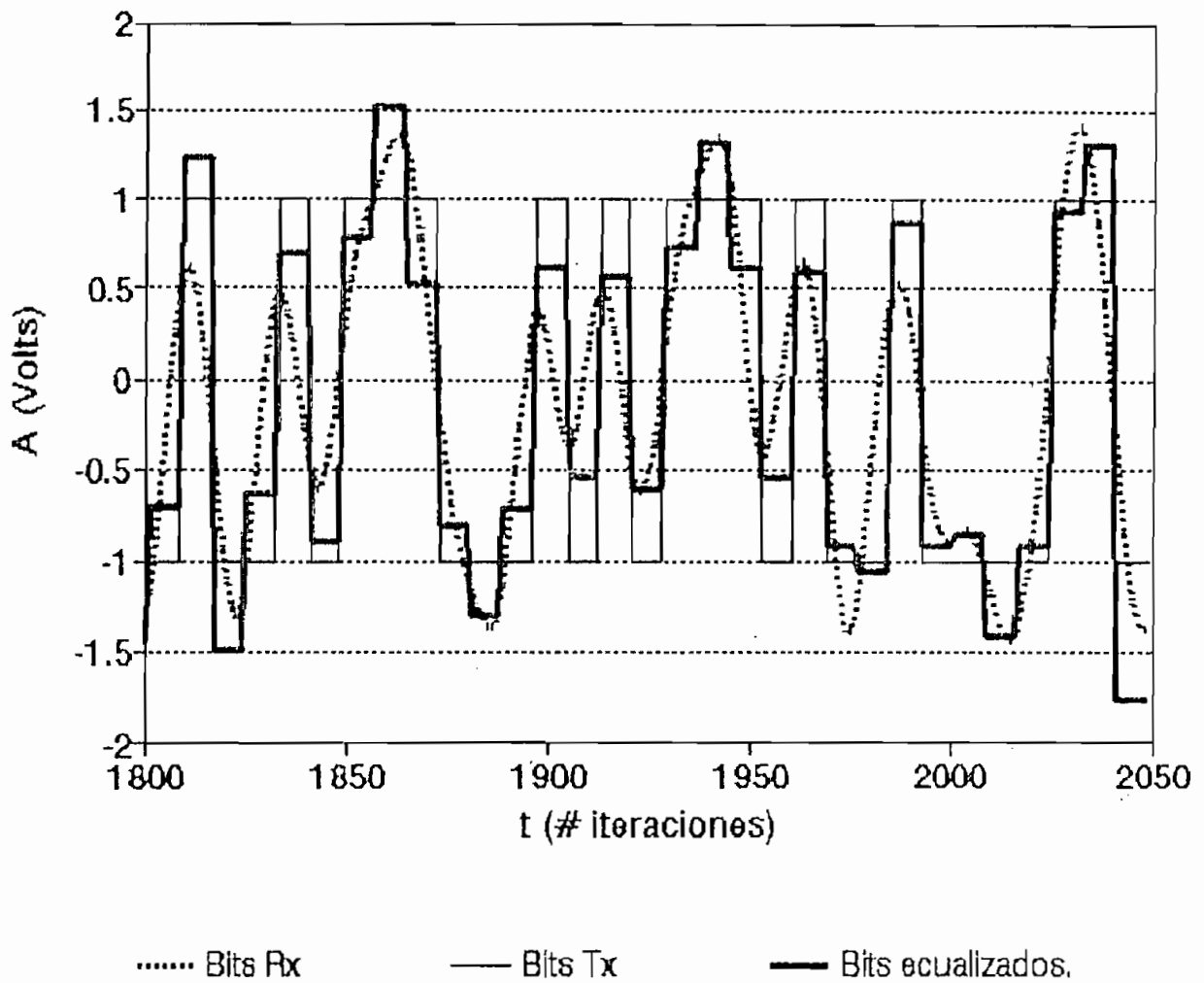


Figura 23 Compendio de Tx sin y con ecualización con el Filtro TDL. A=Amplitud de los bits NRZ; t=# de iteraciones.

8.2 Convergencia del Error Cuadrático y de los coeficientes del Filtro TDL.

Las figuras 24 y 25 nos muestran cómo se produce la convergencia del error cuadrático, así como la convergencia de los taps del ecualizador.

La Convergencia del Error Cuadrático no consiste de otra

cosa que de tomar la diferencia entre el bit almacenado y el bit transmitido y elevarla al cuadrado. Es claro que se produce un error cada vez que el error cuadrático es mayor o igual a uno, (ver lá fig. 24).

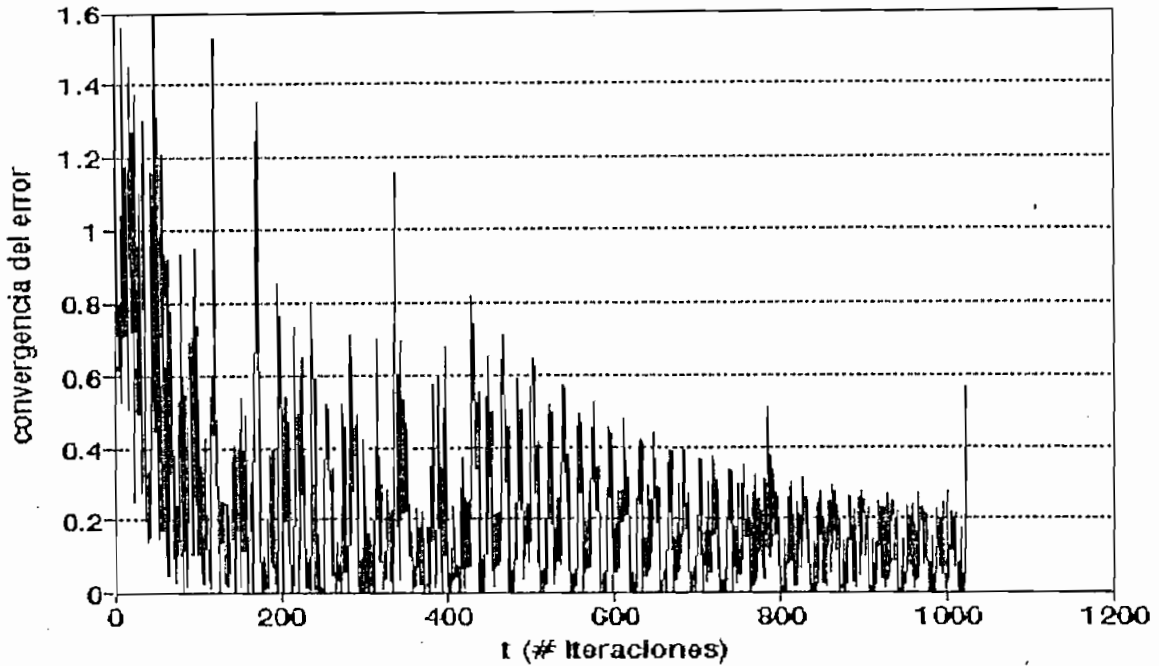


Figura 24 Convergencia del error cuadrático para 1400 bits/s. eje vertical=Magnitud del Error Cuadrático; t=# de iteraciones.

A continuación presentamos la fig. 25, la cual muestra la Convergencia de los coeficientes del Filtro TDL.

Es importante anotar que se observa la convergencia del algoritmo (definida como el crecimiento del coeficiente principal h_1 del 10% al 90%, Schwartz, ob. cit.) en alrededor de 580 iteraciones, lo que será comparado con el resultado obtenido con el Filtro de Malla.

De la figura se puede observar la convergencia de los 7 coeficientes principales, de los cuales hemos remarcado los 2 más importantes.

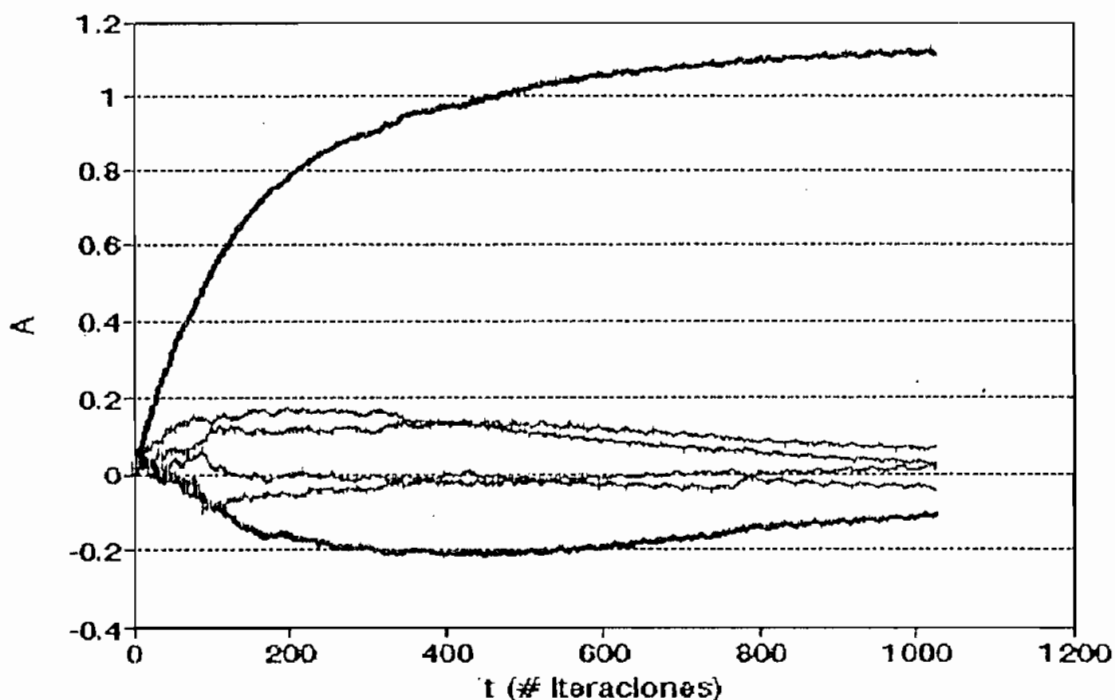


Figura 25 Convergencia de los coeficientes del Filtro TDL para una velocidad de 1400 bits/s y $\mu=0.01$. A=Amplitud de los coeficientes; t=# de iteraciones.

Se debe consignar el hecho de que para iniciar la simulación se encerró todos los coeficientes.

Otra aproximación igualmente valedera, consiste en inicializar el coeficiente central a 1 y el resto a 0. Nosotros no hemos predefinido un coeficiente central, ya que ello implica retardar $(ca-1)/2$, en el presente caso 10 períodos, a los bits recibidos, lo que complica innecesariamente el algoritmo.

8.3 Respuesta Impulsiva del Canal

Como veremos a continuación, nuestra aproximación al asunto presenta una mayor semejanza al comportamiento físico de un canal.

Es decir, si graficáramos los coeficientes obtenidos (fig. 25), una vez que han alcanzado la convergencia, el resultado sería un gráfico que empieza en cero, sube hasta 1.11, baja a -0.11, va a 0.066, continúa a -0.044 y finalmente va a 0.012.

Esto nos recuerda a un pulso que al ser transmitido empieza en cero, sube a su valor máximo, para luego decaer en forma oscilatoria exponencial.

Tal tipo de comportamiento es típico de los canales telefónicos, como lo encontramos en L.J.Millot, A General Class of PAM Equalizers en IEEE Transactions on Communications, June 1980. Gráfico que reproducimos en la fig. 26.

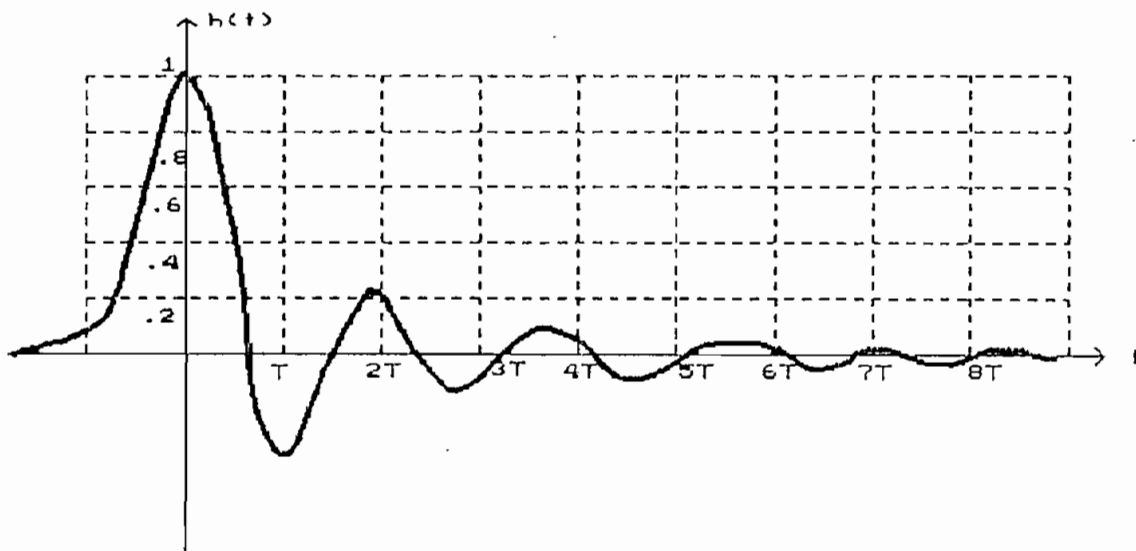


Figura 26 Respuesta impulsiva de un canal telefónico.

Es por este comportamiento físico que en lo sucesivo, para el Filtro de Malla, también usaremos la misma aproximación de inicializar los coeficientes a cero.

8.4 Resultados obtenidos al variar el coeficiente de convergencia μ .

En la figura 27 hemos variado μ de 0.01 a 0.001. En esta figura se observa que no se alcanza la convergencia del error, lo que es claro de la figura 28, ya que se ve que para 1024 bits, los coeficientes apenas si han llegado a la mitad de los correspondientes valores de la fig. 25.

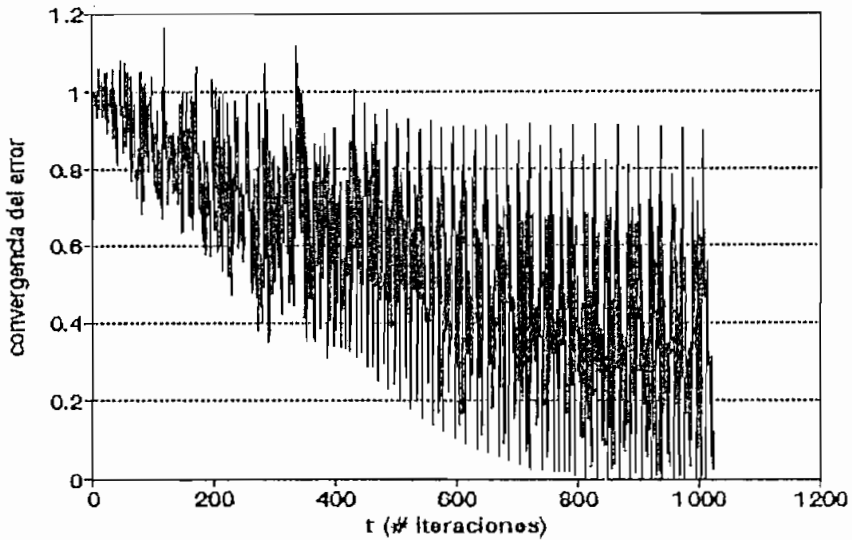


Figura 27 Convergencia del Error Cuadrático para 1400 bits/s y $\mu=0.001$. eje vertical=Amplitud del Error Cuadrático; t=# de Iteraciones.

En la fig. 28., se ve que los coeficientes no oscilan tanto en el proceso de convergencia, con lo que al disminuir μ se aumenta la estabilidad pero disminuye la velocidad de convergencia.

De aquí resulta entonces que el valor recomendado de $\mu=0.01$ es un valor de compromiso entre velocidad de convergencia y valores de los taps en estado estable que aseguren que el algoritmo no diverja.

Si la variación de μ es en el sentido de acelerar la

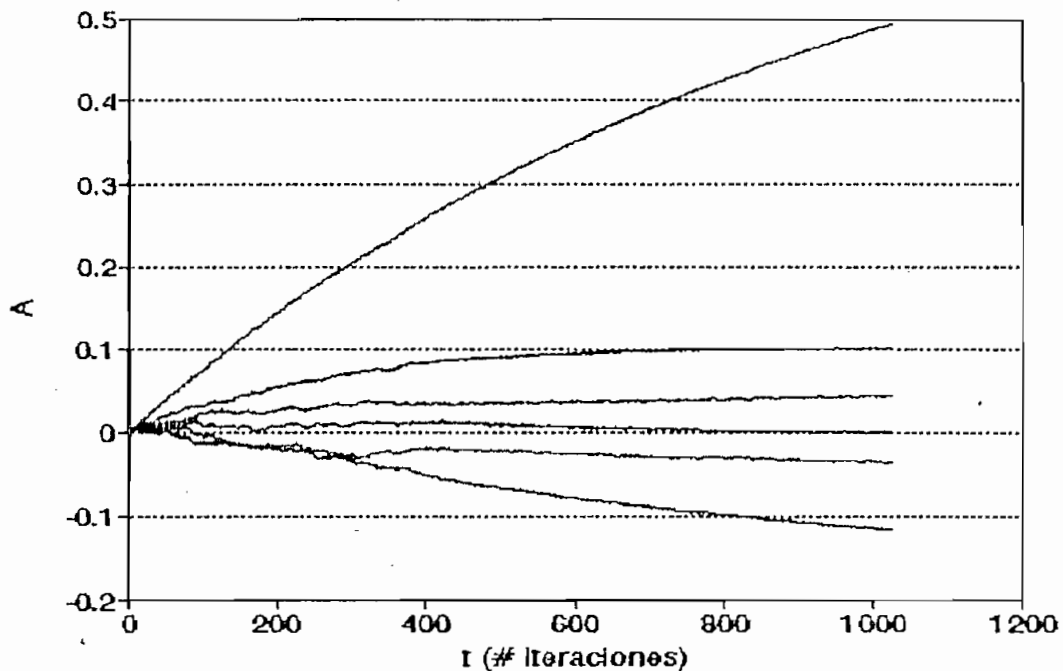


Figura 28 Convergencia de los Coeficientes del Filtro TDL para una velocidad de 1400 bits/s y $\mu=0.001$. A=Amplitud de los Coeficientes; t=# de Iteraciones.

velocidad de convergencia, debe tenerse cuidado de elegir valores cercanos al valor recomendado, por cuanto el incremento excesivo de μ puede generar inestabilidad y divergencia del algoritmo.

8.5 Incidencia del número de coeficientes en el Error Cuadrático.

En el gráfico 29 podemos observar los resultados de convergencia obtenidos cuando al Filtro TDL lo hicimos correr con 3 coeficientes y con el μ recomendado de $.2/3 = .067$.

En la figura se puede observar que el algoritmo converge a un valor muy cercano a 1, por lo que se concluye que el número de coeficientes no es suficiente.

En la figura 30 se muestra el resultado para 100

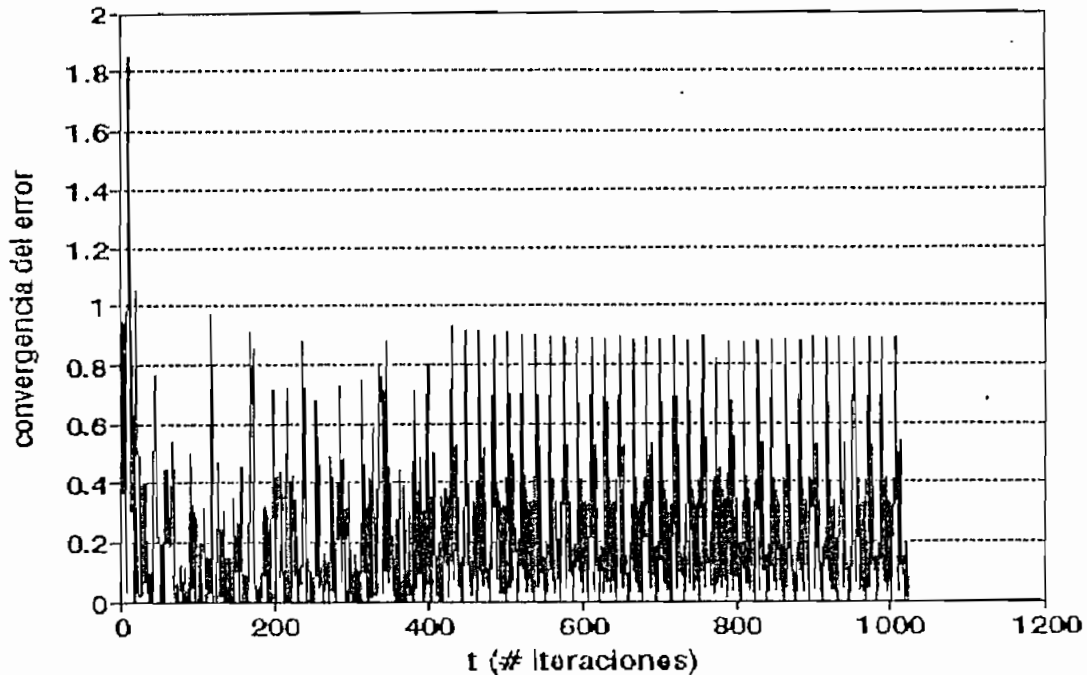


Figura 29 Convergencia del Error Cuadrático para el Filtro TDL con 3 coeficientes. eje vertical=Amplitud del Error Cuadrático; t=# de Iteraciones.

coeficientes y $\mu = .2/100 = .002$. De ella se puede concluir que una vez que se encuentra el número de taps para el cual el algoritmo converge (20 taps), el aumentar el número de coeficientes no conlleva mayor mejora. La decisión respecto al número de coeficientes vendrá dada entonces por la mayor o menor dificultad de implementación práctica.

8.6 Resultados de la Ecuación con el Filtro de Malla.

En lo que sigue trataremos con el Filtro de Malla, sobre el cual realizaremos un estudio similar al precedente, es decir, analizaremos la ecuación obtenida en los bits transmitidos, la convergencia del Error Cuadrático y la convergencia de los Coeficientes.

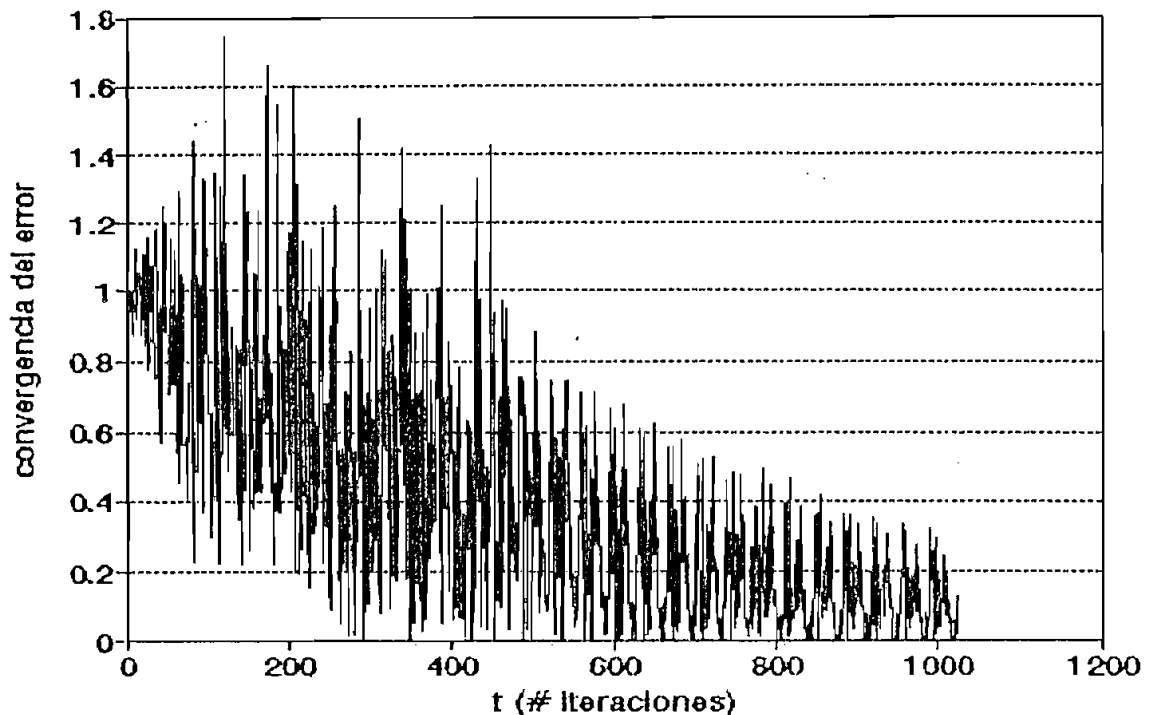


Figura 30 Convergencia del Error del Filtro TDL para $ca=100$. eje vertical=Amplitud del Error Cuadrático; t=# de Iteraciones.

Procederemos a variar los dos parámetros del Filtro de Malla: el coeficiente de inversibilidad de la matriz de covarianza epsilon, (ϵ) y el factor de olvido μ , demostrando como en ³⁹, que el algoritmo es relativamente insensible a la variación de epsilon.

Para la simulación con el Filtro de Malla se inicializó las variables de la siguiente manera:

```
semilla, sem=50;
velocidad de modulación, baud=1400 bits/s;
exceso de ancho de banda, alfa=0.6;
```

³⁹ Satorius and Pack. Application of Least Squares Lattice Algorithms to Adaptive Equalization. IEEE Transactions on Communications, Feb. 1981.

frecuencia de portadora, $f_p=1800$;
de bits tratados, $bits=1024$;
período de aprendizaje de 1024 bits;
de coeficientes, $ca=20$;
instante de decisión, $decis=4$;
coeficiente de inversibilidad, $eps=0.001$;
factor de olvido, $w=1$.

8.7 Convergencia del Error Cuadrático y de los coeficientes del Filtro de Malla.

Al igual que la fig.23, la figura 31 muestra la mejoría obtenida luego de aplicar la ecualización sobre los bits recibidos.

En ella se observa que la ecualización es más uniforme que en el caso del Filtro TDL, fig.23.

Esto se debe a la mayor velocidad de convergencia que presenta el algoritmo de Malla.

A continuación, de las figuras 32 y 33 obtendremos información para determinar la velocidad de convergencia del Filtro de Malla.

Vemos en la figura 32 que el primer coeficiente adquiere el 90% de su valor de estado estable en, aproximadamente, 180 pulsos.

Esto es mejor en casi 3 veces que los 580 del filtro TDL con $\mu=0.01$.

Debemos anotar que aquí se observa también la respuesta oscilatoria exponencial que es típica de un canal telefónico (Millot, ob. cit.). Los coeficientes varían desde 1.127 a -0.017 ,

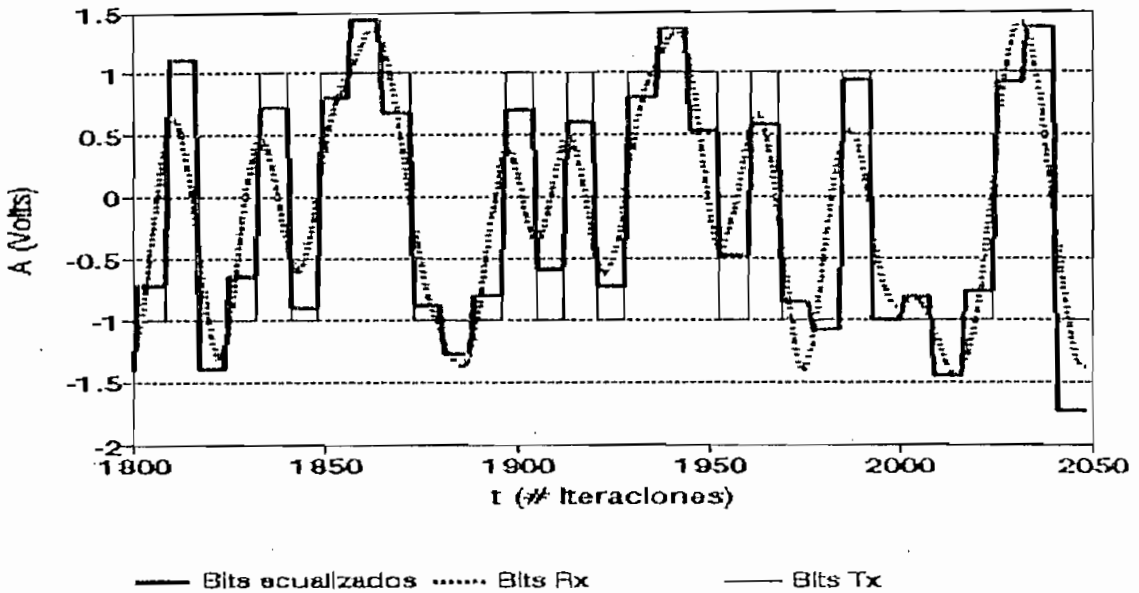


Figura 31 Bits Ecualizados con el Filtro de Malla. A=Amplitud de los bits NRZ; t=# de Iteraciones.

-.12, .104, -.01, -.03 y .03.

En lo que concierne al error en estado estable, hemos hallado una total analogía con el Filtro TDL. Con 3 coeficientes el error en estado estable fue de .88, muy cerca de 1. Con 20 coeficientes se muestra el resultado en la fig. 33, el cual no llega a alcanzar la unidad y siempre es menor que el correspondiente error del filtro de malla. Y con 100 coeficientes si bien existe una mejoría respecto al caso de 20 coeficientes (el error en estado estable es menor que 0.2), sin embargo existe un gran incremento en la complejidad de implementación (100 taps contra 20).

Antes de pasar al análisis del factor de olvido, vale la pena anotar que conforme aumenta el grado de distorsión introducido por el canal, el comportamiento de los dos tipos de filtros tiende a ser comparable. Para corroborar lo dicho hemos corrido el programa para la velocidad de 2400 bits/s, la cual

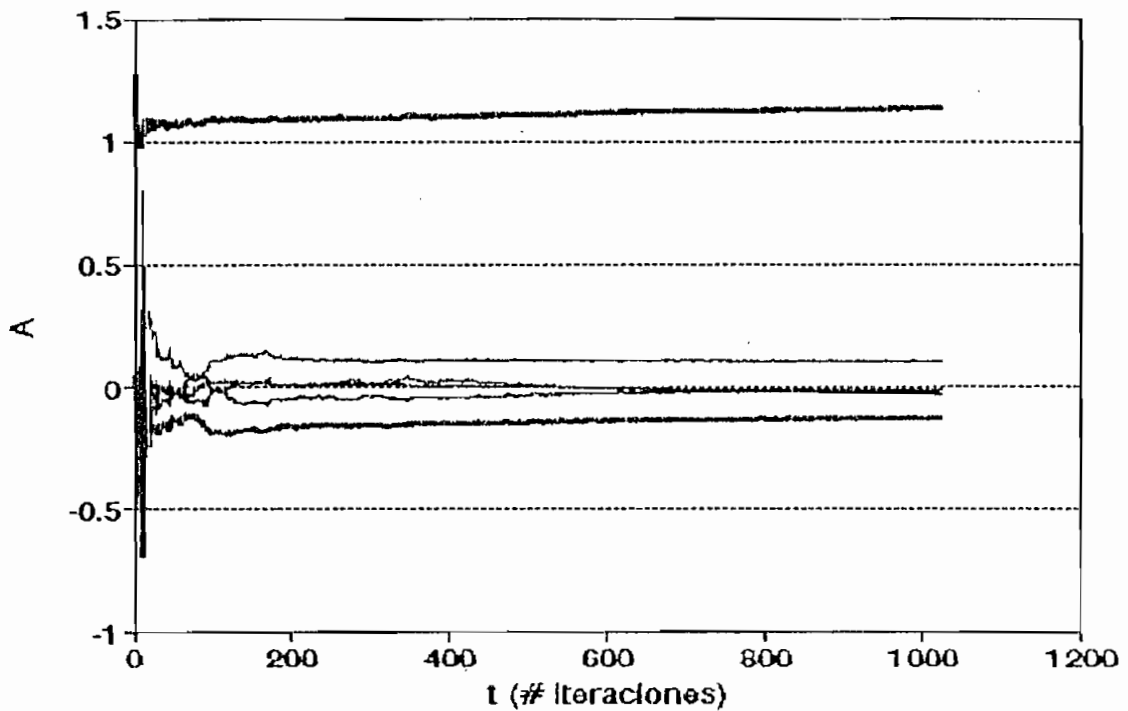


Figura 32 Convergencia de los coeficientes del Filtro de Malla. A=Amplitud de los Coeficientes; t=# de Iteraciones.

constituye el límite actual para la transmisión usando modulación BPSK (Lucky, Schwartz, etc.). Los resultados se muestran en las figuras 34 y 35, en las cuales se ve que los dos filtros convergen aproximadamente en 600, siendo el Filtro de Malla algo superior por cuanto tan solo un coeficiente no da muestras de converger, en tanto que este comportamiento lo poseen dos coeficientes para el caso del Filtro Transversal.

8.8 Incidencia del factor de olvido w sobre la Convergencia del Error Cuadrático.

En la fig. 36 presentamos la corrida del programa COM_CAD (De sus siglas: Computer Aided Design for Communications, nombre que no existe en los paquetes comerciales y con el que hemos

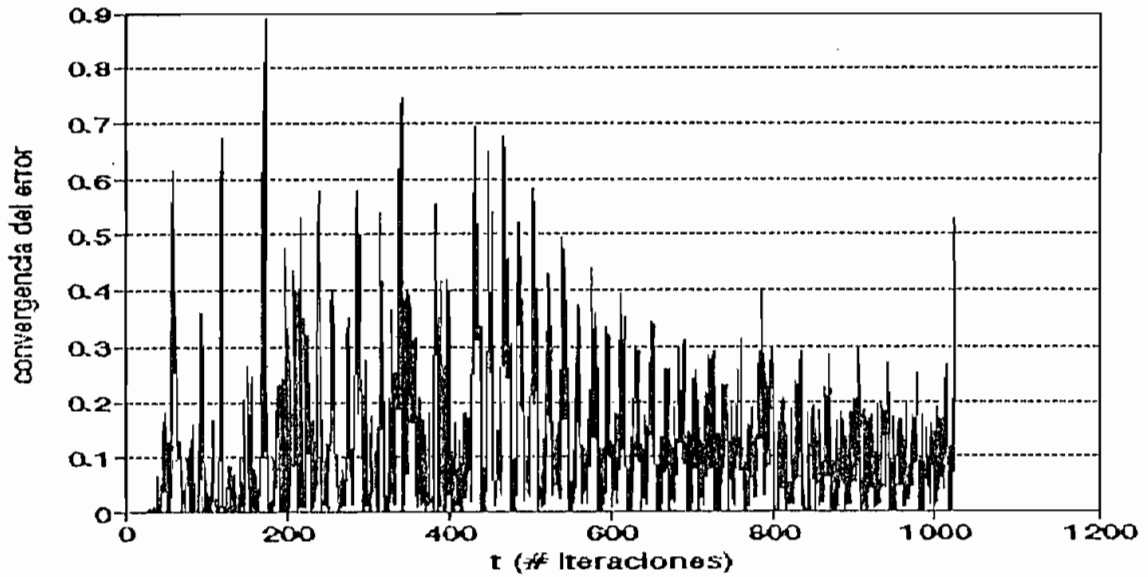


Figura 33 Convergencia del error cuadrático del filtro de Malla para 20 coeficientes, 1400 bits/s, sn = 36 dB, eps=.001. eje vertical = Convergencia del error cuadrático; t=# de Iteraciones.

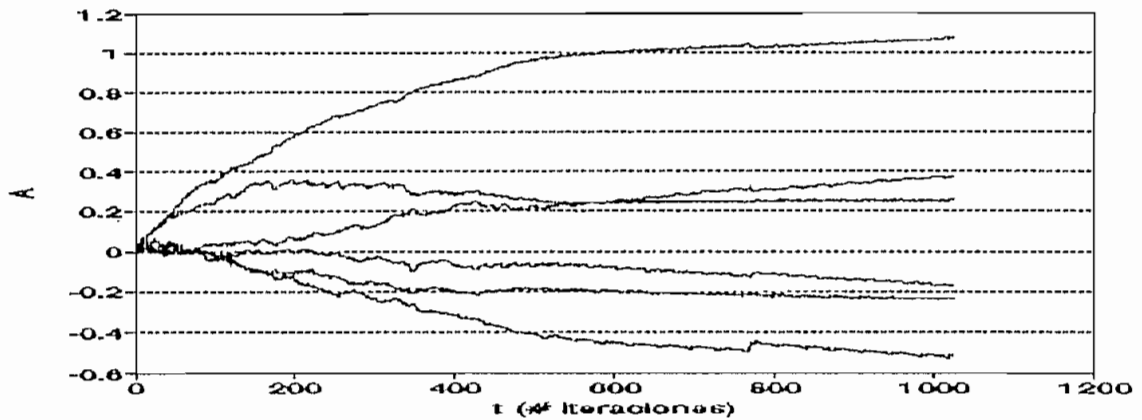


Figura 34 Convergencia de los coeficientes del Filtro Transversal para 2400 bits/s.

bautizado nuestro software), para $w=0.8 < 1$.

Recordemos que el factor de olvido es analizado en la deducción del algoritmo del Filtro de Malla, y se lo emplea para que las muestras más recientes tengan una mayor incidencia en los resultados que las muestras anteriores.

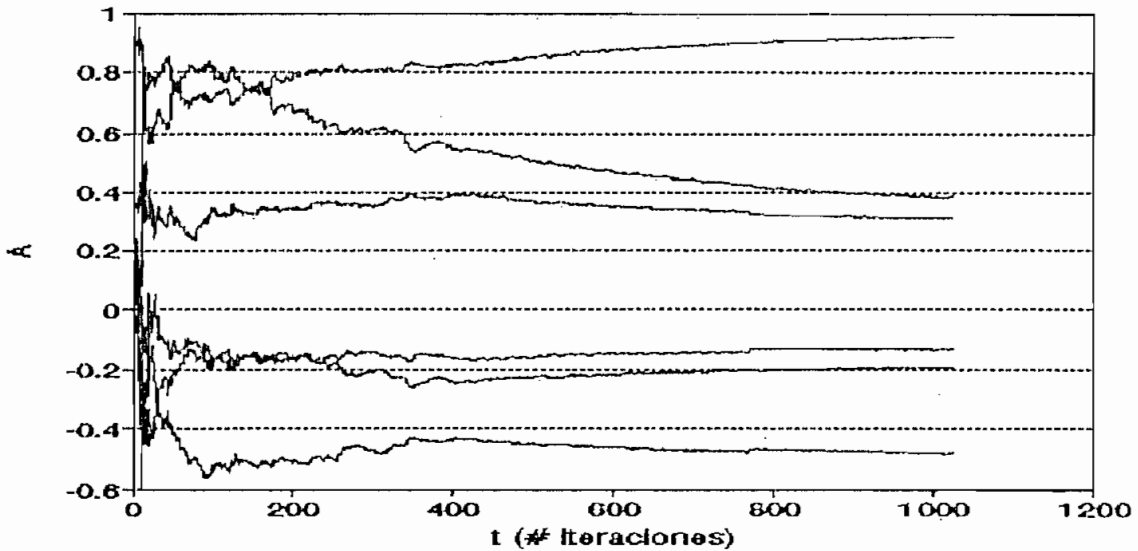


Figura 35 Convergencia de los coeficientes del Filtro de Malla para 2400 bits/s.

De lo visto en la fig. 36, un factor de olvido inferior a la unidad es más deseable para la ecualización, puesto que el error converge en alrededor de .0001.

Qué tan inferior es la pregunta que surge inmediatamente. Para responderla debemos recordar el hecho capital de que los resultados vistos los hemos obtenido en el período de aprendizaje.

Para ver qué tan pequeño puede ser w , hemos desconectado el aprendizaje luego de 768 bits, es decir 3 secuencias de entrenamiento.

Para el efecto usaremos la mencionada velocidad de transmisión a 1400 bits/s con $\alpha=0.6$.

8.9 Resultados obtenidos en el período de autoajuste.

Con el objeto de determinar qué tan pequeño puede ser w por

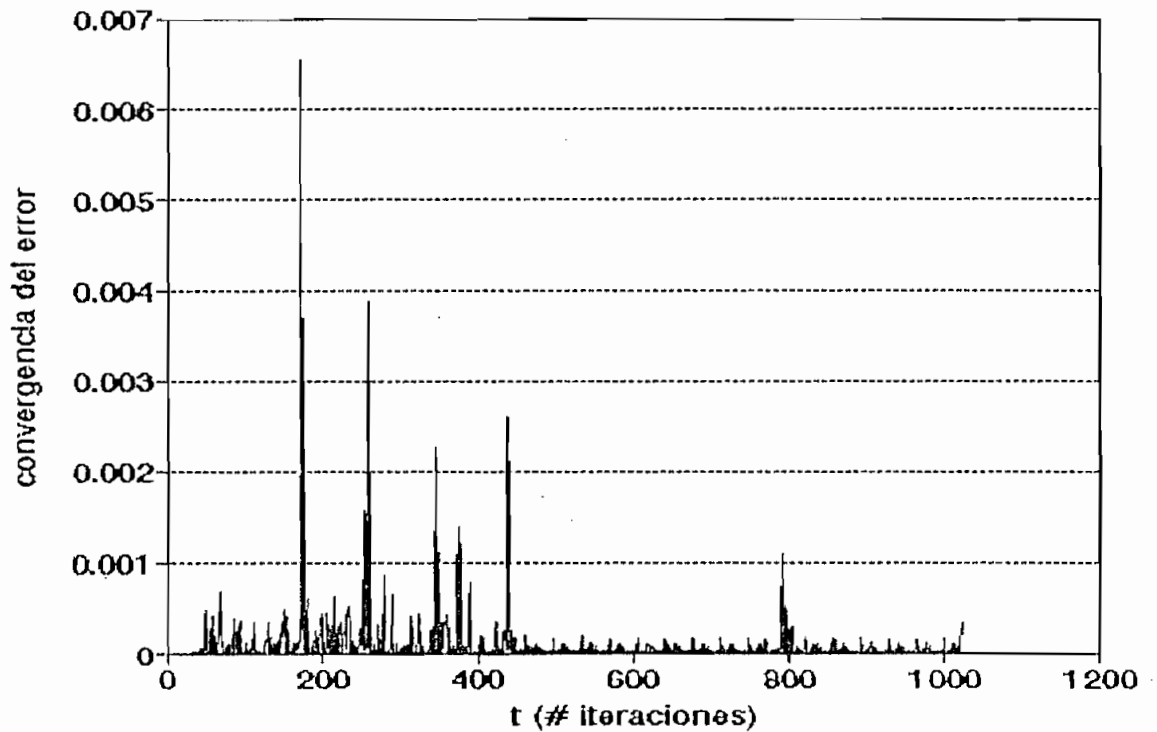


Figura 36 Convergencia del Error Cuadrático para el Filtro de Malla con $w=0.8$ y 1400 bits/s. eje vertical=Amplitud del Error Cuadrático; t=# de Iteraciones.

debajo de 0.8 efectuamos la simulación con $w=0.3$.

La ecualización no funcionó.

En el período de aprendizaje el error era tan pequeño como 10^{-20} , lo que en principio nos hizo pensar que funcionaría; sin embargo, al graficar los coeficientes se vio que estos se disparaban hasta 0.9 en el estado estable (una relación de casi $10^{20} : 1$).

8.10 Límite inferior para el factor de olvido w .

Adicionales simulaciones mostraron que el límite inferior en

lo que respecta a w , viene dado por 0.9, porque en ese caso la relación entre los valores en el aprendizaje y el entrenamiento es de 6:1, lo cual es razonable. El resultado se muestra en la fig. 37.

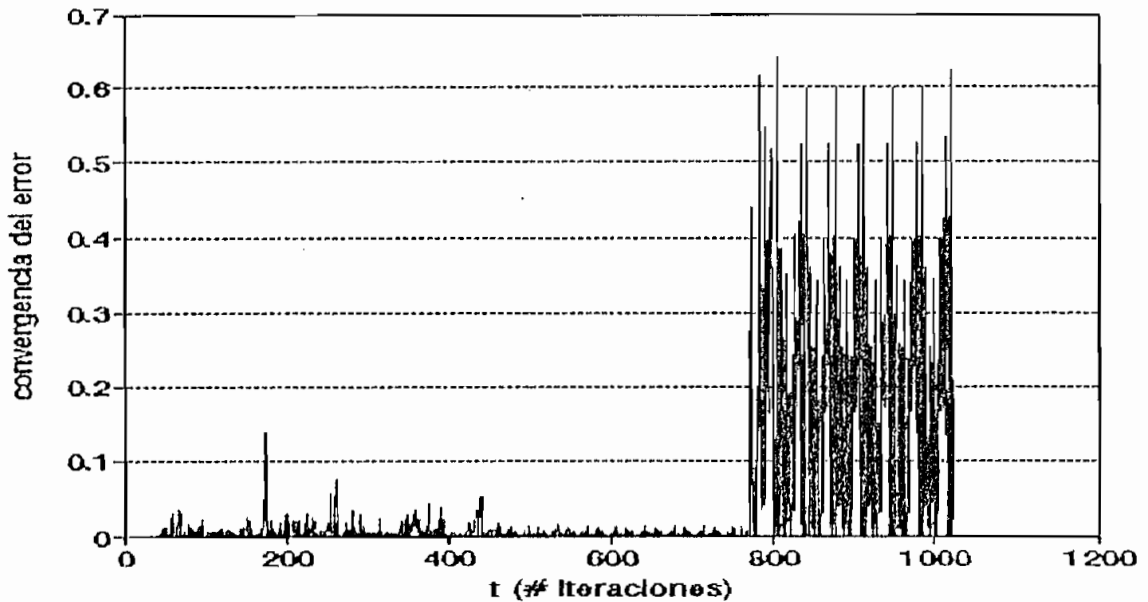


Figura 37 Convergencia del error del Filtro de Malla para $w=0.9$ y 1400 bits/s.

De todo ello concluimos que el valor óptimo de w está entre 0.9 y 1, puesto que para $w=1$ la relación entre los valores en el aprendizaje y el entrenamiento es de alrededor de 1. Los resultados para $w = 1$ se muestran en la fig. 38 (para una secuencia de entrenamiento de 768 bits).

Finalmente, hemos comprobado, como establecen Satorius y Pack (ob. cit.), que el algoritmo de Malla es insensible a la variación de ϵ o factor de inversibilidad de la matriz de covarianza.

Las simulaciones realizadas demostraron que la diferencia entre el Error Cuadrático obtenido con $\epsilon=0.001$ y aquel error obtenido con $\epsilon=0.005$ no superaba el 1/10000.

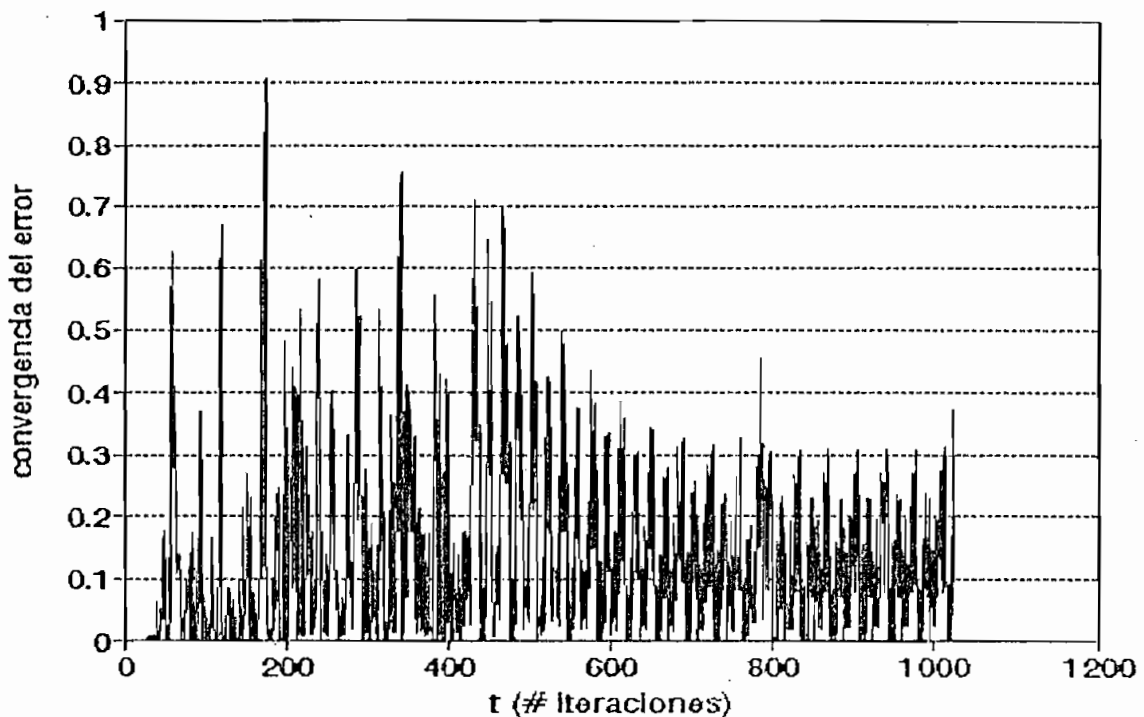


Figura 38 Convergencia del error del Filtro de Malla para $w=1$ y un período de entrenamiento de 768 bits.

Adicionalmente anotemos que la simulación realizada con una relación señal a ruido de 10 dB determinó que el error en estado estable sea mayor que en el caso de $SNR = 36$ dB.

Este comportamiento se muestra en las figuras 39 y 40 para los filtros TDL y de Malla (compararlo con las figuras 24 y 33). En la transmisión sin ecualización se observó que los errores crecen interminablemente, en tanto que con ecualización no se producen más errores después del período de aprendizaje, de lo que concluimos que la ecualización es capaz de controlar la producción de errores en presencia de Interferencia Intersímbolo y Ruido Aditivo.

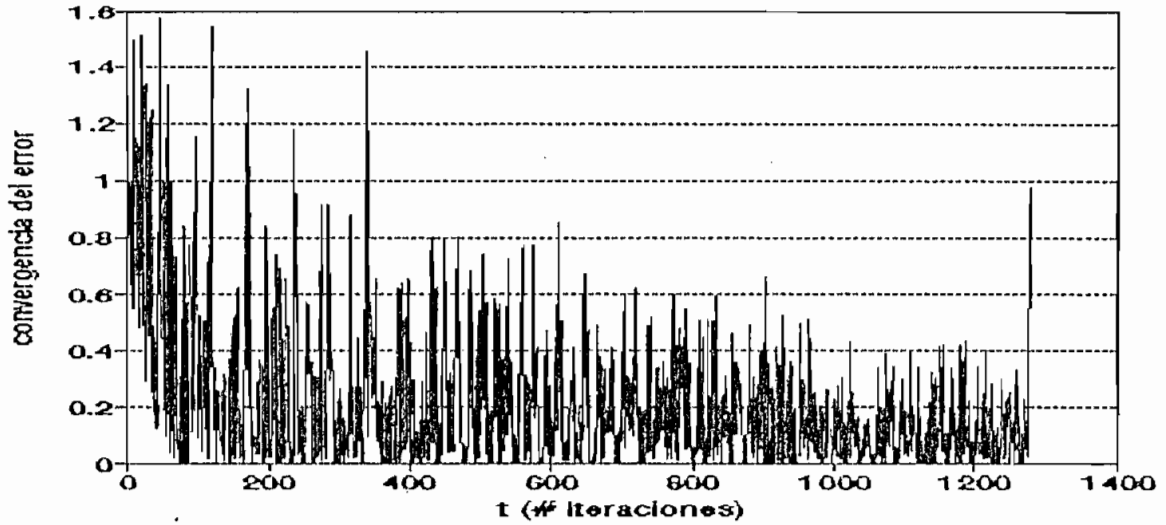


Figura 39 Convergencia del error del Filtro TDL para una SNR de 10 dB., baud = 1400 bits/s y alfa = .6.

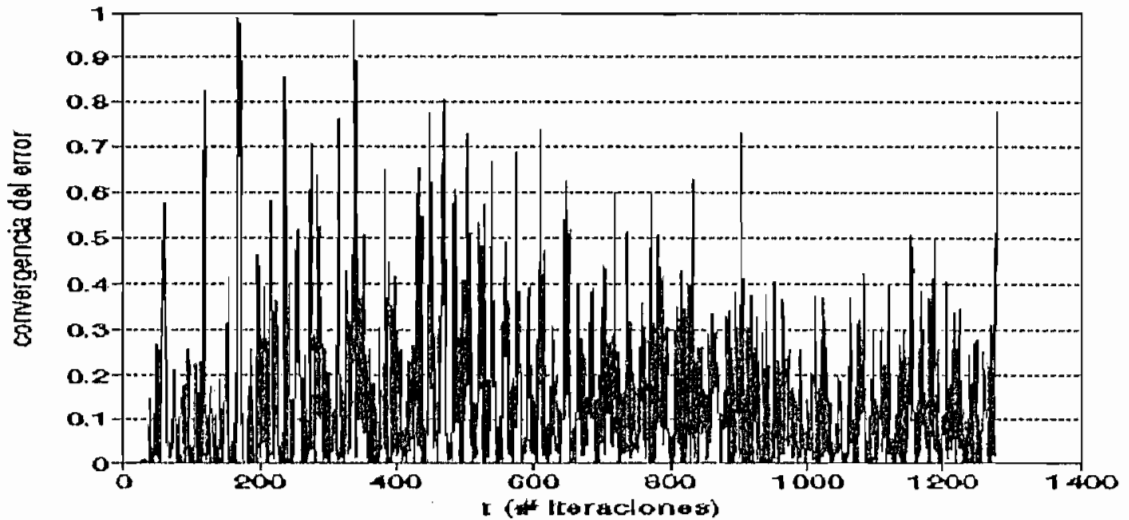


Figura 40 Convergencia del error para el Filtro de Malla. SNR=10, 1400 bits/s, alfa=.6.

8.11 Análisis comparativo de los Filtros TDL y de Malla.

A continuación procedemos a resumir las características más importantes de uno y otro filtro en forma comparativa.

1.- El Filtro TDL se caracteriza por su simplicidad en lo que concierne al algoritmo de implementación así como al análisis matemático.

2.- El Filtro de Malla es más complejo en cuanto al algoritmo y al análisis matemático, pero esto determina un aumento en la velocidad de convergencia respecto del Filtro TDL.

3.- La entrada al Filtro TDL es la señal $y(i)$, la cual es no ortogonal, en tanto que la entrada a la "parte tipo TDL" del Filtro de Malla la constituye la señal de error reverso $r(n,i)$, la cual es ortogonal.

4.- En el Filtro de Malla los eigenvalores de la matriz de covarianza son iguales, lo que determina un aumento en la velocidad de convergencia.

5.- En las simulaciones llevadas a cabo hemos hallado que la velocidad de convergencia del Filtro de Malla es, para las mismas condiciones de convergencia de los coeficientes en estado estable (figs. 25 y 32), de alrededor de 3 (580/180) veces la del Filtro TDL. Honig⁴⁰ ha hallado que el aumento en la velocidad de convergencia es de alrededor de 4 veces, por lo que nuestro resultado está dentro de un rango aceptable.

6.- El error en estado estable del Filtro de Malla es siempre menor, en una relación de aproximadamente 2 respecto del Filtro Transversal. (Ver las figuras 24 y 33 en donde el valor residual del Filtro TDL se ve que es .25 y el del Filtro de Malla, .15).

7.- El Filtro de Malla es recursivo en el orden y en el tiempo, de modo de que si se necesita un filtro de mayor orden no se

⁴⁰ Honig. Echo cancellation of voiceband data signals. IEEE Transactions on Communications, Jan. 1985.

precisa minimizar el error para hallar todo un nuevo conjunto de coeficientes, sino que el nuevo coeficiente es hallado en función de los calculados previamente. En el caso del Filtro TDL es preciso hallar un nuevo conjunto de coeficientes.

8.- Experimentalmente se halló que el Filtro de Malla se demora alrededor de 4 veces en una corrida de 256 bits respecto de lo que se demora el Filtro TDL. De esta forma, la mejora en el número de bits necesarios para converger se paga con tiempo de computación.

8.12 CONCLUSIONES GENERALES.

1.- La presente tesis nos ha permitido incursionar en el ámbito de la transmisión de datos con ecualización, en donde a través de los algoritmos del Filtro TDL y del Filtro de Malla hemos podido efectuar el análisis matemático de velocidad de convergencia, incidencia de los parámetros: μ , ϵ , w y número de coeficientes sobre los bits recibidos, con el fin de dilucidar cuál de los dos Algoritmos es mejor.

2.- Como premisa para tal análisis hemos debido modelizar el canal telefónico con las recomendaciones del CCITT y hemos tenido que investigar la mejor configuración de los filtros transmisor y receptor con el objeto de minimizar la Interferencia Inter Símbolo.

3.- Usando un modelo en banda base para el sistema en la banda de paso, se pueden usar todos los resultados desarrollados para la banda base, los cuales pueden de esta forma ser inmediatamente transferidos a la banda de paso.

4.- Hemos hallado que la característica del canal que asegura transmisión libre de ISI es la de caída senoidal, en donde el

parámetro alfa nos permite disminuir la sensibilidad a la sincronización aumentando el ancho de banda de transmisión.

5.- Del análisis de maximización de la señal al ruido se halló el interesante resultado de que la SNR depende únicamente de la energía de la señal y de la densidad espectral del ruido blanco. De esta forma dos señales de formas de onda diferentes pueden proporcionar la misma probabilidad de error en presencia de ruido blanco siempre que tengan la misma energía y que estén filtradas por filtros acoplados.

6.- Hemos comprobado que la velocidad de transmisión en la actualidad, usando ecualización, está alrededor de 2400 bits/s para transmisión BPSK. El resultado obtenido fue de 2400 bits/s para un exceso de banda alfa de 0.125, con 20 coeficientes, tanto para el Filtro TDL ($\mu=0.01$) como para el Filtro de Malla ($w=1$, $\epsilon=0.001$). Debe observarse que el error en estado estable es cada vez mayor conforme se aumenta la velocidad, pudiendo existir problemas si por una u otra razón existen cambios grandes en el canal.

7.- Uno de los resultados centrales de la tesis consiste el haber establecido que la velocidad de convergencia del algoritmo del Filtro TDL depende de la relación del máximo eigenvalor al mínimo eigenvalor de la matriz de covarianza.

8.- Como otro resultado capital tenemos el haber podido demostrar que la ortogonalización de los datos de entrada al ecualizador de Malla determina que la matriz de covarianza sea diagonal, lo que, por la conclusión anterior, acelera la convergencia.

9.- Los resultados anteriores en lo que respecta al análisis de convergencia, no hubieran sido posibles sin previamente haber profundizado en la teoría del Algebra Matricial. Por esta razón hemos hecho un estudio de dicha teoría, el cual se consigna en el

Apéndice IV. De igual forma, el análisis de los procesos estocásticos estacionarios (ergódicos), a través del valor esperado; y del ruido, a través de la autocorrelación y la Transformada de Fourier, se ha basado en la teoría que se presenta en el Apéndice III.

10.- Se ha podido establecer la conexión entre el proceso de actualización de los coeficientes de reflexión del Filtro de Malla, así como la actualización de los errores directo y reverso, con el proceso de ortogonalización de Gram-Schmidt.

11.- Hemos comprobado experimentalmente, a través de simulaciones, que para procesos estacionarios, la energía de predicción del error directo es igual a la del error reverso.

12.- Hemos sido capaces de explicar el por qué de la recomendación del CCITT de usar una portadora a 1800 Hz. en base a las curvas de distorsión dadas por el mismo organismo.

13.- Finalmente, a través de los gráficos de bits transmitidos, recibidos y ecualizados, hemos demostrado que la teoría de los ecualizadores, considerados como el ajuste de parámetros a través del uso de los mínimos cuadrados, realmente conduce a aumentar la velocidad de transmisión, manteniendo una tasa de error casi nula. Esto permite el mayor aprovechamiento de los canales telefónicos y el consiguiente ahorro al poder usar líneas normales, a las que se les ha dado el tratamiento de ecualización, para la transmisión a velocidades cada vez mayores.

14.- En el principio del apéndice IV, consideramos a la ecualización como la búsqueda de algoritmos que permitan invertir eficientemente la matriz de covarianza del sistema. Por tanto, el desarrollo futuro de los ecualizadores no consiste de otra cosa que de buscar nuevos métodos de inversión de matrices en forma

óptima. Por otra parte, la corriente actual en los reportes de telecomunicaciones consiste en hallar formas de analizar la convergencia de los Filtros de Malla, los cuales son altamente no lineales.

Además, del acápite 1.5, sabemos que la investigación actual en lo que respecta a ecualizadores, contempla el uso de varios códigos, entre ellos el de Trellis, para alcanzar velocidades cada vez mayores.

Con todo esto, el campo de la ecualización tiene mucho que ofrecer para futuras investigaciones. El afán de la presente tesis ha sido contribuir a esta corriente de investigación de ecualización con un modelo de análisis matemático, dado por la Predicción Lineal y la Teoría de Algoritmos que minimizan el Error Cuadrático. Modelo que ha sido implementado en Lenguaje C++, de modo que la comprobación de lo establecido en la tesis sea inmediata.

El listado del programa que se muestra en el Apéndice II, revelará que es posible generalizar la investigación a otro tipo de canales; usar diferentes configuraciones de filtros; usar una modulación distinta; implementar otro tipo de ecualizador, etc; lo cual se puede lograr modificando las correspondientes subrutinas.

Con ello se deja abierta la posibilidad de futuros análisis de convergencia para distintos tipos de filtros y distintos canales de transmisión.

ANEXOS .

APENDICE I .

APENDICE I

EL LENGUAJE TURBO C++ Y EL MENU DE INGRESO.

Para el efecto de la simulación se ha usado el lenguaje Turbo C++ v. 1.0 de Borland, debido a que permite el uso de números complejos de dimensión de hasta 3500.

Puesto que para nuestro caso $\text{dim} = 2050$, la capacidad del lenguaje es suficiente. Además este lenguaje es un lenguaje que nació en el campo de las telecomunicaciones, a saber, lo inventó Bjarne Stroustrup en los Bell Laboratories (ref: Getting Started, Borland 1990.)¹ para simular un sistema de telefonía.

C++ es un lenguaje en cuya base está la creación de ciertas estructuras comprendidas por funciones y datos, llamadas clases, a partir de las cuales, por herencia, se puede generar otras estructuras más complicadas. Esto es de suma utilidad cuando se programa a una escala enorme ya que entre otras cosas permite un ahorro sustancial de memoria, a la vez que evita que los datos sean cambiados indiscriminadamente (ver ref. citada).

Sin embargo, en nuestra aplicación usaremos el lenguaje Turbo C++ a un nivel llamémoslo clásico, debido a que no usamos estructuras para las funciones sino que usamos dichas funciones en el sentido del Quick Basic ó del Fortran. Esto determina que nuestros datos no sean protegidos y que cualquier persona con acceso a ellos puede cambiarlos; además de que el código fuente aumenta respecto a la programación con clases.

Si debemos detallar algo del Turbo C++ nos referiremos a su extensa disponibilidad de librerías. Por ejemplo, la manipulación de complejos no es algo intrínseco al lenguaje, el cual en

¹ Turbo C++. Getting Started. Borland International, INC. 1990. v. 1.0

realidad posee un reducido número de palabras clave (tokens). Sin embargo, en base a las estructuras anteriormente mencionadas se puede crear complejos, los cuales vienen en la librería `complex.h`.

Al igual que en el QB o en el Fortran, usamos lazos (`for`) para las repeticiones, (`if`) para las condiciones a ser cumplidas, e inicialización de las variables previo su uso.

A diferencia de los lenguajes clásicos mencionados, el C++ precisa englobar el conjunto de órdenes que comprende la rutina principal en llaves {}, así como también se necesitan llaves en todo conjunto de órdenes que sigue a los lazos (`for`) y a los condicionales (`if`), empero si sólo hay una orden a continuación de `for` ó `if`, no se precisa llaves.

La rutina principal, así como las subrutinas, tienen la posibilidad de retornar valores del tipo que precede a la función en su declaración; e.d. si tenemos la siguiente declaración:

```
int main()  
{  
    .  
    .  
    .  
  
    return 0  
}
```

la orden `return 0` se encarga de hacer cumplir el estamento de que la función `main()` retorna un valor entero (`int` como está en el párrafo anterior). Sin embargo para la función genera:

```

void genera(complex y[], float baud, float fp, int b)
{
    .
    .
    .
}

```

no existe ningún valor de retorno, ni siquiera la orden return. Esto se debe a que "genera" trabaja con los vectores en memoria, razón por la cual no necesitamos que retorne nada (void = nada).

El manejo de vectores complejos de N dimensiones, es sencillo con C++. Esto se ve en la función genera, en donde complex y[] determina que automáticamente se pasen de la función main a la subrutina un arreglo de vectores complejos, de dimensión 2050 en nuestro caso.

Más sobre el lenguaje lo mencionaremos a medida que lo necesitemos.

Una referencia especialmente útil la constituye The C Programming Language, Second Edition, de Kernighan and Ritchie, editado por Prentice Hall 1988.

III.1 Optimización del Programa

Fue preciso rehusar al vector complejo y[] como elemento de paso de resultados a la subrutina de almacenamiento en archivo. Esto fue posible ya que temporalmente no son necesarios los contenidos de y[]. Esto se debió a que cada definición de variables consume memoria, y puesto que estamos trabajando con la memoria convencional de 640 K (en el tiempo de compilación llegamos a valores bajos como 36 K), es preciso efectuar optimizaciones en donde sea posible.

Antes de efectuar la optimización tuvimos problemas como el siguiente: con dos vectores que empezaban con la misma letra y eran de igual longitud, ocurrió que inexplicablemente el dato pa[7] se convertía en el dato p[7], lo cual detectamos después de una prolija búsqueda. Solucionamos el problema dimensionando pa y p con dim=8 y usando solo hasta dim=7. Pero esto hace notar que el manejador de memoria del Lenguaje C estaba en su límite.

Tuvimos además problemas en tiempo de linking. Estos problemas no se deben a la cantidad de memoria disponible sino al número de líneas de lenguaje fuente.

Para solucionar ello se debió usar las subrutinas filtro y fft de manera óptima. Es decir, la subrutina filtro (ftr) realiza el filtrado de Tx y Rx dependiendo del valor de los parámetros. La subrutina fft por su parte, realiza la Transformada de Fourier Directa e Inversa, dependiendo del tipo de parámetros del argumento.

Además se llevó a efecto una prolija revisión de partes del código que se repetían de modo de ponerlas como subrutinas (p.ej: subrutina void decide()) y a que eliminemos rutinas de biblioteca que inicialmente las habíamos puesto para ciertas operaciones como obtener la inicialización de variables string\$ desde el teclado. Esta inicialización fue realizada a través de un programa independiente, el cual constituye el Menú introductorio, como lo discutiremos posteriormente.

Regresando al Programa Principal, puesto que la dimensión de y[] es de 2050, esto limita la graficación de la convergencia de los coeficientes a los 7 primeros coeficientes, que son los más representativos.

Puesto que son 7 coeficientes y digamos que queremos usar una secuencia de 1024 bits para el entrenamiento, estamos hablan-

do de 7168 elementos, los cuales no los soporta y[2048]. Hemos salido del problema efectuando la escritura en el diskette b: en modo append; de manera que en cada pasada de 256 bits, del vector y[] se usan 1792 valores, que está dentro de su capacidad.

En modo append, el archivo debe estar inicialmente vacío, ya que en esta modalidad no se borran los datos anteriores sino que se agrega los nuevos datos al final.

Es por esta razón que siempre que se inicia el programa borramos el archivo correspondiente con la orden remove.

En lo que respecta a la graficación misma, cada gráfico supuso una escritura diferente en el vector y[], como se puede notar en el lenguaje fuente incluido en el Apéndice II.

III.2 La generación de números aleatorios en Turbo C++.

Tal generación no es del todo satisfactoria, puesto que los valores aleatorios generados están únicamente en el rango de 1 a 32767.

Esto determina que exista una asimetría entre +1s y -1s, lo cual se puede observar al imprimir la evolución en pantalla de la generación de números aleatorios, en donde la asimetría está en un orden de 10 a 12%.

A pesar de ello, los resultados obtenidos están dentro del orden de magnitud de los similares resultados que hemos visto en la literatura de telecomunicaciones usada, como se anota en donde es pertinente hacerlo.

APENDICE II.

APENDICE II -

LISTADO DEL PROGRAMA EN LENGUAJE C++.

El presente programa ha sido desarrollado en Lenguaje C++ de Borland, versión 1.0. Es necesario por tanto tener un PC con al menos 7 Megas de Disco Duro disponible para poder instalar el Lenguaje. En el Apéndice I se cita la bibliografía usada para desarrollar el programa.

Como se hace notorio en dicho Apéndice, si bien el programa puede ser corrido desde una XT, el tiempo que se tarda el computador para efectuar la Transformada Rápida de Fourier es de aproximadamente 3 minutos.

Es por esto que se precisa al menos de una máquina con una velocidad de 25 MHz, en la cual el tiempo de la FFT se reduce a 30 seg.

La estructura del programa es modular, es decir que para facilidad de manejo y de comprensión se lo ha dividido en subrutinas. El programa principal se halla constituido por la transmisión sin ecualización, y con ecualización usando el Filtro TDL y el Filtro de Malla.

Se ha procurado que las variables utilizadas sean las mismas que fueron usadas en la teoría.

A continuación presentamos el listado del Programa.

```

/*      PROGRAMA PARA TRANSMISION CON Y SIN ECUALIZACION.
   PARA ECUALIZACION SE USA EL FILTRO TDL Y EL FILTRO DE MALLA. */

#include <stdlib.h>
#include <stdio.h>
#include <complex.h>
#include <iostream.h>
#include <conio.h>

#define NM 2048+1    //Dimensión máxima del vector complejo y[].
#define N 256+1     //Número máximo de bits de la secuencia.
#define CO 101      //Número máximo de coeficientes.
#define CO_G 7      //Número de coeficientes a ser graficados.
#define MUESTRAS 8  //Número de muestras por bit

int main(void)
{

clrscr();

int i=0,j,k,n,o,oo=0,b=N-1;
int tdl, malla; //Variables para la evolución en pantalla.
int evol_g;     //Evolución de genera.
int graf, tipo_g;
char *file="b:\\data.txt";//Puntero al archivo usado.
int entrena;    //# de secuencias de b bits para el
                //entrenamiento.

complex y[NM];
double y_r[N], b_r[N], h_r[CO-1], ye_r[CO-1];
double a_r=0,ad_r,e_r;
double a[CO], g[CO], f[CO], r[CO], c[CO], kf[CO], kb[CO],
        kc[CO], p[CO], pc[CO], sf[CO], sb[CO];
double ga[CO], ra[CO], pa[CO], pca[CO], sba[CO], sfa[1],aca;

FILE *textfile; // Puntero al archivo inicia.dat
char line[15];  // Guarda las líneas leídas

if ((textfile = fopen("b:\\inicia.dat", "r")) == NULL) {
    printf("Error opening text file for reading\n");
    exit(0);
} //Carga las variables desde b:\inicia.dat

while ((fscanf(textfile, "%s", line) != EOF))
{
    j=i++;
    b_r[j]=atof(line);
}

fclose(textfile);

```

```

double fp, alfa, err_t_a=0, bit_t, pe_a, mu, eps, w, sn;
int bits, cuanto, decis, ca;
int l;
double baud;
double sem;
int transm;

transm=b_r[0];
sem=b_r[1];
baud=b_r[2];
alfa=b_r[3];
fp=b_r[4];
decis=b_r[5];
bits=b_r[6];
if (transm==1) evol_g=b_r[7];
tdl=malla=b_r[7];
sn=b_r[8];
entrena=b_r[9];
ca=b_r[10];
mu=eps=b_r[11];
w=b_r[12];
graf=b_r[13];

if (remove(file) == 0 ){ //Borra b:data.txt para escritura
                        //en modo append.
  cout<<("\ndata.txt ha sido borrado debido a que se usa");
  cout<<(" en modo append\n");
}

cuanto = bits/b; // Múltiplo de b bits.

int NN=NM-1;
int N1;          //variable usada para generar data.txt
int t_r;        //variable usada en la subrutina filtro para
                //decidir si es filtro de Tx (0) o de Rx (1)

//Declaración de Funciones
void genera(complex y[], double b_r[], double &sem,
            int evol_g, int NN);
//double &sem cambia el valor de sem por referencia
void fft(complex y[], double d_i, int NN);
void ftr(complex y[], double baud, double alfa, int b,
          int t_r, int NN);
void canal(complex y[], double baud, double fp, int b,
            int NN);
void noise(complex y[], double sem, double sn, int NN);
void intro(complex y[], double y_r[], int NN, int decis,
            int evol_g);
void archivo(complex y[], int N1, int tipo_g, int l,
              int ca,char *file, int b);
void decide(double a1, double &a2);

```

```

for (l=1 ; l<=cuanto; l++) //secuencias de 256 bits.
{
    cout << "\n\nResumen de Variables: ";
    cout << "\n\nsemilla= " << sem << "\nfp= " << fp;
    cout << "\nbaud= " << baud << "\ndecis= " << decis;
    cout << "\nexceso de banda= " << alfa;
    cout << "\n# bits a ser tratados= " << bits;
    if(transm!=1)
    {cout << "\n# de secuencias de entrenamiento (";
        cout << N-1 << " bits c/u)= " << entrena; }
        cout << "\nRelación Señal a Ruido sn= " << sn << " dB";
        cout << "\nGráfico en curso: ";
        if (graf==0) cout << "Bits Transmitidos";
        if (graf==1) cout << "Espectro NRZ";
        if (graf==2) cout << "Bits Recibidos";
        if (graf==3) {cout << "Convergencia del Error del ";
            cout << "Filtro TDL";}
        if (graf==4) {cout << "Convergencia del Error del ";
            cout << "Filtro de Malla";}
        if (graf==5) {cout << "Convergencia de los Coeficientes ";
            cout << "del Filtro TDL";}
        if (graf==6) {cout << "Convergencia de los Coeficientes ";
            cout << "del Filtro de Malla";}
        if (graf==7) cout << "Bits Ecualizados con el Filtro TDL";
        if (graf==8) {cout << "Bits Ecualizados con el Filtro";
            cout << " de Malla";}

        if (graf==20) cout << "Ninguno";
        if (transm==2) cout << "\n# coef. TDL=" << ca << " mu=" << mu;
        if (transm==3) {cout << "\n# coef. Malla=" << ca;
            cout << " eps=" << eps << " w=" << w;}

        cout << "\n\nPaso " << l << " de " << cuanto;
        genera(y, b_r, sem, evol_g, NN);
        if (graf==0 && l==cuanto){
            cout << "\nGráfico de los Bits Tx.";
            N1=NN;
            archivo(y, N1, 2, l, ca, file, b);
        }
        cout << "\n\nTransformada Directa.";
        cout << "\nCalculando...";
        fft(y, -1, NN); // d_i: -1, FFT directa; +1, FFT inversa.
        if (graf==1 && l<=1)
        {
            cout << "\nGenerando Espectro NRZ en data.txt";
            N1=NN;
            archivo(y, N1, 1, l, ca, file, b);
        }
        ftr(y, baud, alfa, b, 0, NN);
        canal(y, baud, fp, b, NN);
        cout << "\n\nTransformada Inversa.";
        cout << "\nCalculando...";
        fft(y, 1, NN);
        noise(y, sem, sn, NN);
    }
}

```

```

cout << "\n\nTransformada Directa.";
cout << "\nCalculando...";
fft(y, -1, NN);
ftr(y, baud, alfa, b, 1, NN);
cout << "\n\nTransformada Inversa.";
cout << "\nCalculando...";
fft(y, 1, NN);
    if (graf==2 && l==cuanto)
    {
        cout << "\nGráfico de los Bits Rx.";
        N1=NN;
        archivo(y, N1, 2, l, ca, file, b);
    }

if (transm==1)    //TRANSMISION SIN ECUALIZACION.
{
    cout << "\n\naCálculo del Error. Presione";
    cout << " cualquier tecla para seguir...";
    getch();
    double re, yr;
    k=0;
    for (i=decis; i<=NN; i+=MUESTRAS)
    {
        k++;
        re=real(y[i]);
        decide(re, yr);
        y_r[k]=yr;

        cout << "\ny_e["<<i<<"]="<<real(y[i]);
        cout <<" br="<<b_r[k];

        err_t_a=err_t_a+abs((b_r[k]-y_r[k])/2);
        cout <<" err_t_a="<<err_t_a;
    }
    cout << "\nfin error\n";
    goto comun;
}

//intro.cpp es común para los dos algoritmos,
//se encarga de alimentar a la línea de retardos.

intro(y, y_r, NN, decis, evol_g);

//    Ecualización mediante el Filtro TDL.
if (transm == 2)
{
    cout<<"\n\naEcualización mediante el Filtro TDL";
    cout<<"\nPulse cualquier tecla...";
    getch();
    o=0;
}

```

```

//inicialización de los coeficientes y de los
//contenidos del Filtro TDL.

if (l<=1)
{
for (i=1; i<=ca; i++) // i++ equivale a i=i+1
h_r[i]=0; //Ganancias del filtro TDL
//inicializadas a 0.

for (i=1; i<=ca-1; i++){
ye_r[i]=0;
if (tdl==1)//Evolución de las variables en
//pantalla
{
cout << " ye_r["<<i<<"]="<<ye_r[i];
}
}
}

for (k=1; k<=b; k++)
{
o--; //Sirve para la graficación.
//Ingresa nuevo dato.
ye_r[ca]=y_r[k];
if (tdl==1)
{
cout << " nuevo dato= ";
cout << " ye_r["<<ca<<"]="<<ye_r[ca];
}
//Cálculo de a_r, salida del ecualizador;
//ad_r, valor deseado y de e, error.
cout << "\n\nsalida del ec.";
a_r=0;
for (i=1; i<=ca; i++)
{
a_r+=h_r[i]*ye_r[i]; // p+=q equivale a p=p+q
}

if (graf==7 && l==cuanto){//Gráfico de
//Convergencia de los Coeficientes del Filtro
//TDL.
for (i=1; i<=MUESTRAS; i++){
oo++;
y[oo]=complex(a_r,0);
}
}

decide(a_r,ad_r);
cout<<" ar= "<<a_r;
cout<<" b_r["<<k<<"]="<<b_r[k];
err_t_a+=abs((b_r[k]-ad_r)/2);
cout << " errt_a="<< err_t_a;

```

```

if (l<=entrena)
{ //Período de Aprendizaje.
  //Actualización de las ganancias h
  //del filtro.
  e_r=a_r-b_r[k];

  for (i=1; i<=ca; i++)
  {
    h_r[i]=h_r[i]-mu*(ye_r[i]*e_r);
    if (tdl==1) cout<<"\nh_r["<<i<<"]="<<h_r[i];
    if (graf==5 && i>=(ca-CO_G+1) && i<=ca){
      o++;
      y[k+o]=complex(0,h_r[i]);//con ca=7 y
      //256 bits estamos en 1792, dentro de la
      //capacidad de y[].
      //Graf. conv. coef. TDL.
    }
  }
}
else{
e_r=a_r-ad_r;

for (i=1; i<=ca; i++)
{
  if (tdl==1) cout<<"\nh_r["<<i<<"]="<<h_r[i];
  if (graf==5 && i>=(ca-CO_G+1) && i<=ca){
    o++;
    y[k+o]=complex(0,h_r[i]);//Graf. conv.
    //coef. TDL.
  }
}
}
if (graf==3){//Graf. de Conv. del Error del
  //Filtro TDL
  y[k]=complex(e_r*e_r,0);
  cout <<" error cuadrático=";
  cout <<real (y[k]);
}

//Desplazamiento de los ye en el ecualizador.
for (i=2; i<=ca; i++)
{
  ye_r[i-1]=ye_r[i];
  if (tdl==1)
  {
    cout << "\ndespl: i-1="<<i-1;
    cout <<" ye_r[i-1]="<<ye_r[i-1];
  }
}
} //cierra el lazo de ecualización para b bits

} //Cierra la ecualización con el Filtro TDL.

```

```

if (transm == 3)
{
cout << "\n\n\aEcuación mediante el";
cout << " Filtro de Malla.";
getch();
o=0;
if (l<=1){
for (n=0; n<=ca-1; n++)
{
ra[n]=0; //backward error anterior
//(al tiempo i-1).
ga[n]=0; //gama anterior.
pa[n+1]=0; //parcor anterior de la malla.
pca[n+1]=0; //parcor anterior del proceso adjunto.
sba[n]=eps; //covarianza backward anterior
//inicializada a un valor suficientemente
//pequeño para asegurar inversibilidad
//de la matriz de covarianza.
sfa[0]=eps; //covarianza forward anterior.
}
}

for (i=1; i<=b; i++)
{
a[0]=g[0]=0;
f[0]=r[0]=y_r[i];
cout << "\ni="<<i;
cout << "\nf[0]=r[0]="<<f[0];
sf[0]=sb[0]=w*sfa[0]+y_r[i]*y_r[i];
cout << " sf[0]=sb[0]="<<sf[0];
sfa[0]=sf[0];
if (l<=entrena)
{
c[0]=b_r[i]; //Período de entrenamiento.
cout << " c[0]="<<c[0];
}
else
{//Período de autoajuste.
for (n=0; n<=ca-1; n++)
{
p[n+1]=w*pa[n+1]+f[n]*ra[n]/(1-ga[n]);
kb[n+1]=p[n+1]/sba[n];
kf[n+1]=p[n+1]/sf[n];
f[n+1]=f[n]-kb[n+1]*ra[n];
r[n+1]=ra[n]-kf[n+1]*f[n];
a[n+1]=a[n]+kc[n+1]*r[n];
}
aca=a[ca];
decide(aca,ad_r);
c[0]=ad_r;
cout << "\nautoajuste: c[0]="<<c[0];
}
}
}

```



```

o--;
for (n=0; n<=ca-1; n++)
{
p[n+1]=w*pa[n+1]+f[n]*ra[n]/(1-ga[n]);
kb[n+1]=p[n+1]/sba[n];
kf[n+1]=p[n+1]/sf[n];
f[n+1]=f[n]-kb[n+1]*ra[n];
r[n+1]=ra[n]-kf[n+1]*f[n];
sf[n+1]=sf[n]-p[n+1]*kb[n+1];
sb[n+1]=sba[n]-p[n+1]*kf[n+1];
g[n+1]=g[n]+r[n]*r[n]/sb[n];

if (malla==1) //Evolución de las variables
//en la pantalla.
{
cout <<" kb["<<n+1<<"]~kf="<<kb[n+1];
cout <<" f["<<n+1<<"]="<<f[n+1];
cout <<" r["<<n+1<<"]="<<r[n+1];
cout <<" g["<<n+1<<"]="<<g[n+1];
}

if (l<=entrena)
{//Período de entrenamiento.
pc[n+1]=w*pca[n+1]+c[n]*r[n]/(1-g[n]);
kc[n+1]=pc[n+1]/sb[n];
a[n+1]=a[n]+kc[n+1]*r[n];
c[n+1]=b_r[i]-a[n+1];
}
else
{//Período de autoajuste.
a[n+1]=a[n]+kc[n+1]*r[n];
//Los kc[n+1] ya fueron calculados en
//el período de entrenamiento.
aca=a[n+1];
decide(aca,ad_r);
c[n+1]=ad_r-a[n+1];
}

if (malla==1){
cout <<" c["<<n+1<<"]="<<c[n+1];
cout <<" kc["<<n+1<<"]="<<kc[n+1];
}

if (graf==6 && n<=CO_G-1){
o++;
y[i+o]=complex(0,kc[n+1]);
}

//Desplazamiento en los registros.
ra[n]=r[n]; sba[n]=sb[n]; pa[n+1]=p[n+1];
pca[n+1]=pc[n+1]; ga[n]=g[n];

```

```

} // Fin de una iteración n.

if (malla==1){
  cout << " sf["<<ca<<"]~sb="<<sf[ca];
  cout << " a["<<ca<<"]="<<a[ca];
  cout << " c["<<ca<<"]="<<c[ca];
}
aca=a[ca];

if (graf==8 && l==cuanto){ //Gráf. de Ecu. Malla.
  for (j=1; j<=MUESTRAS; j++){
    oo++;
    y[oo]=complex(aca,0);
  }
}
decide(aca,ad_r);

err_t_a+=abs((b_r[i]-ad_r)/2);
cout << "\nb_r["<<i<<"]="<<b_r[i];
cout << " ad_r="<< ad_r<< " errt_a="<< err_t_a;
if (graf==4) //Graf de Conv. del Error
{
  y[i]=complex(c[ca]*c[ca],0);
  cout<< " error cuadr="<<real (y[i]);
}
} // Fin de una secuencia de b bits.
} // Cierra la ecualización con el Filtro de Malla

if (graf==3 || graf==4)
{
  N1=b;
  if (transm==2){
    cout<< "\n\nConvergencia del Error del Filtro TDL";
  }
  else{
    cout<< "\n\nConvergencia del Error del Filtro de Malla";
  }
  archivo(y,N1,3,1,ca,file,b);
}

if (graf==5 || graf==6)
{
  N1=b;
  if (transm==2){
    cout<< "\n\nConvergencia de los coefic. del Filtro TDL.";
  }
  else{
    cout<< "\n\nConvergencia de los coefic. del Filtro ";
    cout<< "de Malla.";
  }
  archivo(y,N1,4,1,ca,file,b);
}

```

```

if ((graf==7 || graf==8) && l==cuanto)
{
    N1=NN;
    if (transm==2){
        cout<<"\n\nBits Ecualizados con el Filtro TDL.";
    }
    else{
        cout<<"\n\nBits Ecualizados con el Filtro ";
        cout<<"de Malla.";
    }
    archivo(y,N1,2,l,ca,file,b);
}

comun: bit_t=b*l;
        pe_a=err_t_a/bit_t;
        cout << "\npe_a="<< pe_a;
        cout << "\n";

} // cierra el lazo l, para secuencias mayores de b bits.
cout << "\nEso es todo";
getch();
}

void decide(double a1, double &a2)
{
    if (a1<0) a2=-1; else a2=1;
}

```

```

//Subrutina que almacena los valores a ser graficados,
//los cuales le llegan en real(y[]), en el archivo data.txt

void archivo(complex y[],int N1, int tipo_g, int l, int ca,
             char *file, int b)
{
FILE *textfile;          /* Puntero al archivo usado */
char  modulus[50],is[20];
double m,n[8];
int  cal;
if (ca>CO_G) cal=CO_G; else cal=ca;
int  i,j,k=0,q,o=1;
int  sig = 24; /* digitos significantes */
cout<<"\n\n\AIntroduzca el diskette en b:.";
cout<<" Pulse cualquier tecla cuando esté listo...\n";
getch();
if ((textfile = fopen(file, "a")) == NULL) {
    printf("Error opening text file for writing\n");
    exit(0);
}
for (i=1; i<=N1; i++){
    switch (tipo_g){
    case 1: m=abs(y[i]);o=0;
            cout<<"\n"<<i<<" NRZ="<<m;
            break;
    case 2: m=real(y[i]);o=0;
            cout<<"\n"<<i;
            cout<<" Bit(Tx/Rx/Ecualizado)="<<m;
            break;
    case 3:
            m=real(y[i]);
            cout<<"\n"<<i+b*(l-1);
            cout<<" error cuadrado del Filtro (TDL o Malla)="<<m;
            break;
    case 4: k--;
            for (j=1; j<=cal; j++){
                k++;
                n[j]=imag(y[i+k]);
                cout<<"\n"<<i+k+b*(l-1)*cal<<" Conv. coef. n[";
                cout<<j<<"]="<<n[j];
            }
            break;
    }

if (tipo_g==4){
    itoa((i+k+b*(l-1)*cal)/cal,is,10);
    fprintf(textfile, "%s ", is);
    for (q=1; q<=cal-1; q++){
        gcvt(n[q], sig, modulus);
        fprintf(textfile, "%s ",modulos);
    }
    gcvt(n[cal], sig, modulus);
}
}
}

```

```
    fprintf(textfile, "%s\n",modulos);
}
else{
    itoa(i+b*(l-1)*o,is,10);
    gcvt(m, sig, modulos);
    fprintf(textfile, "%s %s\n", is,modulos);
}
}
fclose(textfile);
cout << "\n";
}
```

```

//Subrutina que genera ruido gaussiano.
void noise(complex y[], double sem, double sn, int NN)
{
  cout << "\n\nRuido Gaussiano\n";
  cout << "Calculando...";
  double gauss(double sem, double vr);
  double randu(double sem, double &g);
  int i;
  double vr,g; //vr = voltaje de ruido gaussiano
  vr=sqrt(1/pow(10,sn/10));
  for(i=1;i<=NN;i++){
    randu(sem,g);
    sem=g/i*(NN/20)+300;
    y[i]=complex(real(y[i])+gauss(sem, vr),0);
  }
}

//Subrutina que genera variables aleatorias
//gaussianas, de media nula y voltaje de ruido vr.
double gauss(double sem, double vr)
{
  int i;
  double a=0,ruido;
  double g,g1;
  double randu(double sem, double &g);
  for(i=1;i<=12;++i){
    randu(sem,g);
    g1=(32767.-g)/32767.;
    a+=g1;
    sem=g;
  }
  ruido=(a-6)*vr;
  return ruido;
}

```

```

//Subrutina que sirve para generar randómicamente,
//con distribución uniforme los datos del tren NRZ,
//con valores de +1 y -1.

void genera(complex y[], double b_r[], double &sem,
            int evol_g, int NN)
{
    double g;
    double randu(double sem, double &g);
    cout << "\n\naGenerando valores aleatorios (";
    cout <<MUESTRAS<<" muestras por dato)";
    cout << "\nPresione cualquier tecla...";
    getch();
    double g1,g2;
    int u,v,c=0;
    for (u=1 ; u<=NN; u+=MUESTRAS)
    {
        c++;
        randu(sem,g);
        g1=(32767-g)/32767; //RAND_MAX = 32767, máximo valor
                           //randómico
        if (g1 < 0.5)
            b_r[c]=-1;
        else
            b_r[c]=1;

        sem=g;

        for (v=u; v<=u+MUESTRAS-1; v++) // MUESTRAS muestras por dato
            y[v] = complex(b_r[c], 0);
            if (evol_g==1)
            {
                cout << "\ny["<<v-1<<"] = " << y[v-1];
            }
    }
    cout << "\nFin Genera.\n";
}

//Subrutina que genera valores aleatorios entre 1 y 32767
//a partir de un valor inicial o semilla.

double randu(double sem, double &g)
{
    srand ((unsigned) sem);
    g=(rand() % 100000);
    return 0;
}

```

```
//Subrutina de Transformada Rápida de Fourier  
//(Directa e Inversa).
```

```
void fft(complex y[], double d_i, int NN)
```

```
{  
  double pi = 4 * atan(1);  
  int j,i,m;  
  complex temp,w;  
  j=1;  
  
  for (i=1; i<=NN; i++)  
  {  
    if (i<j)  
    {  
      temp = y[j];  
      y[j] = y[i];  
      y[i] = temp;  
    }  
  
    m = NN/2;  
  
    while(j>m)  
    {  
      j = j - m;  
      m = (m+1)/2;  
    }  
    j = j + m;  
  }  
  
  double istep, mmax=1;  
  double theta;  
  
  while (mmax < NN)  
  {  
    istep = 2 * mmax;  
    for (m=1; m<=mmax; m++)  
    {  
      theta = pi * (d_i*(m-1))/mmax;  
      w = complex (cos(theta), sin(theta));  
      for (i=m; i<=NN; i=i+istep)  
      {  
        j = i + mmax;  
        temp = w * y[j];  
        y[j] = y[i] - temp;  
        y[i] = y[i] + temp;  
      }  
    }  
    mmax = istep;  
  }  
  
  if (d_i>0) // Cómputo de FFT inversa para d_i=1.
```



```
{
  for (i=1; i<=NN; i++)
    y[i] = y[i]/NN;
}
```

```

//Subrutina Filtro (de Tx y Rx).
void ftr(complex y[], double baud, double alfa, int b,
        int t_r, int NN)
{
    if (t_r==0)
        cout<<"\n\nFiltro de Tx";
    else
        cout<<"\n\nFiltro de Rx";

    double delta = baud/b, fn=baud/2, f1=fn * (1 - alfa);
    double f2=fn*(1+alfa);
    double T=1/baud,a,ce;
    // delta = paso discreto de frecuencia = 1/256;
    // a= corrección de amplitud = x/sen(x)
    // T= período de muestreo.
    // ce= coseno elevado.
    double pi = 4 * atan(1);
    int i=1;
    double f=delta;

    while (f<f1){
        i++;
        f = delta*(i-1);
        if (t_r==0){
            a=pi*f*T/sin(pi*f*T);
            y[i]=y[i]*a;
            y[NN-i]=y[NN-i]*a;
        }
    }

    while (f<f2){
        i++;
        f = delta*(i-1);
        a=pi*f*T/sin(pi*f*T);
        ce=sqrt(.5*(1-sin(pi*T*(f-fn)/alfa)));
        y[i]=y[i]*ce;
        y[NN-i]=y[NN-i]*ce;

        if (t_r==0){
            y[i]=y[i]*a;
            y[NN-i]=y[NN-i]*a;
            if (abs(y[i])>10000) y[i]=y[i-1];
            if (abs(y[NN-i])>10000) y[NN-i]=y[NN-i+1];
            //El par de líneas anterior sirve para permitir
            //el ingreso de alfas > 1 y que aún así no
            //existan picos desmesurados en el espectro NRZ.
        }
    }

    while (i<=NN/2-1){
        i++;
    }
}

```

```
y[i]=complex (0,0);  
y[NN-i]=complex (0,0);  
}  
y[NN/2]=complex (0,0);  
}
```

```

//Subrutina que implementa las características
//de Amplitud y de Fase como recomienda el CCITT.
//(Rec. M102. Tomo IV. Libro Naranja).

```

```

void canal(complex y[], double baud, double fp, int b,
           int NN)
{
    double delta=baud/b, pi=4*atan(1);
    double f1=300, f2=400, f3=1500, f4=2200, f5=2700;
    double f6=3000, f7=3200, f;
    double tf, theta, a;
    //tf distorsión de fase en milisegundos.
    int i=1;
    double integer, re, im;

    cout<<"\n\nCanal";

    for (i=2; i<=NN/2; i++)
    {
        f = fp + delta*(i-1);
        if (f<f4)
            {
                continue;
            }
        else
            if (f<f5)
                {
                    a=-.0001034*f+1.2274;
                    y[i]=y[i]*a;
                    continue;
                }
            else
                if (f<f6)
                    {
                        a=-.0001034*f+1.2274;
                        tf=(5.602747*log10(f)-19.225)*.001;
                        theta = 2*pi*modf(f*tf, &integer);
                        re=a*cos(theta);
                        im=a*sin(-theta); // un retraso temporal corresponde
                                           // a una fase negativa en la
                                           // frecuencia.
                        y[i]=y[i]*complex(re,im);
                        continue;
                    }
                else
                    if (f<f7 || f>f7)
                        {
                            a=-.0028125*f+9.3548;
                            tf=(5.602747*log10(f)-19.225)*.001;
                            theta = 2*pi*modf(f*tf, &integer);
                            re=a*cos(theta);
                            im=a*sin(-theta);
                        }
    }
}

```

```

        y[i]=y[i]*complex(re,im);
    }
}

for (i=2; i<=NN/2; i++)
{
    f = fp - delta*(i-1);
    if (f3<f)
    {
        continue;
    }
    else
    if (f2<f)
    {
        a=1;
        tf=(-2.50368*log10(f)+7.95193)*.001;
        theta = 2*pi*modf(f*tf, &integer);
        re=a*cos(theta);
        im=a*sin(-theta);
        y[NN+2-i]=y[NN+2-i]*complex(re,im);
        continue;
    }
    else
    if (f1<f || f<f1)
    {
        a=.005*f-1;
        if (f>f1)
        tf=(-2.50368*log10(f)+7.95193)*.001;
        theta = 2*pi*modf(f*tf, &integer);
        re=a*cos(theta);
        im=a*sin(-theta);
        y[NN+2-i]=y[NN+2-i]*complex(re,im);
    }
}
}
}

```

```

//Subrutina que introduce los datos que llegan luego
//de atravesar el canal y los filtros de Tx y Rx.

void intro(complex y[], double y_r[], int NN, int decis,
           int evol_g)
{
  cout << "\n\n\taIntroducción de datos al Ecualizador";
  cout << " (TDL ó Malla).";
  cout << "\nPulse cualquier tecla...";
  getch();
  int j=0,i;
  for (i=decis; i<=NN; i+=MUESTRAS)
  {
    j++; //j=j+1
    y_r[j]=real(y[i]);
    if (evol_g==1)
      cout << "\nyr[" << j << "]:"<<y_r[j];
  }
}

```

APENDICE III

APENDICE III .

TRANSFORMADA DE FOURIER, SISTEMAS ESTOCÁSTICOS Y RUIDO.

A3.1 La forma de onda correspondiente al espectro coseno elevado.

Está por demostrarse que la respuesta impulsiva del coseno elevado,

$$h(t) = \frac{w_n}{\pi} \frac{\text{sen}(w_n t)}{(w_n t)} \cos(w_n t) / (1 - (2w_n t/\pi)^2) \quad (\text{A3.1.1})$$

se deduce a partir de la definición en frecuencia de $H(w)$:

$$H(w) = \begin{cases} 1/2 (1 + \cos(\pi w / 2 w_n)), & \text{abs}(w) \leq 2 w_n \\ 0 & \text{otro caso} \end{cases} \quad (\text{A3.1.2})$$

Para el efecto usamos la definición de la transformada de Fourier:

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(w) \exp(jwt) dw \quad (\text{A3.1.3})$$

$$= \frac{1}{2\pi} \frac{1}{2} \int_{-2w_n}^{2w_n} \exp(jwt) dw + \frac{1}{2\pi} \frac{1}{2} \int_{-2w_n}^{2w_n} \cos(\pi w/2w_n) \exp(jwt) dw \quad (\text{A3.1.4})$$

$$= \frac{1}{4\pi} \frac{\exp(jwt)}{(jt)} \left[\frac{\exp(j\pi w/2w_n)}{-2w_n} + \frac{\exp(j\pi w/2w_n)}{2w_n} \right] + \frac{1}{4\pi} \int_{-2w_n}^{2w_n} (\exp(j\pi w/2w_n) + \exp(-j\pi w/2w_n)) / 2 \exp(jwt) dw \quad (\text{A3.1.5})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) + \frac{1}{8\pi} \int_{-2\omega n}^{2\omega n} \exp(j\omega(\pi/2\omega n + t)) + \exp(j\omega(-\pi/2\omega n + t)) d\omega \quad (\text{A3.1.6})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) + \frac{1}{8\pi} \left[\frac{1}{j(\pi/2\omega n + t)} \exp(j\omega(\pi/2\omega n + t)) \right]_{-2\omega n}^{2\omega n} + \frac{1}{8\pi} \left[\frac{1}{j(-\pi/2\omega n + t)} \exp(j\omega(-\pi/2\omega n + t)) \right]_{-2\omega n}^{2\omega n} \quad (\text{A3.1.7})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) + \frac{1}{4\pi} \left[\frac{1}{(\pi/2\omega n + t)} (-\text{sen}(2\omega t)) + \frac{1}{(-\pi/2\omega n + t)} (-\text{sen}(2\omega t)) \right] \quad (\text{A3.1.8})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) - \frac{1}{4\pi} \text{sen}(2\omega t) \left(\frac{1}{(\pi/2\omega n + t)} + \frac{1}{(-\pi/2\omega n + t)} \right) \quad (\text{A3.1.9})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) - \frac{1}{4\pi} \text{sen}(2\omega t) \left(\frac{1}{(t + \pi/2\omega n)} + \frac{1}{(t - \pi/2\omega n)} \right) \quad (\text{A3.1.10})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) - \frac{1}{4\pi t} \text{sen}(2\omega t) \left(\frac{1}{(1 + \pi/(2\omega n t))} + \frac{1}{(1 - \pi/(2\omega n t))} \right) \quad (\text{A3.1.11})$$

$$= \frac{1}{2\pi t} \text{sen}(2\omega t) - \frac{1}{4\pi t} \text{sen}(2\omega t) \frac{2}{(1 - (\pi/(2\omega n t))^2)} \quad (\text{A3.1.12})$$

$$= \frac{\text{sen}(2\omega t)}{2\pi t} \left(1 - \frac{1}{(1 - (\pi/(2\omega n t))^2)} \right) \quad (\text{A3.1.13})$$

$$= \frac{1}{2} \frac{\text{sen}(2\omega t)}{\pi t} \frac{1}{(1 - (2\omega n t/\pi)^2)} \quad (\text{A3.1.14})$$

multiplicando y dividiendo (A3.1.14) por $w\pi$ y usando la identidad $\sin 2a = 2 \sin a \cos a$,

$$h(t) = \frac{\sin(\pi w t)}{\pi w t} \cos(\pi w t) / (1 - (2w\pi t)^2) \quad (\text{A3.1.15})$$

LOQD.

A3.2 Demostración de que las funciones pares en el tiempo son funciones sólo del coseno.

Partimos de la definición de la transformada inversa de Fourier (A3.2.1)

$$H(w) = \int_{-\infty}^{\infty} h(t) \exp(-j\omega t) dt \quad (\text{A3.2.1})$$

$$H(w) = \int_{-\infty}^{\infty} h(t) (\cos \omega t - j \sin \omega t) dt \quad (\text{A3.2.2})$$

Para una función par se tiene $f(t) = f(-t)$, por lo que de (A3.2.2), poniendo $-t$ en lugar de t :

$$H(w) = \int_{-\infty}^{\infty} h(t) (\cos \omega t + j \sin \omega t) dt \quad (\text{A3.2.3})$$

en donde hemos aplicado la definición de función par para $f(t)$ y para el $\cos \omega t$.

Debido a que el producto de la función par $f(t)$ y la impar $\sin \omega t$ dan una integral cero ó puesto que $H(w)$ es única, la forma en que (A3.2.2) iguale a (A3.2.3) es que $H(w)$ no tenga componentes imaginarios; se tiene:

$$H(\omega) = \int_{-\infty}^{\infty} h(t) \cos(\omega t) dt$$

(A3.2.4)

LOQD.

A3.3 Potencia de Ruido y Representación Espectral del Ruido.

En el estudio de la ecualización hemos supuesto que el ruido que se agrega a la señal tiene un comportamiento estadístico gaussiano con varianza σ^2 . Para continuar indicaremos la forma de medir la varianza σ^2 , así como la media y otros parámetros de interés.

Deseamos fundamentalmente saber cómo puede determinarse en forma cuantitativa el efecto que los filtros lineales y el sistema tienen sobre el ruido $n(t)$.

El análisis espectral de las señales aleatorias difiere ligeramente del análisis de las señales determinísticas. Sin embargo comparte muchos resultados hallados para el caso determinístico, a saber:

- el análisis espectral se basa en la distribución de potencia de la onda;
- se mantiene la relación inversa entre el tiempo y el ancho de banda, etc.

En la fig. 1 se muestra una señal aleatoria, la cual parece tener una apariencia impredecible e irregular.

A pesar de que la apariencia temporal es totalmente irregular, a nivel estadístico el proceso presenta los parámetros media y varianza los cuales son constantes y fáciles de manipular. Nos interesa saber si la transformación espectral de la señal presenta alguna regularidad que permita que dicha señal sea analizada

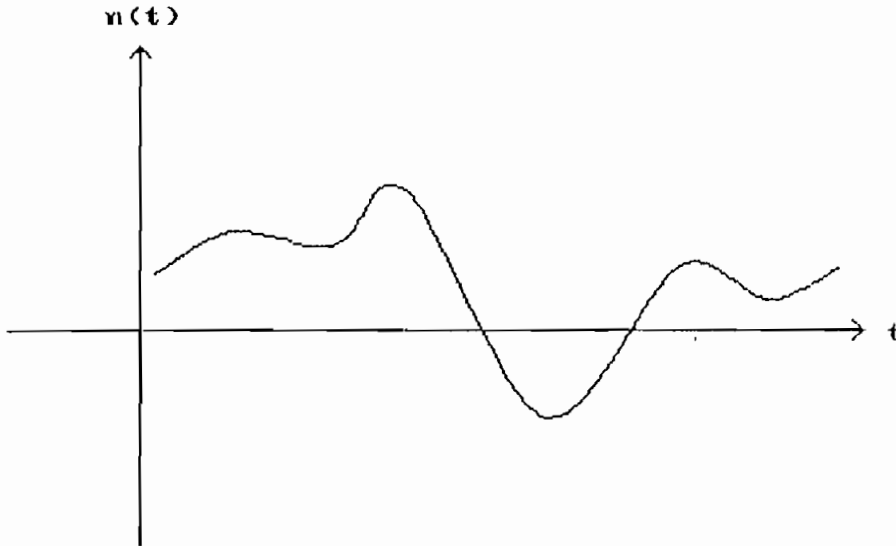


Figura 1 Proceso aleatorio.

cuantitativamente al pasar por filtros (por ej. de caída senoidal) cuya respuesta es determinística.

Para hallar el valor promedio (Schwartz, op. cit.) se usan medidores de tiempo promedio. Por ejemplo, si se suministrara la onda de ruido $n(t)$ a un medidor de cc. se supone intuitivamente que se obtendrá una medición del valor esperado $E(n)$. *En este caso se está comparando implícitamente un promedio esperado o estadístico con un promedio temporal, que es el que lleva a cabo un medidor de cc.*

Específicamente, si el instrumento tiene una constante de tiempo T , su lectura debe ser

$$\bar{n} = \frac{1}{T} \int_0^T n(t) dt \quad (A3.3.1)$$

Es claro que como $n(t)$ varía aleatoriamente, lo mismo sucede con \bar{n} . Dependiendo de cuándo se realiza el promedio indicado, se

obtendrán diferentes números \bar{n} . Por lo tanto \bar{n} es una variable aleatoria. Para encontrar en \bar{n} alguna medida de $E(n)$ se deben tomar muchas lecturas sobre diferentes secciones de $n(t)$, cada una de T seg. de duración. El valor esperado \bar{n} es entonces el promedio estadístico de ellas. Entonces

$$E(\bar{n}) = E\left(\frac{1}{T} \int_0^T n(t) dt\right) = \frac{1}{T} \int_0^T E(n(t)) dt \quad (A3.3.2)$$

en donde para el tercer miembro hemos usado la propiedad de que las operaciones de expectación e integración son intercambiables (Papoulis, Probability, Random Variables, and Stochastic Processes. Mc Graw-Hill 1965).

Con la suposición de que $E(n)$ es constante, independiente del tiempo, se halla a partir de (A3.3.2).

$$E(\bar{n}) = E(n) \quad (A3.3.3)$$

Esto indica que en un sentido promedio, el ruido medio \bar{n} proporciona una medición de $E(n)$.

Sin embargo nos interesa saber si podemos evitar el tomar muchas muestras. Nos interesa saber si aumentando el tiempo de medición se puede lograr el mismo resultado, es decir que \bar{n} se aproxime cada vez más a $E(n)$. La forma de averiguar si ello es posible es a través de la varianza de \bar{n} . Si esta varianza tiende a 0 a medida que el intervalo de medición crece sin límite, se habrá logrado demostrar que

$$\lim_{T \rightarrow \infty} \bar{n} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T n(t) dt = E(n). \quad (A3.3.4)$$

Por la definición de varianza,

$$\text{var}(\bar{n}) = E(\bar{n} - E(\bar{n}))^2 = E\left(\frac{1}{T} \int_0^T (n(t) - E(n(t)))^2 dt\right) \quad (\text{A3.3.5})$$

si dividimos la $\text{var}(\bar{n})$ para el valor esperado de n al cuadrado se tiene (Schwartz, op. cit.),

$$\text{var}(\bar{n})/E^2(n) + 1/BT, \quad BT \gg 1 \quad (\text{A3.3.6})$$

donde B es el ancho de banda del proceso, y a grosso modo está relacionado con la rapidez de variación de n (el caso de la relación inversa tiempo vs ancho de banda ya mencionada para las señales determinísticas), y T es el tiempo promedio del medidor.

Por ejemplo, si $B = 100$ KHz y $T = 1$ ms, $1/BT = 0.01$. La dispersión alrededor del valor esperado es entonces de 0.1 del valor esperado (tomando la raíz cuadrada de (A3.3.6) y definiendo la dispersión o desviación estándar como la $\text{sqrt}(\text{var}) = \text{sqrt}(\sigma^2) = \sigma$)

De aquí surge entonces lo que queríamos demostrar con (A3.3.4), de que si el medidor de cc toma promedios sobre intervalos de tiempo más y más largos, sus lecturas \bar{n} tienden a $E(n)$. (En situaciones prácticas es normalmente suficiente tener $T \gg 1/B$). Puesto que la varianza de la lectura \bar{n} tiende a cero conforme $1/T$, esto indica que las variaciones alrededor de $E(n)$ decrecen en la misma forma.

Un proceso para el cual la ecuación (A3.3.4) es válida se dice que es un proceso ergódico, es decir que el promedio del proceso puede ser determinado a través tanto de un promedio temporal como de un promedio estadístico. La ergodicidad implica un proceso estacionario, y cualquier proceso no estacionario es

no ergódico. Explicaremos posteriormente lo que se entiende por proceso estacionario.

A continuación apliquemos la metodología seguida para determinar $E(n)$ a través de un instrumento de medida (en ese caso un medidor de cc) a la determinación de la varianza σ^2 del ruido.

Para el efecto usamos un medidor de potencia total (rms + cc) y se demostrará que la σ^2 está relacionada con la potencia rms.

Si se define la potencia promedio total en un intervalo de T seg. de duración:

$$P_{pr} = \frac{1}{T} \int_0^T n^2(t) dt \quad (A3.3.7)$$

Tomando el promedio de conjunto (esperanza) a los dos miembros de (A3.3.7) se demuestra, nuevamente intercambiando el orden del promedio de conjunto con la integración, que

$$E(P_{pr}) = E(n^2) = \sigma^2 + E^2(n) \quad (A3.3.8)$$

$$\sigma^2 = E(P_{pr}) - E^2(n) \quad (A3.3.9)$$

donde hemos hecho uso en (A3.3.8) de la definición de varianza: $\sigma^2 = E(n^2) - E^2(n)$.

P_{pr} es una variable aleatoria, pero el promedio de lecturas da una medida del segundo momento $E(n^2)$ o Potencia total. Como $E^2(n)$ es aproximadamente el cuadrado del valor de cc., se concluye que la varianza σ^2 debe proporcionar una medida de la fluctuación o potencia rms (lqgd). En adelante llamaremos N (el cuadrado

del valor de voltaje o corriente rms medido por un voltímetro ca) a la varianza σ^2 , puesto que como se demostrará a continuación, conforme T tiende a infinito la varianza es un proceso ergódico.

En Schwartz (ob.cit) se encuentra que

$$\text{var}(P_{pr})/E^2(P_{pr}) \rightarrow 1/BT, \quad BT \gg 1 \quad (A3.3.10)$$

Entonces, cuando $T \rightarrow \infty$ (en la práctica $T \gg 1/B$) la lectura P_{pr} tiende a $E(n^2)$ con probabilidad 1, con lo que se tiene la fórmula complementaria a (A3.3.4)

$$\lim_{T \rightarrow \infty} (P_{pr} - \bar{n}^2) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T (n(t) - \bar{n})^2 dt = N \quad (A3.3.11)$$

Lo que demuestra que la varianza $\sigma^2 = P_{pr} - E^2(n) = P_{pr} - \bar{n}^2$ es un proceso ergódico.

De lo anterior se ve que con un instrumento de cc se determina $E(n)$ y retirando la componente cc, con un instrumento rms se determina σ^2 . Se debe tener, sin embargo, que $T \gg 1/B$, donde B es el ancho de banda del proceso.

La siguiente cuestión es por tanto, cómo se puede determinar B . Intuitivamente se puede concluir que no podría esperarse encontrar ruido con un ancho de banda B de 1 MHz a la salida de un sistema cuyo ancho de banda sea de 100 Hz. Cualquier componente de ruido que varíe tan rápidamente como en este ejemplo no podría aparecer a la salida del sistema.

Básicamente, lo que se requiere es alguna medición de cómo puede variar el proceso del ruido en un intervalo de tiempo dado. Específicamente para la fig 2 se nota que cuando $t_2 \rightarrow t_1$, $n(t_2)$ llega a estar más cercanamente relacionada o es más predecible

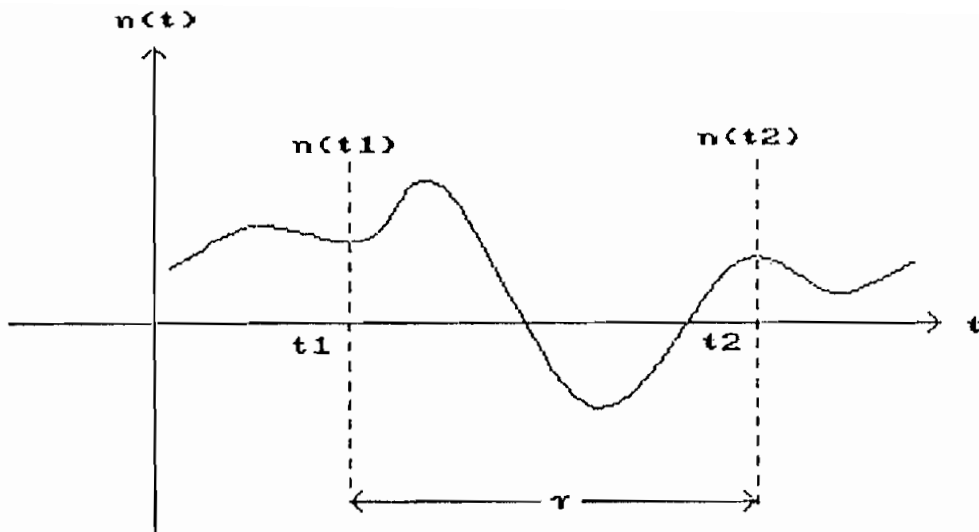


Figura 2 Definición de autocorrelación.

por $n(t_1)$. Cuando $(t_2 - t_1)$ aumenta, se espera menor dependencia entre ambas variables. Este concepto se puede precisar más exactamente definiendo la función de autocorrelación $R_n(t_1, t_2)$:

$$R_n(t_1, t_2) \triangleq E(n(t_1)n(t_2)) \quad (A3.3.12)$$

Nótese que la función de autocorrelación es la extensión a la variable misma (de allí el prefijo auto) de la covarianza de dos variables aleatorias que existe en la teoría de probabilidades. Es evidente que si $t_2 = t_1$, $R_n = E(n^2)$. Si en algún intervalo $t_2 - t_1$, $n(t_2)$ y $n(t_1)$ tienden a ser estadísticamente independientes (lo que ocurrirá en intervalos mayores que $1/B$), $R_n = E^2(n)$ ($= 0$ si $E(n) = 0$). Recordemos de la teoría de Probabilidad que el valor esperado de dos variables independientes es $E(xy) = E(x)E(y)$

R_n proporciona entonces una medición de la dependencia de $n(t_1)$ respecto de $n(t_2)$.

Para simplificar el análisis se supondrá que $R_n(t_1, t_2)$ depende únicamente del intervalo $(t_2 - t_1) = \tau$, y no del origen del

tiempo t_1 . [Esto es similar a la suposición hecha anteriormente de que $E(n)$ es independiente del tiempo]. Entonces,

$$R_n(\tau) \triangleq E(n(t)n(t+\tau)) \quad (A3.3.13)$$

Un proceso para el cual esto se cumple, y para el cual $E(n)$ es independiente del origen del tiempo, se denomina Proceso Estacionario.

Siguiendo un proceso semejante al seguido en el caso de la media y de la varianza del ruido, se puede demostrar que $R_n(\tau)$ para un proceso ergódico puede obtenerse de:

$$\lim_{T \rightarrow \infty} \check{R}_n(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T n(t)n(t+\tau) dt = R_n(\tau) \quad (A3.3.14)$$

Nota:(A3.3.14) incluye como resultado especial, cuando $\tau=0$, a $P_{pr}=E(n^2)$.

Procederemos ahora a relacionar $R_n(\tau)$ con el análisis espectral del ruido $n(t)$ y definir un ancho de banda asociado B .

Recuérdese que la función de autocorrelación $R_n(\tau)$ se introdujo para proporcionar una medición de las variaciones en el tiempo del proceso aleatorio $n(t)$. *$R_n(\tau)$ desempeña entonces el mismo papel para procesos aleatorios que la función del tiempo misma desempeña en el caso determinístico.* En el caso de funciones determinísticas se encontró la representación espectral de una función del tiempo $f(t)$ tomando su transformada de Fourier. En el caso aleatorio se define formalmente la representación espectral del ruido $n(t)$ como la Transformada de Fourier de $R_n(\tau)$. Denominando a esta cantidad $G_n(f)$, se tiene

$$G_n(f) = \int_{-\infty}^{\infty} R_n(\tau) \exp(-j\omega\tau) d\tau \quad (A3.3.15)$$

Esta función tiene un valor finito, lo que determina que cuando $R_n(\tau)$ toma valores más y más próximos a cero (lo que implica que $n(t)$ es cada vez más independiente de $n(t+\tau)$ o que $n(t)$ varía más rápidamente), G_n se hace más ancha en frecuencia. *Esta propiedad es consistente con la relación inversa entre el tiempo y la frecuencia.*

R_n puede encontrarse de G_n mediante la Transformada Inversa de Fourier:

$$R_n(\tau) = \int_{-\infty}^{\infty} G_n(f) \exp(j\omega\tau) d\omega / (2\pi)$$

$$R_n(\tau) = \int_{-\infty}^{\infty} G_n(f) \exp(j\omega\tau) df \quad (A3.3.16)$$

Recordemos que $R_n(0) = E(n^2) = P_{pr}$ es la potencia total de la onda de ruido. A partir de (A3.3.16) se tiene

$$R_n(0) = E(n^2) = \int_{-\infty}^{\infty} G_n(f) df \quad (A3.3.17)$$

De esta última ecuación se ve que la dimensión de G_n es correspondiente a la potencia. Por esta razón $G_n(f)$ se denomina el *espectro de potencia*. Con ella se mide la distribución de la potencia del ruido en la frecuencia. Un instrumento medidor de potencia sintonizado a una frecuencia f_0 y midiendo la potencia

de un estrecho intervalo Δf alrededor de f_0 podría proporcionar una buena aproximación de $2 G_n(f) \Delta f$. [Como las frecuencias negativas son sólo un artificio, podría duplicarse $G_n(f)$ de la ecuación (A3.3.17) e integrarse sobre las frecuencias positivas].

En el caso especial donde el ruido tiene un promedio cero, $E(n)=0$, y

$$N = \int_{-\infty}^{\infty} G_n(f) df, \quad E(n)=0 \quad (A3.3.18)$$

De esta forma, la potencia rms del ruido o varianza $\sigma^2 = N$, (de la cual depende la probabilidad de error en la detección de pulsos), está relacionada directamente con $G_n(f)$.

En la fig. 3 se da el par autocorrelación-densidad espectral correspondiente a $R_n(\tau) = N \text{sen}(2\pi B\tau)/(2\pi B\tau)$. El valor promedio del ruido ha sido considerado cero y de esta forma $R_n(\tau)$ tiende a cero para τ suficientemente grande. Nótese también que $R_n(0)$ ha sido fijado a N y que el área bajo la curva de densidad espectral es por tanto N .

Nota: (A3.3.18) para el caso de ruido blanco da:

$$N = \int_{-\infty}^{\infty} G_n(f) df = n_0 B. \quad (A3.3.18a)$$

En la fig.3 es evidente la relación inversa tiempo-ancho de banda citada anteriormente. La función $R_n(\tau)$ tiende a cero para un espaciamiento τ mayor a $1/B$, lo que implica que $n(t)$ y $n(t+\tau)$ llegan a ser independientes. *El recíproco del ancho de banda desempeña de esta manera un papel muy importante en la determina-*

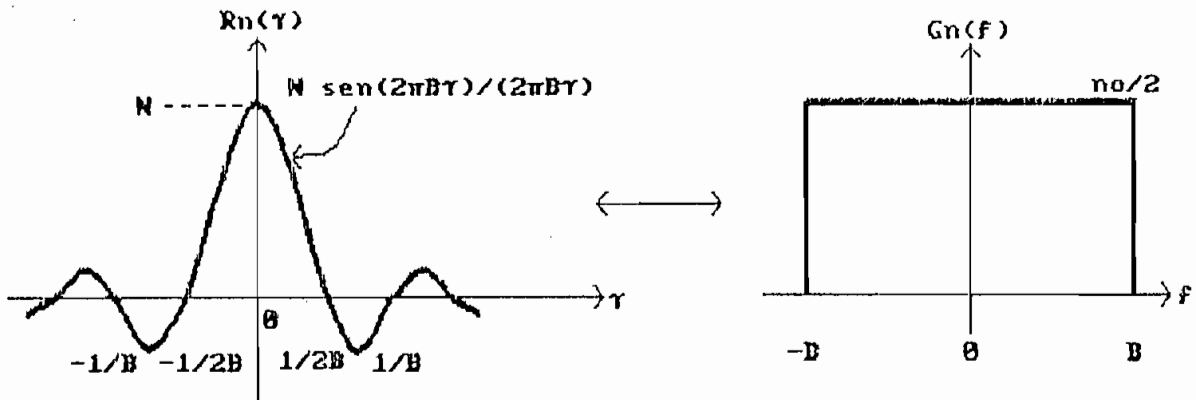


Figura 3 Función de autocorrelación y de densidad espectral

ción de la medida de la correlación entre una muestra de $n(t)$ y una muestra tomada τ seg. después.

Si en la fig. 3 hacemos que el ancho de banda tienda a ∞ , tendremos el caso del pulso de dirac. Este caso particular es en extremo importante, es decir, para

$$G_n(f) = \text{constante} = no/2, \text{ para toda } f \quad (A3.3.19)$$

tenemos el caso del ruido blanco.

Aunque esto es, estrictamente hablando, inadmisibile físicamente, puesto que implica una potencia de ruido infita (de (A3.3.18)), es un buen modelo para muchas situaciones típicas en las cuales el ancho de banda es tan grande que sale fuera del intervalo de medición de los instrumentos (o frecuencias de interés).

A este ruido de característica de frecuencia plana se lo llama ruido blanco debido a que es una "mezcla igual" de todas las frecuencias.

A la fig. 3 se le llama con frecuencia ruido blanco limitado en banda.

Para el ruido blanco, el par de transformadas son:

$$G_n(f) = n_0/2, \quad R_n(\tau) = n_0/2 \delta(\tau) \quad (A3.3.20)$$

Como la función de autocorrelación es un impulso, esto indica que $n(t)$ es siempre una función no correlacionada con $n(t+\tau)$, sin importar lo pequeño que τ pueda ser.

Esto implica que $n(t)$ puede variar en forma infinitamente rápida. En la práctica, tal como se ha mencionado, esto significa simplemente que las variaciones de alta frecuencia del ruido están más allá de la capacidad de los instrumento en la medición particular que se desea hacer. Por ejemplo, si los dispositivos de medición tienen una respuesta temporal $\gg 1/B$, donde B es el ancho de banda del ruido físico real, el ruido se presentará para todo propósito práctico como si fuera ruido blanco. Para un osciloscopio con ancho de banda de 50 MHz, ruido de $B=500$ MHz aparecerá para todo propósito práctico como ruido blanco.

A3.3.1 Señales Aleatorias y Ruido a través de los Sistemas Lineales.

Considérese la señal de ruido $n_e(t)$ con densidad espectral $G_{n_e}(f)$, potencia de ruido N_e y correlación $R_{n_e}(\tau)$ que entra al sistema lineal $H(w)$ tal como se muestra en la fig. 4.a. Nos interesan a la salida del sistema $n_s(t)$, $G_{n_s}(f)$, N_s y $R_{n_s}(\tau)$.

Intuitivamente se espera que si el ruido a la entrada varía a una velocidad proporcional a su ancho de banda, la cual es, digamos lenta en comparación con el ancho de banda del sistema B_{sis} , el ruido de salida difiere muy poco del ruido de entrada. Si por otra parte $B \gg B_{sis}$, las rápidas fluctuaciones de $n_e(t)$ no pueden pasar por el sistema y puede esperarse que $n_s(t)$ fluctúe más o menos a la velocidad de B_{sis} .

Esto es claro de ver de la siguiente expresión para la densidad espectral, la cual depende decisivamente de $H(w)^2$.

$$G_{ns}(f) = \text{abs}(H(w))^2 G_{ne}(f) \quad (A3.3.1.1)$$

Lo cual se puede demostrar como sigue² (los límites de los integrales son de $-\infty$ a ∞):

sea la salida del sistema H ,

$$n_s(t) = \int n_e(t-\alpha) h(\alpha) d\alpha \quad (A3.3.1.2)$$

multiplicando a ambos lados por $n_e(t_1)$ y tomando el valor esperado,

$$E(n_e(t_1)n_s(t)) = \int E(n_e(t_1)n_e(t-\alpha)) h(\alpha) d\alpha \quad (A3.3.1.3)$$

haciendo $t=t_2$,

$$R_{se}(t_1,t_2) = \int R_{ee}(t_1,t_2-\alpha) h(\alpha) d\alpha. \quad (A3.3.1.4)$$

Por otro lado, multiplicando a ambos lados de (A.3.1.2) por $n_s(t_2)$ y tomando el valor esperado,

² Papoulis, Sistemas Digitales y analógicos, transformadas de Fourier, estimación espectral. Marcombo S.A. 1978.

$$E(ns(t_2)ns(t)) = \int E(ns(t_2)ne(t-a)) h(a) da \quad (A3.3.1.5)$$

haciendo $t=t_1$,

$$R_{ss}(t_1, t_2) = \int R_{se}(t_1-a, t_2) h(a) da. \quad (A3.3.1.6)$$

Para procesos estacionarios sabemos que la covarianza depende solo del espaciamiento temporal entre las muestras, luego, con $t_1-t_2 = \tau$, (A3.3.1.4) se vuelve

$$R_{se}(\tau) = \int R_{ee}(\tau+a) h(a) da = R_{ee}(\tau) * h(\tau) \quad (A3.3.1.7)$$

haciendo $a = -a$,

$$R_{se}(\tau) = \int R_{ee}(\tau-a) h(-a) d(-a) = R_{ee}(\tau) * -h(-\tau) \quad (A3.3.1.8)$$

Con $\tau=t_1-t_2$, de (A3.3.1.6),

$$R_{ss}(\tau) = \int R_{se}(\tau-a) h(a) da = R_{se}(\tau) * h(\tau) \quad (A3.3.1.9)$$

por lo que sustituyendo R_{se} de (A3.3.1.7) en (A3.3.1.8) da

$$R_{ss}(\tau) = R_{ee}(\tau) * h(\tau) * -h(-\tau) \quad (A3.3.1.10)$$

y tomando la transformada de Fourier a ambos miembros de (A3.3.1.10) tenemos

$$G_{ss}(f) = \text{abs}(H(w))^2 G_{ee}(f) \quad \text{LQQD.}$$

en donde hemos usado la propiedad de escalado

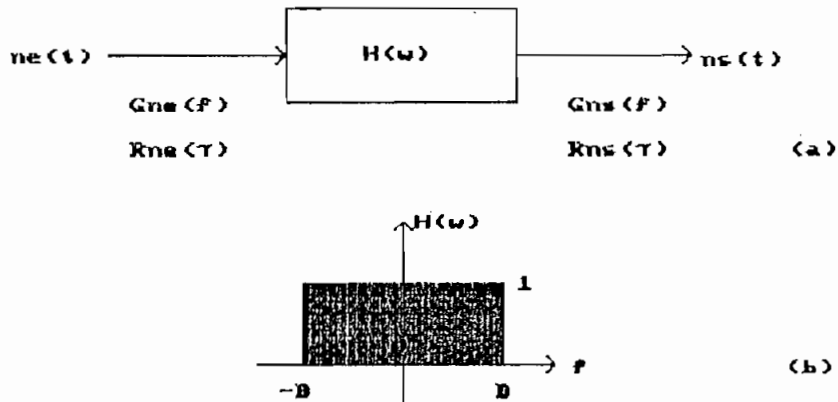
$$h(at) * 1/\text{abs}(a) F(w/a)$$

y hemos tomado el valor absoluto.

El siguiente ejemplo resume lo visto a la vez que profundiza en la relación de ruido de salida vs ancho de banda.

Supóngase que se tiene ruido blanco de densidad espectral $n_0/2$ aplicado a la entrada de un filtro pasabajas de ancho de banda B Hz como se muestra en la fig. 4.a., 4.b.

El ruido de salida es el ruido de banda limitada de densidad



Si $G_{ne}(f) = n_0/2$ (ruido blanco), a la salida del filtro pasabajas se tiene $G_{ns}(f) = n_0/2$, $\text{abs}(f) < B$ y 0 en otro caso, que definimos como RUIDO BLANCO LIMITADO EN BANDA.

Figura 4 Ruido blanco a través de un filtro ideal. (a) Modelo del sistema lineal. (b) Característica de frecuencia del filtro (sistema lineal).

espectral $G_{ns}(f) = n_0/2$, $\text{abs}(f) \leq B$, lo que resulta de aplicar (A3.3.1.1). La potencia del ruido de salida es entonces $N_s = n_0 B$, que resulta de aplicar (A3.3.18).

De aquí se ve que al aumentar el ancho de banda del filtro B , se incrementa la potencia del ruido de salida. También se incrementa la velocidad de variación del ruido de salida o se disminuye la correlación entre $n(t)$ y $n(t+\tau)$ para un τ fijo.

A3.4 El Teorema de la energía.

Definamos $y(t) \leftrightarrow Y(w)$, $y_1(t) \leftrightarrow Y_1(w)$, $y_2(t) \leftrightarrow Y_2(w)$,
 $Y_2^*(w) = Y_2(-w)$,

Se tiene que (los límites de las integrales van de $-\infty$ a ∞),

$$\int \text{abs}(y(t))^2 dt = 1/(2\pi) \int \text{abs}(Y(w))^2 dw \quad (\text{A3.4.1})$$

Demostración:

Asumamos que la señal $y_1(t)$ va a pasar a través de un filtro de característica $Y_2(w) = Y_1^*(w)$, lo que determina que $y_2(t) = y_1(-t)$ (por la propiedad de escalamiento ya mencionada).

$$\text{sea } g = y_1(t) * y_2(t) \leftrightarrow Y_1(w) Y_2(w)$$

obtenemos "g" a partir de $Y_1(w)Y_2(w)$ mediante la Transformada de Fourier Inversa,

$$g = \int y_1(\tau) y_2(t-\tau) d\tau = 1/(2\pi) \int Y_1(w) Y_2(w) \exp(jwt) dw \quad (\text{A3.4.2})$$

en donde con la característica mencionada del filtro Y_2 , y la implicación que ello tiene en y_2 , se tiene

$$g = \int y_1(\tau) y_1(-t+\tau) d\tau = 1/(2\pi) \int Y_1(w) Y_1^*(w) \exp(jwt) dw \quad (\text{A3.4.3})$$

con $t=0$,

$$g = \int y_1(\tau) y_1(\tau) d\tau = 1/(2\pi) \int Y_1(w) Y_1^*(w) dw \quad (\text{A3.4.4})$$

con $t=\tau$,

$$g = \int y_1(t) y_1(t) dt = 1/(2\pi) \int Y_1(w) Y_1^*(w) dw \quad (\text{A3.4.5})$$

de lo cual es inmediato, con $y=y_1$, $Y=Y_1$:

$$\int \text{abs}(y(t))^2 dt = 1/(2\pi) \int \text{abs}(Y(w))^2 dw \quad \text{LQPD.}$$

A3.5 La desigualdad de Schwarz.

Si a_1, \dots, a_n y b_1, \dots, b_n son números reales cualquiera, se tiene

$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right) \quad (\text{A3.5.1})$$

El signo de igualdad es válido si hay un número real x tal que $a_k x + b_k = 0$ para cada valor de $k=1, \dots, n$.

Demostración: (En lo que sigue, todos los sumatorios van de $k=1, \dots, n$). Para todo real x se tiene $\sum (a_k x + b_k)^2 \geq 0$, esto se puede poner en la forma

$$Ax^2 + 2 Bx + C \geq 0 \quad (\text{A3.5.2})$$

donde

$$A = \sum a_k^2, \quad B = \sum a_k b_k, \quad C = \sum b_k^2. \quad (\text{A3.5.2a})$$

Queremos demostrar que $B^2 \leq A C$. Si A es cero, cada a_k es cero y el resultado es trivial. Si $A \neq 0$, podemos completar el cuadrado y escribir

$$Ax^2 + 2 Bx + C = A(x+B/A)^2 + (AC-B^2)/A \quad (\text{A3.5.3})$$

El segundo miembro alcanza su valor mínimo cuando $x=-B/A$, lo que reemplazado en (A3.5.2) da $B^2 \leq AC$, lo que demuestra (A3.5.1).

Para demostrar que el signo de igualdad es válido si existe un x tal que $ax + bk = 0$ para todo k , partimos de $x=-B/A$, lo que con la notación (A3.5.2a) da

$$\begin{aligned} Ax + B &= x \sum a_k^2 + \sum a_k b_k = 0 \\ &= \sum (x a_k^2 + a_k b_k) = 0 \end{aligned} \tag{A3.5.4}$$

para $a_k \neq 0$,

$$Ax + B = \sum (x a_k + b_k) = 0 \tag{A3.5.5}$$

lo que es cero si $x a_k + b_k = 0$ para todo k , LQD.

En lo anterior, el $\sum a_k b_k$ no es más que el producto interno de dos vectores. Por otro lado, el producto interno también se puede representar como $\int f(t) g(t) dt$, donde f, g son dos funciones reales. Si tomamos siempre el módulo de las funciones, siendo el módulo un número real, es posible que f y g sean funciones complejas cualesquiera, lo que demuestra la desigualdad de Schwarz para la aplicación considerada en la maximización de la relación SNR.

APENDICE IV.

APENDICE IV.

ALGEBRA MATRICIAL.

Para motivar el tema empecemos afirmando que la *ecualización no es más que una manera eficiente de invertir la matriz de covarianza del sistema.*

En la ecualización uno trata de determinar el valor más probable de salida en función de los valores anteriores y el actual que ingresan al filtro.

En procesos de economía se usa la regresión lineal múltiple para estimar el valor de una variable en función de otras combinadas linealmente. Por ej. el valor de las acciones de los aviones dependen de los valores del acero, del caucho, etc; todo lo cual ingresa en forma de variables en una ecuación.

Para determinar el estimado se debe pues efectuar observaciones, y mientras mayor sea el número de observaciones el estimado estará más cerca al valor real.

Esto determina que el número de observaciones en algún momento supere al número de incógnitas, con lo cual el sistema de ecuaciones formado para hallar los coeficientes de ajuste se vuelve inconsistente.

A pesar de ello³, a uno le interesaría conocer los mejores valores posibles de las constantes de ajuste.

Ello se logra con el mecanismo de los LMS, el cual ya hemos aplicado en el capítulo 3 para obtener los Copt. De esta manera

³ Franklin, J. Matrix Theory. Prentice Hall 1968.

se logra un sistema que es consistente (el número de ecuaciones es igual al de incógnitas) y que permite conocer C_{opt} de manera única a través de:

$$C_{opt} = R^{-1}P \quad (A4.1)$$

En donde el procedimiento directo implica invertir la matriz de covarianza R .

Y como decíamos al principio, la ecualización trata con algoritmos que invierten en forma óptima esta matriz, por lo que se cumple la afirmación de que la ecualización no es más que la eficiente inversión de una matriz.

Ejemplo: Para un ecualizador TDL de 2 taps, hállese el mejor estimado de los coeficientes dados los siguientes datos:

$br = [1 \ -1 \ 1 \ 1]$	bits almacenados para referencia.
$y_0 = [.7 \ 0 \ 0]$	vector que entra al ecualizador al $t=0$
$y_1 = [.4 \ .7 \ 0]$	$t=1$
$y_2 = [-.6 \ .4 \ .7]$	$t=2$
$y_3 = [.6 \ -.6 \ .4]$	$t=3$

En el ejemplo se ve que inicialmente los bits eran todos cero antes de que empiece la transmisión, lo que corresponde al caso preventaneado de que se habló en el capítulo 5.

A continuación se observa que los bits se desplazan en los registros del Filtro TDL.

En el caso presente tenemos 2 taps que son las incógnitas y 3 datos, por lo que el sistema no tiene una solución exacta, sin embargo podemos usar el método LMS con la ecuación A4.1 para hallar C_{opt} .

Primeramente formamos un sistema de ecuaciones de partida:

$$\begin{array}{ccc|c} y_0 & 0 & 0 & c_0 \\ y_1 & y_0 & 0 & c_1 \\ y_2 & y_1 & y_0 & c_2 \end{array} * \begin{array}{c} \\ \\ \\ \end{array} = \begin{array}{c} br_0 \\ br_1 \\ br_2 \end{array} \quad (A4.2)$$

$$y_3 \quad y_2 \quad y_1 \qquad \qquad \qquad br_3$$

en donde hemos aumentado a la matriz con la última fila, que da cuenta del dato adicional.

Por tanto:

$$\begin{array}{ccc|c} .7 & 0 & .0 & c_0 \\ .4 & .7 & 0 & c_1 \\ -.6 & .4 & .7 & c_2 \end{array} * \begin{array}{c} \\ \\ \\ \end{array} = \begin{array}{c} 1 \\ 1 \\ -1 \end{array}$$

$$.6 \quad -.6 \quad .4 \qquad \qquad \qquad 1$$

$$Y \qquad * \quad C^{T_{opt}} = br$$

Usando el método de la matriz de covarianza dado en la ec. (6.15a) y con y_0, \dots, y_3 , ó considerando que Y^T es⁴:

$$Y^T = \begin{array}{cccc} .7 & .4 & -.6 & .6 \\ 0 & .7 & .4 & -.6 \\ 0 & 0 & .7 & .4 \end{array}$$

a partir de (A4.1)

$$RC_{opt} = P = Y^T br \quad (A4.3)$$

se tiene que:

$$R = Y^T Y = \begin{array}{ccc} 1.37 & -.32 & -.18 \\ -.32 & 1.01 & .04 \\ -.18 & .04 & .65 \end{array}$$

⁴ Friendlander, B. Ob. Cit.

$$Y^T br = \begin{vmatrix} 2.3 \\ -.3 \\ -.3 \end{vmatrix}$$

por lo que

$$R^{-1} = \begin{vmatrix} .816 & .25 & .21 \\ .250 & 1.07 & .003 \\ .21 & .003 & 1.597 \end{vmatrix}$$

y de (A4.1)

$$Copt = \begin{vmatrix} 1.7386 \\ .2536 \\ .004 \end{vmatrix}$$

Finalmente, para comprobar que los coeficientes hallados generan los estimados más aproximados, tenemos que, de (A4.2):

$$\begin{aligned} .7 \quad 1.7386 & & = 1.217 \quad \sim 1 \\ .4 \quad 1.7386 + .7 \quad .2536 & & = .8729 \quad \sim 1 \\ -.6 \quad 1.7386 + .4 \quad .2536 + .7 \quad .004 & = & -.9389 \quad \sim -1 \\ .6 \quad 1.7386 - .6 \quad .2536 + .4 \quad .004 & = & .8926 \quad \sim 1 \end{aligned}$$

que en un proceso de ecualización se convierte en la inversión de la matriz de covarianza a través de la actualización de los coeficientes C, como se demostró en el capítulo 5.

A continuación se explica lo relativo a las transformaciones lineales, la diagonalización de matrices, los eigenvalores y los eigenvectores.

A4.1 Construcción de una matriz a partir de una Transformación Lineal dada.

Recordemos que una Transformación Lineal es una función que partiendo de un conjunto llamado dominio genera elementos de otro conjunto llamado recorrido. La propiedad básica que satisface la TL es la de superposición:

$$L(cA+dB) = c L(A) + d L(b). \quad (A4.4)$$

propiedad que es satisfecha entre otras funciones por la derivación, la integración, la rotación respecto a un eje, etc.

Para el efecto de construir una matriz a partir de una TL dada, consideremos las ecuaciones de rotación de un punto en el plano (x,y) en sentido contrario al de las manecillas del reloj a través de un ángulo θ . Así, si P tiene coordenadas (x,y), entonces después de rotar, obtenemos P' con coordenadas (x',y'), donde:

$$\begin{aligned} x' &= x \cos\theta - y \operatorname{sen}\theta \\ y' &= x \operatorname{sen}\theta + y \cos\theta \end{aligned} \quad (A4.5)$$

a partir de las cuales se puede formar la matriz de la transformación lineal:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = L \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = A \begin{vmatrix} x \\ y \end{vmatrix}$$

Donde la matriz buscada es A, e inversamente, dada A se puede hallar la TL.

Encontremos la representación matricial del operador derivación definido en el espacio P2 con base (t²,t,1) que tiene como recorrido el espacio P1 con base (t,1).

Tenemos que:

$$L(t^2) = 2t = 2t + 0(1) \text{ de modo que } [L(t^2)]^T = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$

$$L(t) = 1 = 0t + 1(1) \text{ de modo que } [L(t)]^T = \begin{vmatrix} 0 \\ 1 \end{vmatrix}$$

⁵ Kolman. Algebra Lineal. Fondo Educativo Interamericano 1981.

$$L(1) = 0 = 0t + 0(0) \text{ de modo que } [L(1)]^T = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

con lo que la matriz correspondiente a la derivación de P2 en P1 es:

$$A = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}$$

Tomemos $p(t) = 5t^2 - 3t + 2$, entonces $L(p(t)) = 10t - 3$, lo que con la notación matricial:

$$[L(p(t))]^T = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} \begin{vmatrix} 5 \\ -3 \\ 2 \end{vmatrix} = \begin{vmatrix} 10 \\ -3 \end{vmatrix}$$

A continuación formemos la representación matricial para la derivación a partir de una base distinta en P2: $(t^2+t+1, t+1, 1)$ y la misma base en P1: $(t, 1)$:

$$L(t^2+t+1) = 2t + 1 = 2t + 1(1) \text{ de modo que } [L(t^2)]^T = \begin{vmatrix} 2 \\ 1 \end{vmatrix}$$

$$L(t+1) = 1 = 0t + 1(1) \text{ de modo que } [L(t)]^T = \begin{vmatrix} 0 \\ 1 \end{vmatrix}$$

$$L(1) = 0 = 0t + 0(0) \text{ de modo que } [L(1)]^T = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

con lo que la matriz correspondiente a la derivación de P2 en P1 es:

$$A = \begin{vmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \end{vmatrix}$$

Tomemos $p(t) = 5t^2 - 3t + 2$, con lo que la representación en la nueva base P2 es $p(t) = 5(t^2+t+1) - 3(t+1) + 2 = 5t^2 + 2t + 4$; entonces $L(p(t)) = 10t + 2$, lo que con la notación matricial:

$$[L(p(t))]^T = \begin{vmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \end{vmatrix} \begin{vmatrix} 5 \\ -3 \\ 2 \end{vmatrix} = \begin{vmatrix} 10 \\ 2 \end{vmatrix}$$

Se ve entonces (Apostol, ob. cit.), que la derivada del vector en la nueva base es inmediatamente obtenida con la representación de la derivación como matriz en la nueva base multiplicada por el vector original.

De aquí surge el hecho de que con bases distintas se obtienen distintas representaciones matriciales de una transformación lineal. Parece natural entonces intentar elegir bases de modo que la matriz resultante tenga una forma lo más sencilla posible.

La forma más sencilla es aquella en que todos los elementos, excepto los de la diagonal principal son cero. En este caso hablamos de las matrices diagonales, las cuales son extremadamente fáciles de invertir, por ejemplo, la inversa de:

$$A = \begin{vmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{vmatrix} \text{ es simplemente: } A^{-1} = \begin{vmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/4 \end{vmatrix}$$

A4.2 Transformaciones lineales representadas mediante matrices diagonales.

Notación: Si $A = (a_{ij})$ es una matriz diagonal, $A = \text{diag}(a_{11}, \dots, a_{nn})$

Teorema A4.2.1 Dada una transformación $L:V$ en V , donde $\dim V = n$. Si L tiene una representación en matriz diagonal, existe entonces un conjunto de elementos independientes u_1, \dots, u_n de V y un correspondiente conjunto de escalares $\lambda_1, \dots, \lambda_n$ tales que

$$L(u_k) = \lambda_k u_k \text{ para } k=1,2,\dots,n. \quad (\text{A4.6})$$

Demostración:

Previamente recordemos de la matriz de la derivación, que cuando se eligió los elementos $(t^2, t, 1)$ obtuvimos una matriz diagonal, no así cuando se eligió $(t^2+t+1, t+1, 1)$ como elementos de la base. De esto se ve, no rigurosamente, que se cumple (A4.6), ya que para el recorrido de un elemento k -ésimo sólo existe un elemento de partida, el cual es diferente a cualquier otro usado como elemento de partida para otro elemento j , $j \neq k$.

Supongamos que L tiene una representación en matriz diagonal $A = (a_{ik})$ con respecto a la base (e_1, \dots, e_n) . La acción de L sobre los elementos de la base viene dada por la fórmula

$$\begin{aligned}
 L(e_k) &= \sum_{i=1}^n a_{ik} e_i && (A4.7) \\
 &= a_{kk} e_k
 \end{aligned}$$

puesto que $a_{ik} = 0$ para i diferente de k . Esto demuestra (A4.6) con $u_k = e_k$ y $\lambda_k = a_{kk}$.

Así pues, el problema de hallar una representación en matriz diagonal de una transformación lineal se ha reducido a otro, el de hallar los elementos independientes u_1, \dots, u_n y los escalares $\lambda_1, \dots, \lambda_n$ que satisfagan (A4.6). Tales elementos u_k y escalares λ_k se llaman autovectores y autovalores de L respectivamente.

A4.3 Polinomios Característicos.

Si $\dim V = n$, el problema de determinar los autovalores de una transformación lineal $L: V \rightarrow V$ puede resolverse con la ayuda de los determinantes. Deseamos hallar aquellos escalares λ tales que la ecuación $L(x) = \lambda x$ tenga una solución $x \neq 0$. (Se elimina $x=0$ debido a que se trata de que a cada eigenvalor le corresponda un único eigenvector). La ecuación $L(x) = \lambda x$ puede escribirse:

$$(\lambda I - L)(x) = 0 \quad (\text{A.4.8})$$

donde I es la transformación identidad. λ es un autovalor si la ecuación anterior tiene solución para x no nulo, en cuyo caso $(\lambda I - L)$ es no invertible (ya que es igual a 0). Si A es una representación matricial de L , entonces $\lambda I - A$ es no singular (invertible) si $\det(\lambda I - A) \neq 0$. Así pues, si λ es un autovalor de L debe satisfacer la ecuación

$$\det(\lambda I - A) = 0 \quad (\text{A.4.9})$$

Esto sugiere que el determinante mencionado debería estudiarse como una función de λ .

Lema A.4.1: Si A es una matriz $n \times n$ cualquiera e I es la matriz identidad $n \times n$, entonces la función f definida por la ecuación

$$f(\lambda) = \det(\lambda I - A) \quad (\text{A.4.10})$$

es un polinomio en λ de grado n .

Definición: Si A es una matriz $n \times n$ el determinante $f(\lambda) = \det(\lambda I - A)$ se denomina polinomio característico de A .

Antes de pasar a ver la traza de una matriz, veamos cómo se halla los eigenvectores hallados los eigenvalores.

Ejemplo: la matriz

$$A = \begin{vmatrix} 2 & 1 & 1 \\ 2 & 3 & 2 \\ 3 & 3 & 4 \end{vmatrix}$$

tiene el polinomio característico $(\lambda-1)(\lambda-1)(\lambda-7)$. Cuando $\lambda=7$ el sistema $AX = 7X$ se convierte en:

$$\begin{aligned} 5x_1 - x_2 - x_3 &= 0 \\ -2x_1 + 4x_2 - 2x_3 &= 0 \\ -3x_1 - 3x_2 + 3x_3 &= 0 \end{aligned}$$

el cual tiene la solución $x_2=2x_1$, $x_3=3x_1$, así que los autovectores correspondientes a $\lambda=7$ son $t(1,2,3)$. Para el autovalor $\lambda=1$, el sistema $AX=X$ es

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_1 + x_2 + x_3 &= 0 \\ x_1 + x_2 + x_3 &= 0 \end{aligned}$$

Para resolver la ecuación hagamos $x_1=a$, $x_2=b$ y $x_3=-a-b$, por lo que todo autovector correspondiente a $\lambda=1$ tiene la forma:

$$(a,b,-a-b) = a(1,0,-1) + b(0,1,-1)$$

<u>Autovalor</u>	<u>Autovectores</u>	<u>dim E(λ)</u>
7	$t(1,2,3)$ $t \neq 0$	1
1, 1	$a(1,0,-1)+b(0,1,-1)$, a,b no simult. 0	2

donde $\dim E(7)=1$ significa que el autovalor 7 tiene 1 autovector asociado y $\dim E(1) = 2$ significa que los autovalores 1 tienen 2 vectores asociados (lo cual sin embargo no siempre es cierto, pero sí es cierto que la dim es al menos 1).

A4.4 Traza de una matriz.

Sea $f(\lambda)$ el polinomio característico de una matriz A , $n \times n$. Designemos las n raíces de $f(\lambda)$ por $\lambda_1, \dots, \lambda_n$, repitiendo cada raíz múltiple las veces que indique su orden de multiplicidad. Tenemos entonces la factorización

$$f(\lambda) = (\lambda - \lambda_1) \dots (\lambda - \lambda_n). \tag{A.4.11}$$

La suma de las raíces de $f(\lambda)$ se llama traza de A , y se designa con $\text{tr } A$. Así pues, por definición

$$\text{tr } A = \sum_{i=1}^n \lambda_i \quad (\text{A.4.12})$$

Una propiedad sumamente interesante es la siguiente, que podemos calcular la $\text{tr } A$ sin tener que hallar los eigenvalores, ya que es igual a la suma de los elementos diagonales de A (Apostol, ob. cit).

$$\text{tr } A = \sum_{i=1}^n a_{ii} \quad (\text{A.4.13})$$

A4.5 Matrices que representan la misma transformación lineal.
Matrices lineales.

Mostraremos que dos representaciones matriciales distintas de una transformación lineal tienen el mismo polinomio característico.

Recordemos cómo se definen las representaciones matriciales. Supongamos que $L:V \rightarrow W$ sea una aplicación de un espacio n -dimensional V en un espacio m -dimensional W . Sean (e_1, \dots, e_n) y (w_1, \dots, w_m) bases ordenadas para V y W respectivamente. La representación matricial de L relativa a las bases elegidas es la matriz $m \times n$ cuyas columnas constan de los componentes de $T(e_1), \dots, T(e_n)$ respecto a la base (w_1, \dots, w_m) . Partiendo de bases distintas se obtienen representaciones matriciales distintas.

Consideremos ahora el caso en el que $V=W$, y supongamos que se utiliza la misma base (e_1, \dots, e_n) para V y W . Sea $A=(a_{ij})$ la matriz de T relativa a esta base. Esto significa que

$$L(e_k) = \sum_{i=1}^n a_{ik} e_i, \quad k=1,2,\dots,n \quad (\text{A.4.14})$$

Elijamos ahora otra base (u_1, \dots, u_n) para V y W y sea $B=(b_{kj})$ la matriz de L relativa a esta nueva base. Entonces tenemos

$$L(u_j) = \sum_{k=1}^n b_{kj} u_k, \quad j=1,2,\dots,n \quad (\text{A.4.15})$$

ya que cada u_j pertenece al espacio engendrado por e_1, \dots, e_n podemos escribir

$$u_j = \sum_{k=1}^n c_{kj} e_k, \quad j=1,2,\dots,n \quad (\text{A.4.16})$$

La matriz $n \times n$ $C=(c_{kj})$ determinada por los escalares c_{kj} es no singular puesto que representa una transformación lineal que aplica una base de V en otra base de V . Aplicando L a los dos miembros de (A.4.16)

$$L(u_j) = \sum_{k=1}^n c_{kj} L(e_k), \quad j=1,2,\dots,n \quad (\text{A.4.17})$$

Introduzcamos las matrices

$$E=[e_1, \dots, e_n] \quad \text{y} \quad U=[u_1, \dots, u_n]$$

Entonces (A.4.16) puede reescribirse

$$U=EC \quad (A.4.18)$$

Si además

$$E'=[L(e_1), \dots, L(e_n)] \quad \text{y} \quad U'=[L(u_1), \dots, L(u_n)]$$

(A.4:14, 15, 17) se transforman en

$$E'=EA, \quad U'=UB, \quad U'=E'C \quad (A.4.19)$$

respectivamente. De (A.4.18),

$$E = UC^{-1} \quad (A.4.20)$$

con lo que

$$U'=UB=E'C=EAC=UC^{-1}AC$$

por lo tanto

$$B=C^{-1}AC \quad (A.4.21)$$

con lo que hemos demostrado el

Teorema A4.5.1: Si dos matrices $n \times n$ A y B representan la misma transformación lineal L , existe una matriz no singular C tal que $B=C^{-1}AC$

Además, si A es la matriz de L relativa a la base $E=(e_1, \dots, e_n)$ y B es la matriz de L relativa a la base $U=(u_1, \dots, u_n)$, entonces

como C podemos tomar la matriz no singular que relaciona las dos bases a través de la ecuación matricial $U=EC$.

Definición: Dos matrices $n \times n$ A y B se llaman semejantes si existe una matriz no singular C tal que $B=C^{-1}AC$.

Lema A4.5.1: Dos matrices $n \times n$ son semejantes si representan la misma transformación lineal.

Para demostrar el teorema A4.5.4 (dado posteriormente), precisamos demostrar previamente:

Teorema A4.5.2: Sea $L:V \rightarrow V$ una transformación lineal, siendo F el campo de escalares de V , y $\dim V=n$. Sea A una representación matricial de L . Entonces el conjunto de autovalores de L consta de las raíces del polinomio característico de A .

Demostración: Todo autovalor de L satisface la ecuación $\det(\lambda I - A) = 0$ y puesto que esa es la ecuación de definición del polinomio característico de A , $f(\lambda)$, de donde salen las raíces λ_i , L consta de las raíces de $f(\lambda)$.

Teorema A4.5.3: Si u_1, \dots, u_k son autovectores de una TL $L:V \rightarrow V$, y los autovalores $\lambda_1, \dots, \lambda_k$ son distintos, entonces los autovectores u_1, \dots, u_k son independientes.

Demostración: La demostración es por inducción en k . Supongamos que ha sido demostrado el teorema para cualquier conjunto $k-1$ de autovectores. Sean u_1, \dots, u_k , k autovectores pertenecientes a autovalores distintos, y supongamos que existen escalares c_1 tales que

$$\sum_{i=1}^k c_i u_i = 0 \quad (A4.22)$$

Aplicando L a los dos miembros de (A4.22) y utilizando el hecho de que $L(u_i) = \lambda_i u_i$ encontramos

$$\sum_{i=1}^k c_i \lambda_i u_i = 0 \quad (A4.23)$$

multiplicando (A4.22) * λ_k y restando de (A4.23) tenemos

$$\sum_{i=1}^{k-1} c_i (\lambda_i - \lambda_k) u_i = 0 \quad (A4.24)$$

Pero ya que u_1, \dots, u_{k-1} son independientes, debe ser $c_i(\lambda_i - \lambda_k) = 0$ para $i=1, 2, \dots, k-1$. Puesto que los autovalores son distintos tenemos $\lambda_i \neq \lambda_k$ para $i \neq k$ así que $c_i = 0$ para $i=1, \dots, k-1$. Por (A4.22) vemos que c_k es también 0, por lo que los autovectores u_1, \dots, u_k son independientes.

Procedemos entonces a demostrar el

Teorema A4.5.4: Si $L:V \rightarrow V$ es una TL, siendo F el cuerpo de escalares de V y $\dim V=n$ y suponemos que el polinomio característico de L tiene n raíces distintas $\lambda_1, \dots, \lambda_n$ en F , entonces:

- a) Los autovectores correspondientes u_1, \dots, u_n forman una base para V .
- b) La matriz de L relativa a la base $U=[u_1, \dots, u_n]$ es la matriz diagonal G que tiene los autovalores como elementos diagonales:

$$G = \text{diag}(\lambda_1, \dots, \lambda_n).$$

c) Si A es la matriz de L relativa a otra base $E=[e_1, \dots, e_n]$, entonces

$$G=C^{-1}AC,$$

donde C es la matriz no singular que relaciona las dos bases mediante la ecuación

$$U=EC.$$

Demostración: Según el teorema A4.5.2 cada raíz λ_i es un autovalor. Puesto que existen n raíces distintas, el teorema A4.5.3 nos dice que los autovectores correspondientes u_1, \dots, u_n son independientes. Luego forman una base para V (que es de $\dim=n$). Esto demuestra a). Como $L(u_i)=\lambda_i u_i$, la matriz de L relativa a U es la matriz diagonal G , lo que demuestra b). Para demostrar c) usamos el teorema A4.5.1.

Observación A4.1: La matriz no singular C del teorema anterior se llama matriz diagonalizante. Si (e_1, \dots, e_n) es la base de los vectores coordenados unitarios (I_1, \dots, I_n) , entonces la ecuación $U=EC$ demuestra que la columna k de C consta de los componentes de los autovectores u_k relativos a (I_1, \dots, I_n) .

Para aclarar los conceptos, hagamos el siguiente ejemplo:

Sea $L:R^3 \rightarrow R^3$ definida por

$$L([a_1, a_2, a_3]) = [0, a_2, a_1+a_3].$$

sea $S=\{[1,0,0], [0,1,0], [0,0,1]\}$ la base natural para R^3 . La representación de L respecto a S es

$$A = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix}. \text{ Consideremos ahora}$$

la base ordenada

$$S' = \{[1, 0, -1], [0, 1, 0], [0, 0, 1]\} \text{ para } R^3.$$

La matriz de transición de S' a S es C

$$S' = SC$$

$$\text{con } C = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix}$$

$$\text{y } C^{-1} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix}$$

entonces la representación de L respecto a S' es

$$B = C^{-1}AC = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

la cual es una matriz diagonal.

La elección de la base S' está de acuerdo a todos los lineamientos establecidos en la teoría, es decir, obtenemos los eigenvalores a partir del polinomio característico y a partir de ellos los eigenvectores. Puesto que para R^3 se obtiene 3 eigenvectores independientes, estos forman una base para R^3 , la cual genera la matriz diagonal B .

El polinomio característico se obtiene de:

$$\det(\lambda I - A) = 0$$

$$= \begin{vmatrix} \lambda & 0 & 0 \\ 0 & \lambda - 1 & 0 \\ -1 & 0 & \lambda - 1 \end{vmatrix} = 0$$

por lo que $f(\lambda) = \lambda (\lambda - 1) (\lambda - 1) = 0$

de lo que se ve que las raíces del polinomio, que por el teorema A4.5.2 son los eigenvalores, son $\lambda=0$ y $\lambda=1$ (raíz doble).

Para hallar el eigenvector asociado a $\lambda=1$ se procede de la siguiente manera. De la definición de eigenvectores:

$AX = \lambda X$, donde A es la matriz asociada a la TL L,

$$(\lambda I - A)X = 0 \quad (A.4.25)$$

para $\lambda=1$,

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

el sistema conduce a

$$\begin{aligned} a_1 &= 0 \\ -a_1 &= 0, \text{ donde } a_1=0 \text{ y } a_2, a_3 \text{ son cualquier escalar,} \end{aligned}$$

por lo que los eigenvectores asociados a $\lambda=1$ son:

$(0 \ s \ t) = s(0,1,0) + t(0,0,1)$, con s,t escalares diferentes de 0, digamos:

$$(0 \ 1 \ 0) + (0,0,1).$$

para $\lambda=0$:

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

el sistema conduce a

$$a_2 = 0$$

$$a_1 + a_3 = 0, \text{ donde } a_2=0 \text{ y } a_3=-a_1,$$

por lo que los eigenvectores asociados a $\lambda=0$ son:

$$(r \ 0 \ -r)=r(1,0,-1), \text{ con } r \text{ diferente de } 0, \text{ digamos:}$$

$(1 \ 0 \ -1)$, que conjuntamente a los eigenvectores correspondientes a $\lambda=1$ de multiplicidad dos dan la matriz C:

$$C = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix}$$

que fue usada en la deducción de la matriz diagonal.

La matriz diagonalizante C proporciona en el caso general la transformación de base. En nuestro caso, puesto que usamos la base natural para R^3 :

$$E = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

la base T de vectores u_k se obtiene de $U=EC$, pero como E es la matriz identidad, $U=C$.

U=EC demuestra que la columna k de C consta de los componentes de los autovectores u_k relativos a E, como se estableció en la observación A4.1.

A4.6 Autovalores en espacios euclídeos.

Recordemos que los espacios euclídeos se caracterizan porque poseen producto interno. Las propiedades fundamentales son:

- 1) $(x,y) = (y,x)$ (simetría)
- 2) $(x+z,y) = (x,y) + (z,y)$ (linealidad)
- 3) $(cx,y) = c(x,y)$ (homogeneidad)
- 4) $(x,x) > 0$ si $x \neq 0$ (positividad).

El siguiente teorema establece que los autovalores pueden expresarse en función del producto interior.

Teorema A4.6.1: Si E es un espacio euclídeo, V un subespacio de E y consideramos una transformación lineal $L:V \rightarrow E$ que tenga un autovalor λ con un correspondiente autovector x , entonces

$$\lambda = (T(x),x)/(x,x) \quad (A4.26)$$

Demostración: Puesto que $T(x)=\lambda x$ tenemos

$$(T(x),x)=(\lambda x,x)=\lambda(x,x)$$

como $x \neq 0$ podemos dividir por (x,x) obteniendo (A4.26).

Definición: Sean E un espacio euclídeo y V un subespacio de E . Una transformación lineal $L:V \rightarrow E$ se llama simétrica en V si

$$(T(x),y)=(x,T(y)) \text{ para todo } x \text{ y para todo } y \text{ de } V.$$

El operador L se llama hemi-simétrico en V si

$$(T(x),y)=- (x,T(y)) \text{ para todo } x \text{ y para todo } y \text{ de } V.$$

Dicho en otras palabras, un operador simétrico L puede pasarse de un factor de un producto interior al otro sin modificar el valor del producto. Tal paso en el caso de un operador hemi-simétrico cambia el signo del producto.

Ejemplo: Sea $C(a,b)$ el espacio de todas las funciones reales continuas en un intervalo cerrado $[a,b]$, con el producto interior real

$$(f,g) = \int_a^b f(t)g(t) dt$$

Sea V un subespacio de $C(a,b)$. Si $L:V \rightarrow C(a,b)$ es una TL, entonces

$$(f,T(g)) = \int_a^b f(t)T(g(t)) dt$$

Por consiguiente, las condiciones para la simetría y la hemisimetría se convierten en

$$\int_a^b \{f(t)L(g(t)) - g(t)L(f(t))\} dt = 0 \quad \text{si } L \text{ es simétrica, (A4.27)}$$

y

$$\int_a^b \{f(t)L(g(t)) + g(t)L(f(t))\} dt = 0 \quad \text{si } L \text{ es hemi-simétrica, (A4.28)}$$

con lo que para una función fija en el espacio $C(a,b)$, p , definamos $L(f) = pf$, producto de p y f .

Para esa L , la ecuación (A4.27) se satisface cualesquiera sean f y g en $C(a,b)$ puesto que el integrando es cero. Por consiguiente, la multiplicación por una función fija es un operador simétrico.

Para el operador derivada, sea V el subespacio de las funciones f que tienen derivada continua en el intervalo abierto (a,b) y que satisfacen la condición de contorno $f(a)=f(b)$. Sea $D:V \rightarrow C(a,b)$ el operador derivada dado por $D(f)=f'$.

En (A4.28),

$$\int_a^b \{f(t)g'(t)+g(t)f'(t)\}dt=0$$

$$\int_a^b (f(t)g(t))'dt=f(b)g(b)-f(a)g(a)$$

Puesto que f y g satisfacen ambas la condición de contorno, tenemos $f(b)g(b)-f(a)g(a)=0$. Así pues, la condición de contorno implica la hemi-simetría de D . Las funciones constantes son las únicas autofunciones en el subespacio V . Pertenecen al autovalor 0 por lo que el operador derivación es hemi-simétrico.

A4.7 Ortogonalidad de los autovectores correspondientes a autovalores distintos.

Los autovalores distintos de una transformación lineal cualquiera corresponden a autovectores independientes (según el teorema A4.5.3). Para las transformaciones simétricas y hemisimétricas podemos decir aún más.

Teorema A4.7.1: Si L es una transformación simétrica o hemisimétrica y si λ y Ω son autovalores distintos de T con los autovectores correspondientes x e y , entonces x e y son ortogonales; esto es $(x,y)=0$.

Demostración: Escribamos $L(x)=\lambda x$, $L(y)=\Omega y$ y comparemos los dos productos interiores $(L(x),y)$ y $(x,L(y))$. Tenemos

$$(L(x),y) = (\lambda x,y) = \lambda(x,y) \quad \text{y} \quad (x,L(y)) = (x,\Omega y) = \Omega(x,y)$$

Si L es simétrica esto nos da $\lambda(x,y)=\Omega(x,y)$. Por consiguiente $(x,y)=0$ puesto que $\lambda \neq \Omega$. Si L es hemi-simétrica obtenemos $\lambda(x,y)=-\Omega(x,y)$ que implica también $(x,y)=0$.

A4.8 Representación matricial para operadores simétricos y hemi-simétricos.

Suponemos en lo que sigue que V es un espacio euclídeo de dimensión finita. Una transformación simétrica ó hemi-simétrica puede caracterizarse por su acción sobre los elementos de una base cualquiera.

Nota: en lo que sigue trataremos con los operadores simétricos únicamente, ya que ellos aparecen en la ecuación. Además, a los operadores hemi-simétricos les corresponden siempre autovalores cero (como en el caso del operador derivación).

Teorema A4.8.1: Sean (e_1, \dots, e_n) una base ortonormal para V y $L:V \rightarrow V$ una TL. Tenemos entonces:

L es simétrica si $(L(e_j), e_i) = (e_j, L(e_i))$ para todo par i, j .

Demostración: Tomemos dos elementos cualesquiera x e y de V y expresemos cada uno en función de los elementos de la base, sean $x = \sum x_j e_j$ e $y = \sum y_i e_i$. Tenemos entonces

$$(L(x), y) = (\sum_{j=1}^n x_j L(e_j), y) = \sum_{j=1}^n x_j (L(e_j), \sum_{i=1}^n y_i e_i) = \sum_{j=1}^n \sum_{i=1}^n x_j y_i (L(e_j), e_i).$$

del mismo modo encontramos

$$(x, L(y)) = \sum_{j=1}^n \sum_{i=1}^n x_j y_i (e_j, L(e_i)).$$

por lo que la proposición se deduce al momento de estas ecuaciones.

Expresemos estos conceptos por medio de la representación matricial de L .

Teorema A4.8.2: Sean (e_1, \dots, e_n) una base ortonormal para V , y sea $A=(a_{ij})$ la representación matricial de una TL $L:V \rightarrow V$ respecto de esa base. Tenemos entonces:

L es simétrica si $a_{ij}=a_{ji}$ para todo i, j .

Demostración: Puesto que A es la matriz de L tenemos

$$L(e_j) = \sum_{k=1}^n a_{kj} e_k$$

Tomando el producto interior de $T(e_j)$ por e_i y teniendo en cuenta la linealidad del producto interior obtenemos:

$$(T(e_j), e_i) = (\sum_{k=1}^n a_{kj} e_k, e_i) = \sum_{k=1}^n a_{kj} (e_k, e_i)$$

Pero $(e_k, e_i) = 0$ salvo si $k=i$, así que la última suma se reduce a $a_{ij}(e_i, e_i) = a_{ij}$ ya que $(e_i, e_i) = 1$ (base ortonormal). Luego resulta

$$a_{ij} = (T(e_j), e_i) \quad \text{para todo } i, j$$

de igual forma,

$$a_{ji} = (e_j, T(e_i)) \quad \text{para todo } i, j.$$

Para completar la demostración, basta aplicar el teorema anterior.

Definición A4.8.1: Una matriz cuadrada $A=(a_{ij})$ se denomina simétrica si $a_{ij}=a_{ji}$ para todo i, j .

El teorema A4.8.2 establece que una TL en un espacio V de dimensión finita es simétrica si su matriz relativa a una base ortonormal es simétrica.

A4.9 Diagonalización de una matriz simétrica.

Teorema A4.9.1: Toda matriz A , $n \times n$, simétrica es semejante a la matriz diagonal $G = \text{diag}(\lambda_1, \dots, \lambda_n)$ de sus autovalores. Además tenemos

$$G = C^{-1}AC$$

en donde C es una matriz no singular cuya inversa es su traspuesta.

Demostración: Sea V el espacio de dimensión finita, y sea (e_1, \dots, e_n) la base ortonormal de vectores coordinados unitarios. Si $x = \sum x_i e_i$ e $y = \sum y_i e_i$, consideremos el producto interior dado por $(x, y) = \sum x_i y_i$. Para la matriz dada A , sea L la TL representada por A relativa a la base elegida. Si el conjunto de eigenvalores tiene n valores distintos, tenemos n eigenvectores ortonormales independientes que forman una base ortonormal (u_1, \dots, u_n) respecto a la cual L tiene una representación en matriz diagonal $G = \text{diag}(\lambda_1, \dots, \lambda_n)$ (teorema A4.5.4), siendo λ_k el autovalor

perteneciente a u_k . Puesto que tanto A como G representan L , serán semejantes, así que tenemos $G=C^{-1}AC$, donde $C=(c_{ij})$ es la matriz no singular que relaciona las dos bases:

$$[u_1, \dots, u_n] = [e_1, \dots, e_n] C$$

Esta ecuación muestra que la columna j de C consta de los componentes de u_j respecto de $[e_1, \dots, e_n]$. Por consiguiente, c_{ij} es el componente i -ésimo de u_j . El producto interno de u_j y u_i viene dado por

$$(u_j, u_i) = \sum_{k=1}^n c_{kj} c_{ki}$$

Puesto que $[u_1, \dots, u_n]$ es un conjunto ortonormal, eso demuestra que $CC^T=I$, con lo que $C^{-1} = C^T$.

Definición: una matriz cuadrada A se llama ortogonal si $AA^T=I$

Ejemplo:

Sea $A = \begin{vmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{vmatrix}$ una matriz simétrica.

El polinomio característico de A es $f(t)=(t+2)^2(t-4)$, de modo que sus valores propios son $\lambda_1=-2$, $\lambda_2=-2$ y $\lambda_3=4$.

Para hallar los vectores propios asociados con -2 resolvemos el sistema homogéneo $(-2I-A)X=0$:

$$\begin{vmatrix} -2 & -2 & -2 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

Una base para el espacio solución del sistema consta de los eigenvectores

$$v_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \quad \text{y} \quad v_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}.$$

v_1 y v_2 no son ortogonales, puesto que $(v_1, v_2) \neq 0$. Podemos usar el proceso de Gram-Schmidt tratado en el capítulo 5 para obtener una base ortonormal para el espacio solución asociado con $\lambda = -2$. Sea $u_1 = v_1$ y

$$u_2 = v_2 - (v_2, u_1) / (u_1, u_1) v_1 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} - 1/2 \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1/2 \\ -1/2 \\ 1 \end{pmatrix}.$$

Para eliminar las fracciones tomamos (usaremos la notación de los programas de computación, en que el nuevo valor reemplaza al anterior):

$$u_2 = 2u_2 = \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}.$$

El conjunto $[u_1, u_2]$ es un conjunto ortogonal. Normalizando tenemos el conjunto ortonormal (mod = módulo y sqrt = raíz cuadrada):

$$u_1 = u_1 / \text{mod } u_1 = 1/\sqrt{2} \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$$

$$u_2 = u_2 / \text{mod } u_2 = 1/\sqrt{6} \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}$$

Hallemos una base para el espacio propio asociado con $\lambda = 4$ resolviendo el sistema homogéneo $(4I - A)X = 0$

$$\begin{vmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

Una base para este espacio propio consta del vector $u_3 = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}$

Normalizando este vector, tenemos el vector propio

$$u_3 = u_3 / \text{mod } u_3 = 1/\sqrt{3} \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}$$

Puesto que los vectores propios asociados con valores propios distintos son ortogonales (lo que se puede comprobar inmediatamente entre (u_3, u_2) y (u_3, u_1)), u_3 es ortogonal a u_1 y u_2 . De este modo el conjunto $[u_1, u_2, u_3]$ es una base ortonormal para R^3 formada por los vectores propios de A . La matriz D es:

$$D = \begin{vmatrix} -1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 0 & 2/\sqrt{6} & 1/\sqrt{3} \end{vmatrix}$$

de lo cual se puede comprobar lo mencionado en la teoría, que $D^{-1} = D^T$. Y finalmente se procede a hallar la matriz diagonal G semejante a A mediante:

$$D^{-1}AD = D^TAD = \begin{vmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 4 \end{vmatrix}$$

lo que se puede comprobar fácilmente.

Finalmente hemos llegado a disponer del conocimiento necesario para entrar a analizar las Formas Cuadráticas, que las hallaremos al analizar la velocidad de convergencia del algoritmo del Filtro TDL.

A4.10 Formas Cuadráticas.

Sean V un espacio euclídeo real y $L:V \rightarrow V$ un operador simétrico. Esto significa que L puede cambiarse de un factor al otro en un producto interior.

$$(L(x), y) = (x, L(y)) \text{ para todo } x, y \text{ de } V.$$

Dada L , definamos una función de valores reales Q en V mediante la ecuación

$$Q(x) = (L(x), x).$$

La función Q se denomina forma cuadrática asociada a L . La palabra cuadrática está sugerida por el teorema siguiente que muestra que en el caso de dimensión finita $Q(x)$ es un polinomio cuadrático en los componentes de x .

Teorema A4.10.1: Sea (e_1, \dots, e_n) una base ortonormal para un espacio euclídeo V . Sean $L:V \rightarrow V$ una TL simétrica y $A=(a_{ij})$ la matriz de L relativa a esa base. Entonces la forma cuadrática $Q(x)=(L(x), x)$ está ligada a A del modo siguiente:

$$Q(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \quad \text{si } x = \sum_{i=1}^n x_i e_i. \quad (\text{A4.29})$$

Demostración: En virtud de la linealidad tenemos $L(x) = \sum x_i L(e_i)$. Por consiguiente,

$$Q(x) = \left(\sum_{i=1}^n x_i L(e_i), \sum_{j=1}^n x_j e_j \right) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j (L(e_i), e_j).$$

Esto demuestra (A4.29) ya que $a_{ij} = a_{ji} = (L(e_i), e_j)$.

Definición: Sea V un espacio euclídeo real con una base ortonormal (e_1, \dots, e_n) , y sea $A=(a_{ij})$ cualquier matriz $n \times n$ de escalares. La función de valores escalares Q definida en cada elemento $x=\sum x_i e_i$ de V por la suma doble

$$Q(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \quad (A.4.30)$$

se denomina la forma cuadrática asociada a A .

Si A es una matriz diagonal, entonces $a_{ij}=0$ para $i \neq j$ de modo que la suma (A4.30) contiene únicamente cuadrados y se puede poner en la forma más sencilla

$$Q(x) = \sum_{i=1}^n a_{ii} x_i^2 \quad (A.4.31)$$

En este caso la forma cuadrática se llama forma diagonal. La suma doble de (A4.30) también se puede expresar como un producto de tres matrices.

Teorema A4.10.2: Sea $X=[x_1, \dots, x_n]$ una matriz fila $1 \times n$, y sea $A = (a_{ij})$ una matriz $n \times n$. Entonces XAX^T es una matriz 1×1 cuyo único elemento es

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

Demostración: El producto XA es una matriz $1 \times n$, $XA=[y_1, \dots, y_n]$, en donde el elemento y_j es el producto escalar de X por la columna j de A :

$$y_j = \sum_{i=1}^n x_i a_{ij}.$$

Por consiguiente, el producto XAX^T es una matriz 1×1 cuyo único elemento es el producto escalar

$$\sum_{j=1}^n y_j x_j = \sum_{j=1}^n \left(\sum_{i=1}^n x_i a_{ij} \right) x_j = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j.$$

con lo que A4.30 queda

$$Q(x) = XAX^T \tag{A4.32}$$

A4.11 Reducción de una forma cuadrática real a forma diagonal.

Una matriz simétrica real A es semejante a la matriz diagonal $G=C^TAC$, siendo C una matriz ortogonal (teor. A4.9.1) y $G=\text{diag}(\lambda_1, \dots, \lambda_n)$. Vamos ahora a demostrar que C puede utilizarse para convertir la forma cuadrática XAX^T a una forma diagonal.

Teorema A4.11.1: Sea XAX^T una forma cuadrática asociada a una matriz simétrica real A , y sea C una matriz ortonormal ($C^{-1}=C^T$) que convierte A en una matriz diagonal $G=C^TAC$. Tenemos entonces

$$XAX^T = YGY^T = \sum_{i=1}^n \lambda_i y_i^2, \tag{A4.33}$$

donde $Y=[y_1, \dots, y_n]$ es la matriz fila $Y=XC$ (de transformación de base), y $\lambda_1, \dots, \lambda_n$ son los autovalores de A .

Demostración: Puesto que C es ortogonal tenemos que $C^{-1}=C^T$. Por consiguiente la ecuación $Y=XC$ implica $X=YC^T$, y obtenemos

$$XAX^T=(YC^T)A(YC^T)^T=Y(C^TAC)Y^T=YG Y^T$$

Observación: El teorema A4.11.1 se expresa diciendo que la TL $Y=XC$ reduce la forma cuadrática XAX^T a una forma diagonal $YG Y^T$.

Para finalizar el estudio del Algebra Matricial hasta donde la usaremos en la tesis, hagamos el siguiente ejemplo sobre formas cuadráticas.

Ejemplo: Determinar una matriz ortogonal C que reduzca la forma cuadrática $Q(x)=ax_1^2+bx_1x_2+cx_2^2=2x_1^2+4x_1x_2+5x_2^2$ a forma diagonal.

Para empezar escribamos $Q(x)=XAX^T$, siendo $A=\begin{vmatrix} a & b/2 \\ b/2 & c \end{vmatrix}=\begin{vmatrix} 2 & 2 \\ 2 & 5 \end{vmatrix}$

La matriz A tiene los autovalores $\lambda_1=1$ y $\lambda_2=6$. Los autovectores correspondientes a 1 son $t(2,-1)$, $t \neq 0$ (o $t(-2,1)$ lo que no altera la respuesta). Los pertenecientes a 6 son $t(1,2)$, $t \neq 0$. Los dos autovectores $u_1=t(2,-1)$ y $u_2=t(1,2)$ con $t=1/\sqrt{5}$ constituyen un conjunto ortonormal. Por consiguiente la matriz

$$C=1/\sqrt{5}\begin{vmatrix} 2 & 1 \\ -1 & 2 \end{vmatrix}$$

es una matriz diagonalizante para A. En este caso $C^{-1}=C^T$ como se puede comprobar inmediatamente. Fácilmente se comprueba que

$$C^TAC=\begin{vmatrix} 1 & 0 \\ 0 & 6 \end{vmatrix}$$

De aquí resulta que la forma diagonal correspondiente se construye a partir de $YG Y^T$ donde

$$YG Y^T=[y_1 \quad y_2]\begin{vmatrix} 1 & 0 \\ 0 & 6 \end{vmatrix}\begin{vmatrix} y_1 \\ y_2 \end{vmatrix}=y_1^2+6y_2^2=\lambda_1 y_1^2+\lambda_2 y_2^2$$

Este resultado tiene una interpretación geométrica sencilla, representado en la fig 5. La TL $Y=XC$ puede considerarse como una rotación que aplica la base i, j en la nueva base u_1, u_2 . Un punto que respecto a la primera base tiene coordenadas (x_1, x_2) tiene las nuevas coordenadas (y_1, y_2) respecto a la segunda base. Puesto que $XAX^T=YG Y^T$, el conjunto de puntos (x_1, x_2) que satisfacen la ecuación $XAX^T=c$ para un cierto c es idéntico al conjunto de puntos (y_1, y_2) que satisfacen $YG Y^T=c$. La segunda ecuación, escrita en la forma $y_1^2+6y_2^2=c$, es la ecuación cartesiana de una elipse si $c>0$. La fig. 5 muestra la elipse correspondiente a $c=9$.

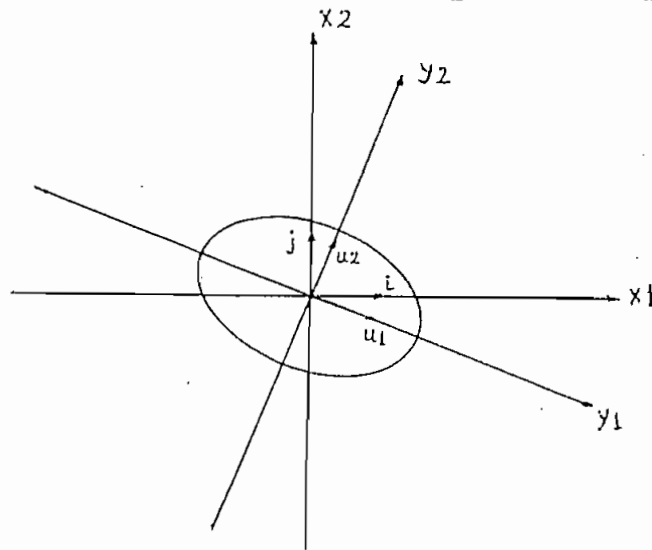


Figura 5 Rotación de ejes mediante una matriz ortogonal. La elipse tiene la ecuación cartesiana $XAX^T=9$ en el sistema x_1x_2 y la ecuación $YG Y^T$ en el sistema y_1y_2 .

APENDICE V.

APENDICE V

MANUAL DEL USUARIO.

A continuación explicaremos brevemente las operaciones básicas del Menú de Ingreso, de modo de que la utilización del programa sea lo más óptima posible.

Previamente, sin embargo, debemos detallar los archivos que contienen los diskettes que se entregan con los programas ejecutables y de datos.

Los archivos son (para el diskette #1):

- 1.- alias.txt : archivo de macros;
- 2.- data.txt : archivo generado por com_cadc.exe;
- 3.- inicia.dat : archivo generado por com_cad.exe;
- 4.- malla.dat : nombres de las variables y rangos recomendados para el Filtro de Malla;
- 5.- noeq.dat : idem para Transmisión sin ecualización;
- 6.- tdl.dat : idem para el Filtro Transversal;
- 7.- com_cad.dat : formatos de pantalla;
- 8.- graf.dat : tipo de gráfico;
- 9.- com_cad.exe : menú introductorio;
- 10.- com_cadc.exe : programa en lenguaje C++.
- 11.- grafico.wql : macro del QPRO.
- 12.- assign.com. : asigna el drive b: al a:

El segundo diskette contiene el archivo del lenguaje fuente en C++ de Borland:

- 13.- com_cad.cpp

La máquina que se va a utilizar debe poseer QPRO v.3.0, con el siguiente directorio:

c:\qpro\files

Debe poseer además la versión 6.0 del Norton Utilities con el siguiente directorio:

c:\nu

(Las máquinas del Laboratorio de Computación poseen tanto QPRO 3.0 cuanto NU 6.0). Verifíquese que el path incluya al NU y al QPRO, en caso contrario añada al path los siguientes directorios:

c:\nu;c:\qpro;c:\qpro\files

En vista de que se usa el comando keystack del NU 6.0, SE DEBE INCLUIR LA ORDEN DEVICE = C:\NU\KEYSTACK.SYS en el config.sys. Además, deben existir las órdenes files=20 y buffers=20 en dicho config.sys.

Estamos entonces en condiciones de inicializar el sistema.

Para ello:

- 1.- Introdúzcase el diskette de programas ejecutables (diskette 1) en el drive a.
- 2.- Si la máquina no posee drive b:, teclee:
a:\assign b=a
- 3.- Seguidamente verifique que existan los archivos ndos.com y ndos.ovl en c:\NU.
- 4.- Una vez con los dos archivos mencionados a su disposición teclee:
ndos.com

esto instala el intérprete de comandos del NU en modalidad shell. Para salir (cuando acabe de trabajar), teclee exit.

5.- Hecho esto teclee

```
alias/r a:\aliases.txt
```

con lo cual se cargan los macros contenidos en el archivo alias.es.txt. Nos interesa el macro start, el cual será utilizado para la graficación en el QPRO.

6.- Ahora teclee

```
a:\com_cad
```

con lo cual se presenta en pantalla el menú de introducción.

7.- Elija la opción deseada para transmisión en el menú Tipo de Transmisión. Esto le conduce a la pantalla de edición, en la cual se puede escoger:

- la velocidad de modulación baud;
- la semilla sem;
- la frecuencia de portadora fp;
- el instante de decisión decis;
- el número de bits transmitidos;
- el número de secuencias de entrenamiento entrena;
- el coeficiente mu;
- el parámetro epsilon;
- el parámetro w;
- el número de coeficientes y
- la relación señal a ruido.

8.- Una vez inicializadas las variables, se pasa a la ventana de Graficación. En ella se puede elegir el tipo de gráfico a efectuar, de 9 posibles:

- bits transmitidos;
- espectro NRZ;
- bits recibidos;
- Convergencia del error del Filtro Transversal;

- Convergencia del error del Filtro de Malla;
- Convergencia de los coeficientes del Filtro Transversal;
- Convergencia de los coeficientes del Filtro de Malla;
- Bits ecualizados con el Filtro Transversal; y
- Bits ecualizados con el Filtro de Malla.

El menú está configurado de tal forma que permite el acceso de datos moviendo las flechas del cursor para posicionarse adecuadamente y a continuación introducir el valor deseado. Si finalmente uno no desea cambiar los valores, sale de la inicialización con escape, y si desea cambiar, se lo hace con F2.

El programa fue diseñado de modo de que el tipo de gráfico sea definible sólo si está previamente definido el tipo de Transmisión. De no estarlo sale en pantalla el mensaje de que previamente se debe definir el tipo de Transmisión.

Si bien es cierto que en pantalla se muestran los rangos más adecuados (según aconseja el CCITT para transmisión en la práctica), es posible introducir valores entre 1 y 999999. Pero si uno intenta introducir caracteres que no sean números, el programa no lo acepta.

En el menú existe la actualización de la hora en la esquina superior derecha cada 8 seg.

Una vez que se ha elegido el tipo de transmisión y el tipo de gráfico, nos movemos a la ventana GO, en la cual se graba la inicialización al archivo a:\inicia.dat y automáticamente arranca el programa en lenguaje C++, llamado com_cadc.exe.

Si no se desea reinicializar las variables (es decir, se desea correr el programa con los antiguos valores), simplemente no entramos a la ventana Tipo de Transmisión ni a Graficación y directamente vamos a GO.

Si no se desea ningún tipo de gráfico, desde Tipo de Transmisión se puede ir directamente a GO. En este caso, antes de correr Com_cadc.exe, el programa da la opción de regresar a reinicializar el tipo de gráfico (lo cual requiere reinicializar también el Tipo de Transmisión).

Si por alguna razón se desea terminar la ejecución de com_cadc.exe, teclear Control C o Control Break.

Una vez que se ha corrido exitosamente com_cadc.exe, nos resta graficar en QPRO. Para el efecto tecleamos:

```
start "nombre del gráfico".
```

Por ejemplo, si hemos elegido graficar los bits transmitidos podemos teclear:

```
start btx o simplemente st btx.
```

El nombre "btx" es el nombre con el que se grabará el archivo data.txt, generado por com_cadc.exe, en el directorio c:\qpro\files.

De esta manera, el macro start, que lo listamos a continuación:

```
ST*ART=pushd ^c:^pushd ^ cd\qpro\files^ keystack "/fr" 27 27  
"a:\grafico.wq1" 13 @71"/fd:c:\qpro\files\13"/tic" 27 "b:\da-  
ta.txt" 13 @34 13 27 "/fa"27 27 27 "c:\qpro\files\%1" 13 ^  
q^popd^popd
```

es multifuncional.

Primero guarda el directorio actual con pushd, luego cambia al drive c:.. Guarda el directorio presente en c: y cambia a qpro\files; alimenta al teclado con la orden file retrieve a:\grafico.wq1 y cambia de directorio a c:\qpro\files. Luego importa b:\data.txt, arranca el macro grafico.wq1 y almacena