

01170

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE INGENIERIA  
DIVISION DE ESTUDIOS DE POSGRADO**

**TESIS**

**Estudio y Comparación de Algoritmos de  
Filtrado Adaptable para la Igualación de Canal**

**PRESENTADA POR**

**JOSE EDUARDO TORRES FERNANDEZ**

**PARA OBTENER EL GRADO DE**

**MAESTRO EN INGENIERIA  
(ELECTRICA)**

**DIRIGIDA POR:**

**Dr. ROGELIO ALCANTARA SILVA**

**Ciudad Universitaria, Junio de 1997**

**TESIS CON  
VALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Resumen

En la actualidad, las sociedades modernas experimentan la creciente necesidad de transmitir información a altas tasas en tiempo real, por esta razón en los países desarrollados se realizan grandes esfuerzos por comercializar aplicaciones tales como el teléfono de manos libres, el video-teléfono, los sistemas personales de comunicaciones mediante enlaces satelitales y las comunicaciones móviles en general. Estas aplicaciones requieren de la transmisión a tasas mayores que las que permite el ancho de banda utilizable del canal, por lo cual se impone el uso de técnicas de igualación de canal. Una de las técnicas más populares en la actualidad para igualar canal lo constituyen los algoritmos adaptables, los cuales han impuesto su hegemonía en este campo debido principalmente a que no requieren de un conocimiento a priori de la función de transferencia del canal, ni de los estadísticos de las señales transmitidas y, además pueden modificar la función de respuesta al impulso del filtro igualador para compensar los cambios en la función de transferencia del canal.

Este trabajo tiene como objetivo estudiar y evaluar el desempeño individual de los algoritmos adaptables más populares para diversos canales tanto a fase mínima como a fase no mínima. Debido a la amplia gama de algoritmos que existen se seleccionaron los algoritmos que han gozado de una aceptación "universal": se escogió el algoritmo "normalized least mean squares" ya que actualmente está consuetudinario en una especie de "patrón" contra el cual se comparan los algoritmos de inspiración más reciente; se escogieron dos métodos representativos de los mínimos cuadrados: "fast transversal filter" y el "least squares lattice" y, finalmente se escogió un algoritmo de filtrado IIR como lo es el "simple hyperstable adaptive recursive filter".

Además se compara el desempeño de estos algoritmos con el objeto de recomendar cual de todos estos métodos exhibe las mejores propiedades para su aplicación en la práctica de acuerdo a los requerimientos del enlace de comunicaciones.

Como resultado de las evaluaciones realizadas se concluye que los canales más difíciles de igualar son aquellos que presentan "antiresonancias" en su respuesta en frecuencia y, que además son de fase no mínima. Esto debido a que el filtro igualador requerido es a respuesta impulsional finita y de un orden muy elevado para poder igualar el canal de manera satisfactoria. Al elevar el orden del filtro el número de operaciones flotantes por muestra de salida sufre un incremento no conveniente para la transmisión de datos a tasas elevadas y en tiempo real, por lo que se impone el uso de algoritmos rápidos tales como el FTF o el LSL. De estos dos últimos se prefiere al LSL debido a su robustez numérica sin embargo, dado que el FTF realiza menos operaciones por muestra de salida se optará por este último cuando la tasa de transmisión sea de vital importancia para el enlace de comunicaciones.

# Indice

<b>I</b>	<b>Filtrado e Igualación de Canal</b>	<b>14</b>
<b>1</b>	<b>Generalidades de la Igualación de Canal</b>	<b>15</b>
1.1	Antecedentes Históricos. . . . .	16
1.2	Sistemas Modernos para la Transmisión de Datos. . . . .	16
1.3	Efectos de la Transmisión a través de Canales no Ideales. . . . .	17
1.4	Caracterización de los Canales de Propagación. . . . .	19
1.5	Importancia del Igualador Adaptivo . . . . .	21
1.6	Estructuras Básicas de los Igualadores de Canal. . . . .	22
1.6.1	El Igualador Lineal. . . . .	22
1.6.1.1	El Algoritmo "Zero Forcing" (ZF) . . . . .	23
1.6.2	Igualador Retroalimentado ó "Decision Feedback Equalizer" (DFE) . . . . .	24
1.7	Igualadores Fraccionalmente Espaciados (FSE) . . . . .	25
1.8	Igualadores para Señales en Cuadratura . . . . .	26
1.9	Consideraciones sobre la Longitud del Igualador. . . . .	27
1.10	Solución Global al Problema de la Igualación de Canal . . . . .	28
1.11	Conclusiones . . . . .	28
<b>II</b>	<b>Algoritmos Recursivos Para La Igualación de Canal</b>	<b>30</b>
<b>2</b>	<b>Algoritmos De Gradiente Estocástico</b>	<b>31</b>
2.1	Algoritmo del Steepest Descent . . . . .	31
2.1.1	El Algoritmo "Steepest Descent". . . . .	33
2.1.2	Organigrama del Algoritmo Steepest Descent . . . . .	34
2.2	Algoritmo LMS ("Least Mean Squares"). . . . .	35
2.2.1	Estructura del Algoritmo "Least Mean Squares" . . . . .	35
2.2.2	Algoritmo LMS . . . . .	35
2.2.3	Organigrama del Algoritmo LMS. . . . .	36
2.2.4	Propiedades del Algoritmo LMS. . . . .	36
2.3	Conclusiones . . . . .	37

<b>3</b>	<b>Algoritmo de los Mínimos Cuadrados (MC)</b>	<b>39</b>
3.1	Desarrollo del Algoritmo de los Mínimos Cuadrados (MC)	39
3.1.1	La Función de Costo en el Algoritmo de los Mínimos Cuadrados	40
3.1.2	Derivación de las Ecuaciones Normales	41
3.1.2.1	Principio de la Ortogonalidad	41
3.1.2.2	Ecuaciones Normales	42
3.1.3	Propiedades del Algoritmo de Mínimos Cuadrados	43
3.2	Algoritmo Adaptable de los Mínimos Cuadrados	43
3.2.1	Desarrollo del Algoritmo MCR	44
3.2.1.1	Recursión para el Vector de Coeficientes del Filtro	46
3.2.2	Organigrama del Algoritmo de Mínimos Cuadrados Recursivos	48
3.3	Algoritmos Rápidos para La Igualación de Canal	49
3.3.1	Principios Básicos	49
3.3.1.1	Predicción Adaptiva Lineal Hacia Adelante	49
3.3.1.2	Predicción Lineal Adaptiva Hacia Atras	52
3.3.1.3	Recursión para el Vector de Ganancia	57
3.3.1.4	Factor de Conversión (Angulo Variable)	57
3.3.1.5	Resumen de Fórmulas	60
3.3.2	Algoritmos Rápidos: Estructura Transversal	62
3.3.2.1	Desarrollo del Algoritmo FTF	62
3.3.2.2	Organigrama del Algoritmo FTF	66
3.3.3	Algoritmos Rápidos: Estructura "Lattice"	66
3.3.3.1	Antecedentes	67
3.3.3.2	Recursiones en el Orden	68
3.3.3.3	Relación entre los Parámetros $\Delta_{m-1}(n)$ y $\Delta'_{m-1}(n)$	70
3.3.3.4	Recursiones en el Orden para los Errores A Posteriori de Predicción	70
3.3.3.5	Recursiones en el Orden para las Sumas Ponderadas del Cuadrado de los Errores A Posteriori de Predicción	72
3.3.3.6	Recursión en el Orden para $\gamma_m(n-1)$	73
3.3.3.7	Recursiones en el Tiempo	74
3.3.3.8	Proceso de Estimación Conjunta	76
3.3.3.9	Organigrama del Algoritmo LSL	84
3.3.3.10	Algoritmo LSL Usando Errores A Priori de Estimación	85
3.3.3.11	Organigrama del Algoritmo LSL Usando Errores A Priori de Estimación	87
3.4	Conclusiones	89
<b>4</b>	<b>Algoritmos Adaptables Para Filtros A Respuesta Impulsional Infinita</b>	<b>90</b>
4.1	Antecedentes Históricos	91
4.2	Formulación del Problema del Filtrado Adaptable IIR	92
4.2.1	Método de la Ecuación de Error	92

4.2.2	Método del Error de Salida . . . . .	94
4.2.2.1	Superficies de Error para el Método del Error de Salida . . . . .	96
4.2.3	Problemas Relativos a los Métodos de la Ecuación de Error y el Error de Salida . . . . .	97
4.3	Forma General de los Algoritmos para Filtrado Adaptable a Respuesta Impulsional Infinita . . . . .	98
4.4	El Algoritmo HARF . . . . .	100
4.4.1	Introducción a la Teoría de la Hyperestabilidad . . . . .	100
4.4.2	El Algoritmo HARF . . . . .	103
4.4.3	Organigrama del Algoritmo HARF. . . . .	105
4.4.4	El Algoritmo SARF . . . . .	106
4.4.5	Organigrama del Algoritmo SARF. . . . .	107
4.5	Algoritmos SIM, AFM e IF . . . . .	108
4.5.1	"Least Mean Square Error Equation" . . . . .	108
4.5.2	Desarrollo de los Algoritmos SIM, AFM e IF . . . . .	110
4.5.3	Organigrama de los Algoritmos SIM, AFM e IF . . . . .	113
4.6	Realizaciones Alternativas . . . . .	115
4.6.1	Estructura en Paralelo . . . . .	115
4.6.2	Estructura "Lattice" . . . . .	115
4.7	Conclusiones . . . . .	115

### **III Implementación de los Algoritmos 117**

<b>5</b>	<b>Evaluación de los Algoritmos</b>	<b>118</b>
5.1	Descripción de las Evaluaciones . . . . .	119
5.1.1	Comentarios Acerca de las Evaluaciones . . . . .	119
5.1.2	Canales Utilizados en las Evaluaciones . . . . .	120
5.2	Presentación de Resultados . . . . .	122
5.2.1	Evaluaciones con el Algoritmo SARF . . . . .	122
5.2.1.1	Evaluaciones con el Algoritmo SARF para Identificar el Canal . . . . .	123
5.2.1.2	Evaluaciones con el Algoritmo SARF para la Igualación de Canal . . . . .	128
5.2.2	Evaluaciones con el Algoritmo "Normalized Least Mean Squares (NLMS)" . . . . .	133
5.2.2.1	Canales de Fase Mínima . . . . .	135
5.2.2.2	Canales de Fase No Mínima . . . . .	141
5.2.3	Evaluaciones con el Algoritmo "Fast Transversal Filters FTF" . . . . .	148
5.2.3.1	Canales de Fase Mínima . . . . .	149
5.2.3.2	Canales de Fase no Mínima . . . . .	151
5.2.4	Evaluaciones con el Algoritmo Least "Lattice" Squares (LSL) con Errores A Posteriori de Estimación . . . . .	153

5.3	Conclusiones . . . . .	151
<b>6</b>	<b>Comparación de los Algoritmos</b>	<b>156</b>
6.1	Comparación de Resultados . . . . .	156
6.1.1	Canales de Fase Mínima . . . . .	158
6.1.1.1	Canal $H_1(z)$ . . . . .	158
6.1.1.2	Canal $H_2$ . . . . .	160
6.1.1.3	Canal $H_5(z)$ . . . . .	163
6.1.2	Canales de Fase no Mínima . . . . .	164
6.1.2.1	Presentación de Gráficas . . . . .	164
6.2	Conclusiones . . . . .	171
<b>IV</b>	<b>Conclusiones y Perspectivas</b>	<b>172</b>
<b>V</b>	<b>Apéndices</b>	<b>176</b>
<b>A</b>	<b>Fundamentos del Algebra Lineal</b>	<b>177</b>
A.1	Espacios Vectoriales . . . . .	177
A.1.1	Espacios Euclidianos . . . . .	178
A.1.2	Espacios de Hilbert . . . . .	178
A.1.3	Matrices . . . . .	178
A.2	Independencia Lineal . . . . .	179
A.2.1	Determinante Gramiano . . . . .	179
A.3	Factores QR . . . . .	179
A.3.1	Procedimiento de Gram-Schmidt . . . . .	180
A.3.2	Transformación Householder . . . . .	181
A.3.3	Transformación de Givens . . . . .	182
A.4	Espacios Lineales . . . . .	183
A.4.1	Bases . . . . .	183
A.4.2	Subespacios Directos . . . . .	184
A.4.3	Dimensión, Rango y Nulidad . . . . .	184
A.5	Matrices Hermiteanas . . . . .	185
A.6	Descomposición en Valores Singulares . . . . .	185
A.6.1	Rango y Espacio Nulo . . . . .	187
A.6.2	Aproximación de Bajo Rango . . . . .	187
A.6.3	Descomposición de la Matriz Identidad . . . . .	188
A.7	Proyecciones, Rotaciones y Pseudo-inversas . . . . .	189
A.7.1	Proyecciones . . . . .	189
A.7.2	Rotaciones . . . . .	189
A.7.3	La Pseudo-inversa . . . . .	190
A.7.4	Representaciones Ortogonales . . . . .	190
A.8	Fórmulas de Inversión de Matrices . . . . .	192

# Indice de figuras

1.1	Modelo de un Sistema de Comunicaciones. . . . .	17
1.2	Modelo para un Canal de Propagación lineal. . . . .	20
1.3	“Linear decision feedback equalizers”. . . . .	23
2.1	Estructura de un filtro transversal adaptivo. . . . .	32
3.1	Predictor hacia adelante de orden $M$ . . . . .	50
3.2	Filtro de Predicción del Error. . . . .	51
3.3	Predictor hacia atrás de orden $M$ . . . . .	54
3.4	Filtro de Predicción hacia atrás del error. . . . .	55
3.5	Filtro transversal utilizado para definir el error de estimación $\gamma_M(n)$ . . . . .	58
3.6	Etapa del Filtro “Lattice”. . . . .	72
3.7	Filtro “Lattice”. . . . .	73
3.8	Proceso de estimación conjunta. . . . .	77
3.9	Filtro transversal. . . . .	79
4.1	Método del Error de Salida. . . . .	95
4.2	Sistema de lazo cerrado a excitación nula. . . . .	102
4.3	Implementación del algoritmo HARF. . . . .	104
4.4	Modo de Identificación de sistemas para el algoritmo SIM. . . . .	109
4.5	Modo de filtrado adaptivo para el algoritmo AFM. . . . .	112
5.1	Migración de polos para el algoritmo SARF. Se utilizo un filtro de seis polos y cuatro ceros. . . . .	125
5.2	Migración de polos para el algoritmo SARF. Se utilizo un filtro de dos polos únicamente. . . . .	126
5.3	Respuestas en frecuencia del canal y del igualador como resultado de aplicar el algoritmo SARF para igualar el canal $H_5(z)$ . Línea continua (Canal). . . . .	130
5.4	Correlación cruzada de la señal de entrada al canal con la señal de salida del mismo como resultado de aplicar el algoritmo SARF para igualar el canal $H_5(z)$ . . . . .	131
5.5	Autocorrelación de la señal de entrada al canal como resultado de aplicar el algoritmo SARF para igualar el canal $H_5(z)$ . . . . .	132

5.6	Correlación de la señal de entrada al canal con la señal de salida del canal. . . . .	133
5.7	Respuesta en frecuencia del igualador y del canal.Línea continua (Canal).135	
5.8	Energía del vector de coeficientes.Método nlms.Canál $H_1(z)$ . . . . .	137
5.9	Energía del vector de coeficientes.Método nlms.Canál $H_2(z)$ . . . . .	138
5.10	Respuesta en frecuencia del igualador y del canal.Algoritmo NLMS con $\mu = 1.5$ y de 0.09 a partir de las 1050 iteraciones.Canál $H_2(z)$ . . .	139
5.11	Respuestas en frecuencia tanto del canal como del igualador.Línea punteada (Canal).Algoritmo NLMS con $\mu = 1.0$ y de 0.09 a un tercio del número total de iteraciones.Canál $H_3(z)$ . . . . .	140
5.12	Convulación de la respuesta al impulso del canal con la del igualador.Algoritmo NLMS con $\mu = 1.0$ y de 0.09.Canál $H_3(z)$ . Línea continua (Canal). . . . .	141
5.13	Convulación delas respuestas al impulso del canal y del igualador.Algoritmo NLMS con $\mu = 1.0$ y de 0.09 a un tercio del número total de iteraciones.Canál $H_4(z)$ . . . . .	143
5.14	Respuesta en frecuencia del canal y del igualador.Algoritmo NLMS con $\mu = 1.0$ y de 0.09 a las 500 iteraciones.Canál $H_6(z)$ .Línea continua (Canal). . . . .	144
5.15	Respuesta en frecuencia del canal y del igualador.Algoritmo NLMS con $\mu = 1.5$ y de 0.09 a las 1200 iteraciones.Canál $H_7(z)$ .Línea continua (Canal). . . . .	145
5.16	Respuesta en frecuencia del canal y del igualador.Algoritmo NLMS con $\mu = 1.5$ y de 0.09 a las 2750 iteraciones.Canál $H_8(z)$ .Línea continua (Canal). . . . .	147
6.1	Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo SARF para igualar el canal $H_1(z)$ .Línea punteada (igualador).159	
6.2	Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo SARF para igualar el canal $H_2(z)$ .Línea punteada (igualador).160	
6.3	Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo FTF para igualar el canal $H_2(z)$ .Línea punteada (igualador).161	
6.4	Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo NLMS para igualar el canal $H_4(z)$ . . . .	165
6.5	Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo FTF para igualar el canal $H_4(z)$ . . . .	166
6.6	Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo LSL para igualar el canal $H_4(z)$ . . . .	167
6.7	Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo NLMS para igualar el canal $H_6(z)$ . . . .	168
6.8	Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo FTF para igualar el canal $H_6(z)$ . . . .	169

- 6.9 Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo LSL para igualar el canal  $H_6(z)$  . . . . 170

# Indice de Tablas

2.1	El método del "Steepest Descent" . . . . .	34
2.2	Algoritmo LMS . . . . .	36
3.1	El algoritmo de Mínimos Cuadrados Recursivos (MCR). . . . .	48
3.2	Analogía de parámetros entre el algoritmo de mínimos cuadrados recursivos, la predicción lineal hacia adelante y la predicción lineal hacia atrás. . . . .	53
3.3	Predicción lineal Adaptiva hacia adelante. . . . .	60
3.4	Predicción lineal Adaptiva hacia atrás. . . . .	60
3.5	Vector de Ganancia. . . . .	61
3.6	Factor de Conversión . . . . .	61
3.7	Organigrama del algoritmo FTF . . . . .	66
3.8	Organigrama del algoritmo LSL. . . . .	84
3.9	Inicialización del algoritmo LSL. . . . .	85
3.10	Organigrama del algoritmo LSL usando errores a priori de estimación. . . . .	87
3.11	Inicialización del algoritmo LSL usando errores a priori de estimación. . . . .	88
4.1	Algoritmo HARF. . . . .	105
4.2	Algoritmo SARF. . . . .	107
4.3	Algoritmo SIM. . . . .	113
4.4	Algoritmo AFM. . . . .	114
4.5	Algoritmo HF. . . . .	114
5.1	Resultados obtenidos de aplicar el algoritmo SARF para identificar $H_9$ . . . . .	124
5.2	Resultados obtenidos de variar el parámetro $c_1$ del filtrado sobre la señal del error de salida en el algoritmo SARF, utilizado a su vez para identificar $H_9$ . . . . .	127
5.3	Resultados obtenidos de aplicar el algoritmo SARF para igualar los canales $H_1, H_2$ . . . . .	129
5.4	Número de operaciones flotantes por muestra de salida para el algoritmo NLMS . . . . .	134
5.5	Número de operaciones flotantes por muestra de salida para el algoritmo FTF . . . . .	148
5.6	Número de operaciones flotantes por muestra de salida para el algoritmo LSL . . . . .	154

6.1	Número de operaciones flotantes por muestra de salida para los algoritmos NLMS, FTF y LSL . . . . .	157
-----	---	-----

# Introducción

La igualación de canal es un tema crucial dentro del contexto de las comunicaciones digitales modernas debido a la creciente necesidad de la sociedad moderna de transmitir información en tiempo real a altas velocidades de transmisión de un punto a otro.

El problema básico para poder transmitir grandes cantidades de información en tiempo real a través de un medio físico radica principalmente en las restricciones impuestas por la propia naturaleza del medio a emplear. Estas limitaciones físicas propias del medio imponen una severa restricción en el ancho de banda que es aprovechable para poder transmitir información. En caso de contar con un ancho de banda infinito, se podría transmitir a una tasa infinita por lo cual no existiría la necesidad de desarrollar técnicas para poder transmitir a tasas por arriba de las permitidas por el canal. Sin embargo como este no es el caso, se impone buscar alguna técnica para poder transmitir en tiempo real a tasas superiores a la que nos permite el canal.

La idea fundamental es igualar el canal mediante un filtro cuya respuesta en frecuencia sea la inversa a la del canal, de tal manera que la acción del canal se anule. Esta idea es bastante sencilla de concebir sin embargo para llevarla a la práctica existen varios problemas que resolver. Primero, si deseamos diseñar un filtro que tenga una respuesta en frecuencia inversa a la del canal, resulta lógico que se debe conocer la respuesta en frecuencia del canal, de lo contrario es imposible diseñar el filtro requerido. El segundo problema parte del hecho de que estamos suponiendo que la función de transferencia del canal es de fase mínima, pues de lo contrario el filtro resultante será inestable, lo cual no es tolerable. Por último y aunque de momento parezca descabellado se debe de disponer de un método que considere las posibles variaciones de la función de transferencia del canal. Este último punto es de importancia fundamental y está relacionado con el segundo punto, ya que si el canal es de fase mínima pero debido a cierta causa esta condición ya no sigue siendo válida entonces el filtro igualador será inestable.

Con el objeto de ilustrar la posibilidad de la variación del patrón de polos y ceros de la función de transferencia de un canal considere un sistema de comunicaciones móviles. En este caso existirá el fenómeno llamado propagación multitraectoria el cual consiste en que la señal transmitida viaja a lo largo de diferentes trayectorias; por lo que el receptor recibirá una combinación de replicas de la misma señal pero con distinto retraso. El problema surge cuando por alguna circunstancia una o más

# Introducción

La igualación de canal es un tema crucial dentro del contexto de las comunicaciones digitales modernas debido a la creciente necesidad de la sociedad moderna de transmitir información en tiempo real a altas velocidades de transmisión de un punto a otro.

El problema básico para poder transmitir grandes cantidades de información en tiempo real a través de un medio físico radica principalmente en las restricciones impuestas por la propia naturaleza del medio a emplear. Estas limitaciones físicas propias del medio imponen una severa restricción en el ancho de banda que es aprovechable para poder transmitir información. En caso de contar con un ancho de banda infinito, se podría transmitir a una tasa infinita por lo cual no existiría la necesidad de desarrollar técnicas para poder transmitir a tasas por arriba de las permitidas por el canal. Sin embargo como este no es el caso, se impone buscar alguna técnica para poder transmitir en tiempo real a tasas superiores a la que nos permite el canal.

La idea fundamental es igualar el canal mediante un filtro cuya respuesta en frecuencia sea la inversa a la del canal, de tal manera que la acción del canal se anule. Esta idea es bastante sencilla de concebir sin embargo para llevarla a la práctica existen varios problemas que resolver. Primero, si deseamos diseñar un filtro que tenga una respuesta en frecuencia inversa a la del canal, resulta lógico que se debe conocer la respuesta en frecuencia del canal, de lo contrario es imposible diseñar el filtro requerido. El segundo problema parte del hecho de que estamos suponiendo que la función de transferencia del canal es de fase mínima, pues de lo contrario el filtro resultante será inestable, lo cual no es tolerable. Por último y aunque de momento parezca descabellado se debe de disponer de un método que considere las posibles variaciones de la función de transferencia del canal. Este último punto es de importancia fundamental y está relacionado con el segundo punto, ya que si el canal es de fase mínima pero debido a cierta causa esta condición ya no sigue siendo válida entonces el filtro igualador será inestable.

Con el objeto de ilustrar la posibilidad de la variación del patrón de polos y ceros de la función de transferencia de un canal considere un sistema de comunicaciones móviles. En este caso existirá el fenómeno llamado propagación multitrayectoria el cual consiste en que la señal transmitida viaja a lo largo de diferentes trayectorias; por lo que el receptor recibirá una combinación de replicas de la misma señal pero con distinto retraso. El problema surge cuando por alguna circunstancia una o más

de las señales que no se propagan a través de la línea de vista tienen una potencia mayor que la señal que sí se propaga a través de la línea de vista: se puede demostrar que esto ocasionará la aparición de ceros fuera del círculo unitario en la función de transferencia del canal lo que ocasionará que el sistema deje de ser de fase mínima.

A lo largo del desarrollo de las telecomunicaciones se han empleado diversas soluciones, las cuales como resulta natural van cediendo su lugar a soluciones más eficientes conforme el desarrollo tecnológico alcanza nuevos horizontes.

Históricamente una de las primeras soluciones adoptadas consistió en emplear un igualador fijo cuyo diseño se basaba en la elaboración de un modelo para el canal que se deseaba igualar. Esta solución no puede compensar los efectos debidos a las variaciones del canal y, dado que en general la función de transferencia del canal es desconocida resulta ser un método bastante ineficiente y no permite la transmisión de datos a tasas muy elevadas de transmisión por lo cual este método no perduró en la práctica.

Otro desarrollo notable lo constituyó la aparición del algoritmo de "zero forcing". Este algoritmo compensa la distorsión de pico que se presenta a la entrada del receptor. Sin embargo uno de los inconvenientes de este método es que no puede compensar los efectos del canal si la distorsión de pico en el receptor es mayor del 100%, lo cual corresponde a situaciones en las que la acción del canal sobre la señal transmitida no es muy severa. A pesar de la significancia de este método en el desarrollo de las comunicaciones y debido a la necesidad cada vez más imperiosa de elevar aun más la tasa de transmisión este método tuvo que dejar su lugar a otras técnicas más eficientes.

En la actualidad las técnicas más aceptadas en la práctica las constituyen los algoritmos adaptables: los cuales sí pueden efectuar un seguimiento de las variaciones de la función de transferencia del canal. El objetivo de este trabajo consiste en el análisis de los algoritmos más importantes empleados para igualar canal.

Dentro de estos algoritmos y debido a la importancia que han tenido en el desarrollo de las telecomunicaciones existe una gran variedad de los mismos. Además de lo cual existen principalmente dos estructuras para realizar la implantación del filtro igualador: ya sea que se prefiera una estructura transversal o una estructura lattice. Debido a esta gran variedad de algoritmos y a las posibles formas de llevar a cabo su implantación práctica resulta muy laborioso sino imposible realizar un análisis al detalle de cada algoritmo del cual se tiene conocimiento a través de la literatura referente al tema de la igualación de canal. De aquí la imperiosa necesidad de establecer claramente los fines y propósitos del presente trabajo.

El presente trabajo se centrará de una manera especial en los algoritmos: "least mean squares", los algoritmos rápidos de mínimos cuadrados tanto para una estructura transversal como lattice del filtro igualador; y finalmente se tratarán los algoritmos que retroalimentan los símbolos ya decodificados con el objeto de mejorar la igualación.

En el capítulo uno se abordará de una manera general el problema de la igualación en el contexto de las telecomunicaciones tratando de dar una perspec-

tiva del problema así como englobar las diversas soluciones al mismo.

En la parte dos de este trabajo se tratarán en detalle los algoritmos más populares utilizados en la actualidad para igualar canal. Como se mencionó anteriormente, debido a la gran importancia que reviste este problema existe un número sin fin de algoritmos y variantes de los mismos dificultando de esta manera la tarea de analizar cada uno de estos por separado y paso a paso; por lo tanto se prefirió tratar los más importantes en el sentido tanto de su importancia teórica como práctica.

Cabe señalar que dentro de estos algoritmos también se analizaron los algoritmos que utilizan un filtro de respuesta impulsional infinita para igualar el canal. Como veremos en el capítulo cuatro estos algoritmos tienen importantes ventajas sobre su contraparte de respuesta impulsional finita. Sin embargo, el desarrollo matemático es mucho más complicado y aun no se ha logrado resolver satisfactoriamente los problemas que conlleva la utilización de un filtro de respuesta impulsional para igualar el canal. Además la limitación más seria de estos filtros lo constituye el hecho de que nada más pueden ser utilizados para igualar canales de fase mínima mientras que por otro lado la gran mayoría de los canales que se presentan en la práctica son de fase no mínima lo cual limita la utilidad de un algoritmo de este tipo.

En la parte tres se presentan los resultados de efectuar las distintas evaluaciones y comparaciones entre los siguientes algoritmos: "normalized least mean squares", "fast transversal filter algorithm", "least squares lattice algorithm" y el "simple hyperstable adaptive recursive filter". Finalmente se presentan las conclusiones y perspectivas que constituyen el punto medular de este trabajo.

**Parte I**

**Filtrado e Igualación de Canal**

# Capítulo 1

## Generalidades de la Igualación de Canal

En este capítulo se da un panorama general del problema del filtrado adaptable aplicado en la igualación de canal. como primer punto se hace un énfasis en la importancia de la igualación de canal. para lo cual se abordan algunos aspectos históricos que nos servirán para situar el problema de la igualación de canal a través del tiempo y, además lo más importante consiste en precisar como ha evolucionado este problema conforme las crecientes necesidades de transmitir una mayor cantidad de datos en tiempo real aumenta día con día en las sociedades modernas.

A continuación se establece que la necesidad de igualar el canal proviene principalmente de que éste se comporta como un filtro. Por lo tanto el canal, en general, tanto atenuará como distorsionará a la señal que se propaga a través de él. De esta manera se ven afectas las características de las formas de onda que son especialmente diseñadas para detectar los símbolos transmitidos a través del canal. Aunado a esto si consideramos la necesidad de transmitir señales a través del canal a tasa mayores a las permitidas por el ancho de banda se impone la imperiosa necesidad de igualar el canal con el fin de restaurar las señales tal cual fueron transmitidas para poder efectuar la detección en el receptor.

A continuación se hace énfasis en que cualquier canal puede ser representado por un "tapped delay line filter". de esta manera la señal en el receptor es una combinación lineal no solo del símbolo presente sino también de los contiguos (esto se le conoce como interferencia intersímbolos (ISI)), lo cual dificulta la detección en el receptor.

Después se hace énfasis en el hecho de que para poder transmitir a altas tasa de transmisión se debe de poder seguir las variaciones en el canal, para lo cual es necesario un igualador adaptivo; de lo contrario no se pueden lograr altas tasas de transmisión.

a continuación se estudian las diversas estructuras para los igualadores lineales; lo cual no significa que se estén menospreciando los igualadores no lineales basados en el criterio de la máxima verosimilitud del símbolo transmitido que además se

combinan con el algoritmo de vitterbi para lograr resultados sorprendentes [bSH94]. Esta clase de algoritmos no se han abordado en este trabajo debido a la necesidad de mantener a este dentro de un límite razonable para su extensión.

Finalmente, se menciona brevemente que para poder solucionar el problema de la igualación de canal se requiere de la inversión de matrices; sin embargo como se verá en la parte dos de este trabajo existen métodos recursivos que solucionan este problema sin necesidad de realizar la inversión matricial.

## 1.1 Antecedentes Históricos.

La revolución en las comunicaciones digitales puede fecharse desde la invención de la igualación automática y adaptiva de canal a finales de la década de los sesenta [GHW92]. El concepto de un igualador lineal, digital y adaptivo de canal fue introducido en la década de los sesenta. En aquél entonces la señal recibida era muestreada a la tasa de símbolos; lo cual era suficiente para identificar la función de respuesta al impulso de canales de fase no mínima pero con función de respuesta al impulso finita [TFJ96]. Algoritmos de igualación de canal con y sin entrenamiento fueron introducidos en los años sesenta. Los algoritmos con entrenamiento rápidamente se volvieron los más populares en aplicaciones tales como la telefonía.

Los años setenta presenciaron el surgimiento de los igualadores de canal fraccionalmente espaciados, este término se debe a que este tipo de equalizadores requieren que la señal en el receptor sea muestreada a una frecuencia mayor a la tasa de símbolos [TFJ96]. Cabe hacer notar que los algoritmos de adaptación casi no requirieron modificaciones para su implementación dentro de este nuevo esquema.

Durante la década de los ochenta, la igualación ciega paso de la teoría a la práctica; pero no fué sino hasta los noventa cuando la igualación ciega fué incorporada a aplicaciones tales como la televisión por cable [TFJ96].

## 1.2 Sistemas Modernos para la Transmisión de Datos.

La figura 1.1 muestra un diagrama de bloques de un sistema de comunicaciones digitales. Los datos son alimentados al modulador, el cual convierte cada dato en una señal analógica de banda limitada y, cuya frecuencia es la adecuada para la transmisión. Al propagarse la señal hacia el receptor, esta se retrasa, se atenúa y, algunas veces sufre una distorsión que depende de su frecuencia [TFJ96]. Todos estos efectos están considerados en el modelo del canal, mientras que el ruido y la interferencia se piensa, para efectos de modelado, que están presentes a la entrada del receptor, el cual intenta recuperar la secuencia original de datos.

La transmisión de datos moderna está basada en la idea de transmitir pulsos, los cuales tienen un ancho de banda limitado [Bin88]. Los datos de entrada

generalmente son particionados en conjuntos de  $N$  bits. Estos bits son usados para

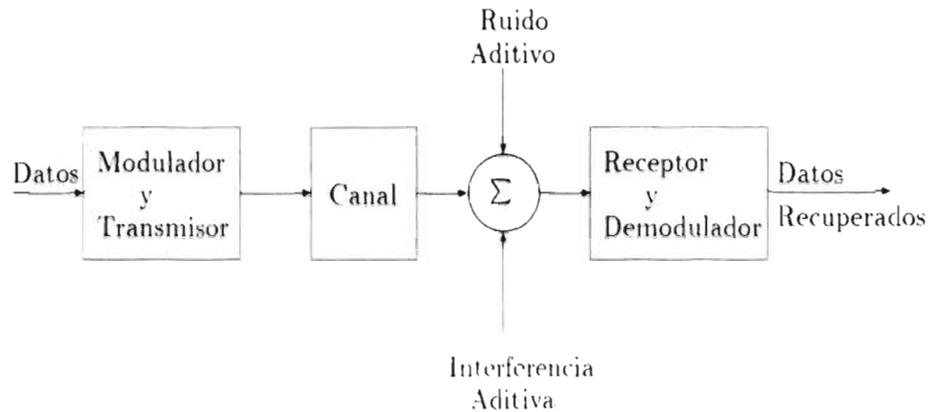


Figura 1.1: Modelo de un Sistema de Comunicaciones.

determinar la fase, frecuencia o amplitud de pico del pulso que será transmitido. La forma del pulso se debe escoger de forma tal que su espectro sea de banda limitada. El receptor está diseñado para determinar la amplitud, fase y/o frecuencia de cada pulso recibido; a partir de esta información determinará cual de los  $2^N$  símbolos fue transmitido y, finalmente nos dará la combinación de bits que corresponde a dicho símbolo. La forma del pulso está especialmente diseñada para transmitir los pulsos tan rápido como sea posible y que todavía el receptor pueda distinguirlos [TFJ96]. De aquí se puede vislumbrar que el tipo de canal utilizado para establecer la comunicación impondrá ciertas restricciones que deberán observarse para la selección de la tasa de transmisión de datos, en el tipo de modulación empleada, en el tipo de codificación empleada para corregir y detectar errores, etc.

### 1.3 Efectos de la Transmisión a través de Canales no Ideales.

La limitante más seria de cualquier canal de comunicaciones no ideal es su ancho de banda limitado y, aunque éste no fuera el caso existen otras consideraciones de índole técnica que restringen el ancho de banda disponible tales como el multiplexaje en frecuencia comúnmente utilizado para dar servicio a varios usuarios al mismo tiempo, lo cual implica que el ancho de banda disponible para un solo usuario se ve reducido proporcionalmente con el número de usuarios con el objeto de evitar la interferencia ó crosstalking entre éstos. Finalmente, en la práctica se deben respetar las normas existentes que regulan el ancho de banda del cual se puede disponer para cierta aplicación.

Considere las siguientes definiciones: se dice que un canal  $C$  con ancho de banda  $W$ , no distorsiona la señal si:

$$|C(f)| \leq \text{constante}, \quad \forall f \leq W \quad (1.1)$$

y,

$$\tau(f) = -\frac{1}{2\pi} \frac{d\theta(f)}{df} = \text{constante} \quad \forall f \leq W \quad (1.2)$$

donde  $|C(f)|$  y  $\theta(f)$  son la magnitud y la fase de la respuesta en frecuencia del canal. En caso de que no se cumpla (1.1) se dice que el canal distorsiona la señal transmitida en amplitud y, en caso de que no se cumpla (1.2) se dice que el canal distorsiona la señal transmitida en fase. Dado que la fase a una determinada frecuencia,  $f_0$ , se puede asociar con el retardo de propagación que sufre una señal senoidal de frecuencia  $f_0$  a través de canal: el hecho de que la fase de la respuesta en frecuencia del canal no sea constante implica que la señal sufrirá un retardo distinto según sea su contenido frecuencial [Pro95].

Como resultado de la distorsión en amplitud y fase causada por la respuesta en frecuencia de un canal no ideal, la transmisión de una sucesión de pulsos, a través de dicho canal, a una tasa de transmisión comparable con el ancho de banda del canal causará un "ensanchamiento" del soporte temporal de cada uno de los pulsos transmitidos de tal manera que la detección de los símbolos transmitidos por parte del demodulador se vuelve más difícil conforme más alta es la tasa de transmisión hasta llegar a un punto en el cual es imposible determinar que símbolo fue transmitido; en esta última condición se dice que los símbolos sufren un aliasing temporal severo ó bien se dice que existe una interferencia intersímbolo significativa [Pro95]. Como un ejemplo de la interferencia intersímbolo considere una señal modulada en PAM, donde la forma básica del pulso a transmitir es diseñada de tal forma que sus cruces por cero ocurran en  $\pm T, \pm 2T$ , etc. De esta manera en un canal sin distorsión es posible transmitir el siguiente símbolo PAM justo cuando ocurre el primer cruce por cero del anterior, esto es posible ya que la señal PAM conveja la información en la amplitud. Por otro lado para un canal con distorsión en fase los cruces por cero del pulso recibido ya no ocurren en  $\pm T, \pm 2T$ , etc. ni tampoco están equiespaciados temporalmente, por tanto en estos instantes de tiempo tendremos un aliasing temporal de pulsos. En caso de que este aliasing temporal de pulsos sea muy pronunciado es posible que los picos de los pulsos no sean discernibles, por lo tanto, el demodulador no podrá llevar a cabo su tarea de una manera satisfactoria. Este aliasing temporal de pulsos lo que se conoce como interferencia intersímbolos [Pro95]. Matemáticamente la interferencia intersímbolo puede expresarse de la siguiente manera: dado que la señal recibida  $r(t)$  es la superposición de las respuestas al impulso de cada uno de los símbolos transmitidos, la podemos expresar como:

$$r(t) = \sum_j x_j h(t - jT) + n(t) \quad (1.3)$$

donde  $n(t)$  representa una señal utilizada para modelar el ruido en el canal. La señal recibida debe ser muestreada en los instantes dados por:  $kT + t_0$ , donde  $t_0$  se requiere

para tomar en cuenta los efectos de retardo de propagación de la señal a través del canal y, el tiempo inicio de la acción del muestreador. De esta manera al muestrear la señal  $r(t)$  tenemos:

$$r(t_o + kT) = x_k h(t_o) + \sum_{j \neq k} x_j h(t_o + kT - jT) + n(t_o + kT) \quad (1.4)$$

El primer término en la derecha es la señal deseada dado que puede ser usada para identificar la amplitud del pulso transmitido. El último término representa el ruido aditivo, mientras que el segundo término representa la interferencia de los símbolos adyacentes [Quy82]. Cada término de la suma es directamente proporcional a una muestra de la respuesta al impulso del canal dada por  $h(t_o + iT)$ . La interferencia intersímbolo (ISI) es igual a cero si y sólo si  $h(t_o + iT) = 0$  para  $i \neq 0$ ; esto es, si la respuesta al impulso del canal presenta cruces por cero a intervalos de  $T$ .

Cuando la respuesta al impulso tiene cruces por cero uniformemente espaciados, se dice que satisface el primer criterio de Nyquist [Quy82]. En términos del dominio de la frecuencia, ésta condición es equivalente a:

$$C'(f) = \text{constante para } |f| \leq 1/2T \quad (1.5)$$

donde  $C(f)$  es la respuesta en frecuencia del canal y  $C'(f)$  es la respuesta en frecuencia del canal después de haber realizado el muestreo a la tasa de símbolos. A la banda de frecuencias dada por  $|f| \leq 1/2T$  se le llama ancho de banda de Nyquist ó ancho de banda mínimo.

A continuación se describe otro efecto que sufre la señal transmitida: considere el caso de una transmisión de datos mediante una antena omnidireccional. Esto significa que la antena receptora no es lo suficientemente selectiva para suprimir la recepción de distintas versiones de la señal transmitida, que son reflejadas por características topográficas del terreno y por distintos objetos hechos por el hombre; a este fenómeno se le conoce como propagación multitrayectoria. Por tanto el demodulador tendrá que decidir que símbolo fue transmitido a partir de la señal recibida, la cual es una suma de las distintas versiones retrasadas y atenuadas de la señal transmitida.

Todos los efectos que sufre la señal transmitida son considerados dentro del modelo del canal (filtro). Al filtrar un pulso, éste tiende a ensancharse en el dominio del tiempo. Este ensanchamiento dependerá de las características del filtro (en este caso del canal) [TFJ96]. Si esta distorsión es lo suficientemente grande el demodulador producirá una secuencia de símbolos errónea.

## 1.4 Caracterización de los Canales de Propagación.

Bello fue el primero en introducir el modelo de un canal de propagación como un "tapped delay line filter" del tipo que se muestra en la figura 1.2 [Bel63]. Bello mostró que prácticamente cualquier canal de propagación real puede ser adecuadamente

modelado mediante la selección adecuada de los espaciamientos del "tap" y los coeficientes de ponderación. En la selección de los espaciamientos del tap, cabe hacer notar que dichos espaciamientos deben ser menores al inverso del ancho de banda de la señal transmitida, es decir, se debe satisfacer el teorema de Nyquist [TFJ96]. Como resultado es común modelar al canal de propagación como un filtro de respuesta impulsional finita (FIR), cuyos "taps" están espaciados por el periodo de muestreo y los coeficientes de ponderación son seleccionados para modelar de una manera adecuada la respuesta al impulso del canal [TFJ96].

Bello introdujo dos conceptos de utilidad el "delay spread" y el "Doppler spread" [Bel63]. El "delay spread" de un canal es el soporte temporal de un canal FIR, y mide el grado del ensanchamiento temporal de un pulso al ser transmitido a través del canal. El Doppler spread mide el grado del ensanchamiento espectral de la señal transmitida. El delay spread puede ser visto como la duración de la respuesta del canal a la función impulso, mientras que el Doppler spread puede ser visto como el ancho del espectro de la señal de salida del canal cuando la entrada al canal es una senoide.

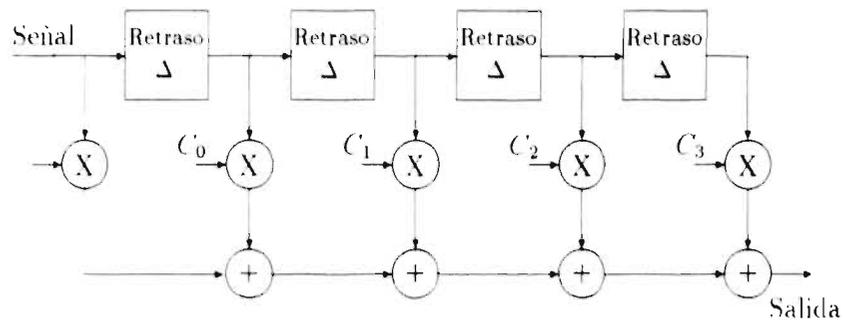


Figura 1.2: Modelo para un Canal de Propagación lineal.

## 1.5 Importancia del Igualador Adaptivo

El diseño del igualador adaptivo toma especial importancia por tres razones [TFJ96]: (1) su desempeño es crucial en la maximización de la tasa de transmisión, (2) es la parte más complicada del demodulador, y (3) consume la mayor parte del tiempo de procesamiento que le toma al demodulador efectuar su tarea. Retomando el último punto es importante limitar la longitud del igualador de tal manera que se logre un desempeño adecuado del demodulador [TFJ96].

Debido a la necesidad de reducir el tiempo de procesamiento del igualador de canal, los primeros demoduladores emplearon los "T-spaced equalizers" [Bin88]. Después de las operaciones de filtrado, control de ganancia y remoción de portadora, la señal de entrada era muestreada una vez por símbolo (pulso) cada T segundos. El periodo de muestreo era ajustado de tal manera que las muestras fueran tomadas en el "top dead center" de los pulsos recibidos [TFJ96]. Estas muestras son las que el filtro de igualación procesa. La salida del igualador es una combinación lineal de dichas muestras: esta salida se alimenta a las etapas de medición y comparación. Las mediciones del error hechas en el circuito de decisión son retroalimentadas al algoritmo de adaptación que utiliza el igualador, de esta manera se optimiza la selección de los coeficientes de ponderación del filtro de igualación.

Aunque es teóricamente razonable y computacionalmente eficiente mostrar a la tasa de símbolos, el diseño práctico de modems de alta velocidad se ha alejado de los "T-spaced equalizers" dando un giro hacia los igualadores fraccionalmente espaciados (FSE de sus siglas en inglés), llamados de esta manera debido a que el espaciamiento temporal de los taps del igualador es una fracción del inverso de la tasa de símbolos (duración del símbolo, la cual denotamos con T) [TFJ96]. Por lo tanto, la entrada al igualador es muestreada a una tasa mayor que la tasa de símbolos,  $f_B$ . La salida del igualador tiene una tasa igual a la tasa de símbolos, por lo tanto el igualador fraccionalmente espaciado realiza una decimación o remuestreo [TFJ96].

Dado que los igualadores fraccionalmente espaciados consumen mayor tiempo de procesamiento que los "T-spaced equalizers", surge la pregunta: ¿por qué utilizar un igualador fraccionalmente espaciado?. La razón principal para preferir un igualador fraccionalmente espaciado sobre un "T-spaced equalizer" es que a pesar de que los pulsos tienen una tasa  $f_B$ , el ancho de banda de la señal es un poco mayor que  $f_B$ , típicamente de un 10% a un 40% mayor [TFJ96]. Como resultado, muestrear la señal de entrada a la tasa de símbolos,  $f_B$  Hz, no es suficiente para satisfacer el teorema de Nyquist. Esta consideración no es de importancia en el caso de que todos los parámetros de la señal sean conocidos, pero este no es el caso ya que la señal recibida debe ser procesada para extraer con exactitud información apartir de la cual se debe estimar sobre la fase y la portadora de la señal transmitida. En conclusión muestrear la señal recibida a la tasa de símbolos no es suficiente [TFJ96].

La elección más común para la tasa de muestreo de la señal recibida es simplemente muestrear a dos veces la tasa de símbolos, es decir a  $2f_B$ , de tal manera

que el espaciamiento temporal entre los taps del filtro igualador sea igual a mitad de la duración del símbolo transmitido, es decir  $T/2$  [TFJ96]. Por tanto, si no se desea operar a una tasa fraccional la entrada al igualador debe ser decimada por un factor de dos, produciendo una muestra de salida por cada dos muestras de entrada [TFJ96].

## 1.6 Estructuras Básicas de los Igualadores de Canal.

Dos estructuras básicas de igualación son empleadas en la práctica [TFJ96]. La primera que se muestra en la figura 1.3.a, es conocida como igualador lineal y, se localiza en cascada entre el canal y el bloque de medición de pulsos. El igualador es lineal debido a que su salida es una combinación lineal de la entrada (la cual es a su vez una combinación de distintas versiones retrasadas y atenuadas de la señal de entrada al canal). El Igualador de decisión retroalimentada ó "decision feedback equalizer" (DFE), mostrado en la figura 1.3.b, es un procesador no lineal dado que la entrada al bloque de medición de pulsos consiste de una señal filtrada de entrada y una versión filtrada de la salida del circuito de decisión [TFJ96]. A continuación se da una breve descripción de ambas estructuras de los igualadores arriba mencionados.

### 1.6.1 El Igualador Lineal.

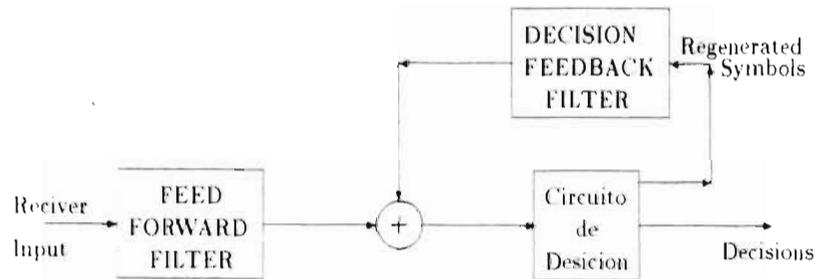
Existen dos tipos de estructuras o topología para los igualadores lineales no recursivos: los igualadores lineales que utilizan una estructura en lattice y aquellos que utilizan una estructura transversal, también llamada "tapped delay line", como la que se muestra en la figura 1.2. La siguiente discusión toma como base el igualador lineal transversal no recursivo, mostrado en la figura 1.2. Obsérvese que esta estructura es exactamente igual a la que se utiliza para modelar al canal, lo único que cambia son los valores de los coeficientes, los cuales representan la respuesta al impulso del canal ó la del igualador según sea el caso.

El igualador transversal mostrado en la figura 1.2 es lineal dado que su salida puede expresarse como una combinación lineal del valor presente de la señal recibida,  $r(t)$ , y de sus valores pasados  $r(t - nT)$ .

El igualador transversal adaptivo se puede implementar de una manera analógica o bien de una manera digital. Para el caso de una implementación analógica, la señal de salida del igualador  $z(t)$  es muestreada a la tasa de símbolos; las muestras resultantes,  $z(nT)$ , se utilizan para decidir que símbolo fue transmitido. Por otro lado, si la implementación es digital, la señal recibida  $r(t)$  se muestra a la tasa de símbolos; las muestras resultantes  $r(nT)$  se almacenan en un registro de corrimiento o memoria y, la salida del igualador se calcula digitalmente, una vez por símbolo,



(a)



(b)

Figura 1.3: "Linear decision feedback equalizaers".

de acuerdo a:

$$z_k = \sum_{n=0}^{N-1} c_n r(t_o + kT - nt) \quad (1.6)$$

Existen principalmente tres criterios para ajustar el valor de los coeficientes del igualador: el algoritmo de "zero forcing" (ZF), el algoritmo "least mean squares" (LMS) y, el algoritmo de mínimos cuadrados (LS). Dado que más adelante se abordarán en detalle los algoritmo LMS y los LS, los cuales son los más utilizados en la práctica, a continuación se describe con cierto detalle el algoritmo ZF.

### 1.6.1.1 El Algoritmo "Zero Forcing" (ZF)

El algoritmo ZF tiene como objetivo reducir la distorsión pico a la salida del igualador, la cual se define de la siguiente manera: la distorsión pico es el peor caso de interferencia de intersímbolos a la salida del igualador. La minimización de este índice de desempeño se conoce con el nombre de criterio de la distorsión pico. El criterio de la distorsión pico para un igualador de longitud finita  $2K+1$  está dado

por:

$$\mathcal{D} = \sum_{\substack{n = -K \\ n \neq 0}}^{K+L-1} |q_n| = \sum_{\substack{n = -K \\ n \neq 0}}^{K+L-1} \left| \sum_j c_j * f_{n-j} \right| \quad (1.7)$$

donde  $c_n$  y  $f_n$  son los coeficientes de la respuesta al impulso del igualador y del canal respectivamente. Cabe hacer notar que debido a la longitud finita del igualador siempre se tendrá una interferencia intersímbolos residual, aun cuando se los coeficientes de la respuesta al impulso del igualador sean los optimos de acuerdo al criterio de la distorsión pico dada por la ecuación 1.7; dicho de otra forma se requiere de un igualador de longitud infinita para eliminar la interferencia intersímbolo completamente [Pro95].

Los coeficientes  $c_n$  se determinan al minimizar la ecuación (1.7), para lo cual no existe un procedimiento computacionalmente sencillo excepto para el caso en el que la distorsión a la entrada del igualador esté dada por:

$$\mathcal{D}_0 = \frac{1}{|f_0|} \sum_{n=1}^L |f_n| \quad (1.8)$$

donde  $\mathcal{D}_0$  sea menor a la unidad. El algoritmo utilizado para minimizar la ecuación anterior recibe el nombre de "zero forcing". Este consiste en forzar que la correlación cruzada entre el error  $e_k = \mathbf{I}_k - \hat{\mathbf{I}}_k$  y la información  $\{\mathbf{I}_k\}$  sea cero para desplazamientos de  $0 \leq |n| \leq K$ . Aquí se omite el procedimiento que conduce a la siguiente recursión para actualizar los parámetros del igualador [Pro95]:

$$c_j^{(k+1)} = c_j^{(k)} + \Delta c_k \mathbf{I}_{k-j}, \quad j = -K, \dots, -1, 0, 1, \dots, K \quad (1.9)$$

donde  $c_j^{(k)}$  es el valor del j-ésimo coeficiente al tiempo  $t = kT$ ,  $c_k = \mathbf{I}_k = \hat{\mathbf{I}}_k$  es la señal de error al tiempo  $t = kT$ , y  $\Delta$  es un factor de escala que controla la tasa de adaptación.

Finalmente cabe hacer la siguiente observación con respecto al algoritmo ZF: la condición de distorsión pico  $\mathcal{D}_0 \leq 1$ , para la cual está diseñado el algoritmo ZF, representa una interferencia de intersímbolos no muy severa, por lo cual no es de interés para aquellos casos donde la velocidad de transmisión es mucho mayor que el ancho de banda del canal, por ello el algoritmo ZF fue relegado de la práctica por los algoritmos LMS y LS, los cuales presentan un desempeño superior.

### 1.6.2 Igualador Retroalimentado ó "Decision Feedback Equalizer" (DFE)

En los igualadores DFE la señal de salida es la suma de las salidas de las partes forward y feedback del igualador. La parte forward del igualador es exactamente igual

a la del igualador lineal transversal que se discutió en la sección anterior. Las decisiones tomadas sobre la señal igualada son retroalimentadas mediante un segundo filtro transversal. La idea básica es que si los símbolos detectados con anterioridad son correctos, entonces la contribución de estos símbolos a la ISI puede ser cancelada de una manera exacta, al sustraer de la salida del igualador los símbolos pasados con los apropiados coeficientes de ponderación. Estos coeficientes de ponderación son muestras de la "cola" de la respuesta al impulso del canal y de la parte forward del igualador [Quy82].

Los coeficientes forward y feedback pueden ser ajustados de una manera simultánea para minimizar el error cuadrático medio. La recursión para los coeficientes feedback es exactamente igual a la del igualador lineal:

$$b_m(k+1) = b_m(k) - \Delta \epsilon_k d_{k-m}, \quad m = 1, \dots, M \quad (1.10)$$

donde  $d_k$  es el  $k$ -ésimo símbolo ya equalizado,  $b_m(k)$  es el  $m$ -ésimo coeficiente feedback al tiempo  $k$ . Cabe hacer mención que para la obtención de la ecuación anterior se utilizó el criterio de minimizar el error cuadrático medio.

Dado un número igual de coeficientes para un igualador lineal y un igualador DFE, cabría preguntarse: ¿Si el igualador DFE logra un error cuadrático medio menor que el logrado por un igualador lineal?. No existe una respuesta definitiva a ésta pregunta. El desempeño de cada tipo de igualador está influenciado por las características del canal, fase del muestreador, así como también del número de coeficientes del igualador y la posición o referencia del principal tap del igualador. Sin embargo, un igualador DFE puede compensar distorsiones de amplitud sin realzar el ruido tal y como lo hace el igualador lineal. Además el igualador DFE es menos sensitivo a la fase del muestreador [Quy82].

## 1.7 Igualadores Fraccionalmente Espaciados (FSE)

Hasta este punto se ha considerado que la separación entre los taps del igualador es igual a la duración  $T$  de los símbolos. Sin embargo en un igualador del tipo FSE la separación de sus taps es una fracción de  $T$ , de aquí el nombre de fraccionalmente espaciados (FSE). El espaciamiento entre taps  $\tau$  se selecciona de tal manera que el ancho de banda  $BW$  ocupado por la señal a la entrada del igualador sea:

$$|BW| \leq 1/2\tau \quad (1.11)$$

Esta última ecuación implica que el muestreo a la tasa  $\tau$  realizado a la señal de entrada al igualador satisfase el teorema del muestreo. En una implementación analógica no existe restricción sobre  $\tau$  mientras que en una implementación digital  $\tau$  esta dada por:

$$\tau = KT/M \quad (1.12)$$

donde  $K$  y  $M$  son enteros tales que  $M > K$ . En la práctica, por conveniencia, se escoge  $\tau = T/M$ , donde  $M$  es un entero pequeño; por ejemplo supóngase que la señal recibida es muestreada y alimentada al igualador a la tasa de  $2/T$ , entonces la salida del igualador es producida a la tasa de símbolos (una muestra de salida del igualador por cada dos muestras que entran al mismo). Matemáticamente:

$$z_k = \sum_{n=0}^{N-1} c_n r(t_\sigma + kT - nT/2) \quad (1.13)$$

Los coeficientes de un equalizador  $T/2$  pueden ser actualizados una vez por símbolo de acuerdo con el error calculado para dicho símbolo según la siguiente recursión:

$$c_n(k+1) = c_n(k) - \Delta e_k r(t_\sigma + kT - nT/2), \quad n = 0, 1, \dots, N-1 \quad (1.14)$$

Una propiedad importante de los igualadores fraccionalmente espaciados (FSE) consiste en que su desempeño no es sensible a la fase del muestreador. Sin embargo el mínimo valor del error cuadrático medio que se puede lograr con igualador fraccionalmente espaciado si es una función de la fase del muestreados [Quy82]. Por último cabe señalar que dado que el espaciamiento  $\tau$  cumple con el teorema del muestreo, no existe traslape espectral a la entrada de un igualador FSE.

Finalmente, la literatura reporta una comparación ente un equalizador espaciado a la tasa de símbolos y uno espaciado a  $T/2$  [Quy82], la modulación utilizada fue QAM y se transmitió en canales telefónicos. Los resultados reportados son los siguientes: (a) el igualador  $T/2$  con el mismo número de coeficientes que el igualador  $T$ , tiene un desempeño tan bueno o mejor que éste último; (b) no se requiere de un filtro de preigualación para el caso del igualador  $T/2$ ; (c) para canales con una severa distorsión de fase alrededor de su frecuencia máxima y mínima de su banda de paso, el igualador  $T/2$  se desempeña mucho mejor que el igualador  $T$  sin importar cual sea la fase del muestreador.

## 1.8 Igualadores para Señales en Cuadratura

Hasta este punto sólo se han discutido igualadores para sistemas de comunicaciones que utilizan PAM en banda base, sin embargo los modems que operan a altas tasa de transmisión (mayores a 4800 bits/s) utilizan una modulación QAM. En los sistema que utilizan este tipo de modulación es conveniente representar a la componente en fase y a la componente en cuadratura de la señal recibida como la parte real y la parte imaginaria del proceso complejo  $y(t)$ . De aquí que se presenten dos formas distintas de implementar el igualador de canal: la primera forma consiste en que el igualador opere en banda base sobre la señal compleja  $y(t)$ , para lo cual la señal recibida  $r(t)$  debe ser desplazada en frecuencia a banda base (multiplicar a  $r(t)$  por una exponencial compleja) de esta manera se genera una señal compleja  $y(t)$ , la cuál es alimentada al igualador para producir la salida compleja  $z(t) = z_r(t) + jz_i(t)$ . Dado

que el igualador opera sobre señales complejas se dice que es un igualador complejo. Por otro lado, el igualador dentro de un sistema de comunicaciones que utilice un esquema de modulación QAM también puede ser implementado en la banda de paso, es decir, antes de que la demodulación se lleve a cabo. En este caso la señal en el receptor  $r(t)$  debe ser descompuesta en sus componentes de fase y en cuadratura mediante un par de filtros de partición de fase, es decir, con respuestas en magnitud iguales pero respuestas en fase que difieren en  $90^\circ$ . La señal compleja obtenida a la salida de estos filtros es muestreada a la tasa de símbolos y alimentada al igualador de la misma manera que para el caso en banda base. La salida compleja  $z(nT)$  del igualador debe ser demodulada (multiplicada por una exponencial compleja), para poder decidir que símbolo fue transmitido y de acuerdo a esta decisión calcular el error, el cual a su vez debe ser modulado antes de que sea utilizado por el igualador complejo. La principal ventaja de este tipo de implementación es que la señal de error puede ser retroalimentada sin retraso para corregir la fase, por lo tanto, los "saltos" indeseables en la fase pueden ser corregidos de una forma más efectiva. La misma ventaja se puede obtener con un igualador en banda base si se utiliza un lazo de seguimiento de "saltos" de fase después del igualador [Quy82].

## 1.9 Consideraciones sobre la Longitud del Igualador.

La necesidad del espaciado fraccional en un diseño práctico de alto desempeño impacta de una manera directa a la carga computacional del demodulador. Por lo que surge la pregunta: ¿qué tan largo debe ser el equalizador, de tal manera que se compense adecuadamente la dispersión del canal?. En realidad no existe una respuesta definitiva a este problema, por lo que la investigación al respecto continúa abierta [TFJ96]. Sólo se mencionará, que la longitud del igualador depende del tipo de canal que se desee igualar.

En la práctica es común seleccionar la longitud del canal de acuerdo a cualquiera de los dos siguientes métodos empíricos: el primero consiste en simplemente construir un prototipo de un canal real y compararlo contra una gran variedad de canales reales; el segundo método consiste en emplear reglas empíricas que están lo bastante probadas en la práctica. Estas reglas generalmente presuponen que la respuesta al impulso del equalizador convergerá, al terminar el proceso de adaptación, con la suficiente exactitud al inverso de la respuesta al impulso del canal. En caso de que el canal sea adecuadamente modelado mediante una respuesta al impulso finita de longitud  $L_c$ , las reglas empíricas que generalmente se aplican para determinar la longitud del equalizador requerirán que la longitud del equalizador  $L_e$  sea varias veces mayor que el "delay spread" del canal, tal vez de tres a cinco veces  $L_c$ . Sin embargo, en el caso de los equalizadores fraccionalmente espaciados esta longitud resulta muy grande, al menos teóricamente [TFJ96]; por lo que una reducción de la longitud del equalizador permitirá reducir la carga computacional sin afectar su desempeño.

## 1.10 Solución Global al Problema de la Igualación de Canal

La solución al problema de la igualación puede plantearse como la solución al problema de la minimización de:

$$\epsilon(n) = y(n) - d(n) \quad (1.15)$$

donde  $d(n)$  es la respuesta deseada,  $y(n)$  es la estimación de la señal transmitida y  $\epsilon(n)$  es el error de estimación. De esta manera se desea determinar los coeficientes de un filtro tal que al filtrar la señal recibida  $u(n)$  nos de por resultado el estimado de la señal transmitida. Aquí cabe señalar que se utilizara filtro de respuesta impulsional finita dado que el desarrollo matemático es más sencillo.

Los coeficientes del filtro de estimación se encuentran al minimizar el error de estimación dado por 1.15. Cuando se logra esto, el error de estimación es ortogonal a la señal estimada  $y(n)$ :

$$E[y(n)\epsilon^*(n)] = 0 \quad (1.16)$$

donde  $y_o$  y  $\epsilon_o$  son los valores óptimos para los cuales el error de estimación es ortogonal a la señal estimada  $y(n)$ . A la ecuación anterior se le conoce con el nombre de principio de la ortogonalidad.

Se puede demostrar utilizando el principio de la ortogonalidad que los coeficientes del filtro de estimación están dados por la solución a la ecuación de Wiener-Hopf [Hay91]:

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \quad (1.17)$$

donde los coeficientes del filtro de estimación están dados por:

$$\mathbf{w}_o = [\mathbf{w}_0, \dots, \mathbf{w}_{M-1}]^T, \quad (1.18)$$

$\mathbf{p}$  está dado por:  $E[\mathbf{u}(n)d^*(n)]$  y  $\mathbf{R}$  está dado por  $E[\mathbf{u}(n)\mathbf{u}^H(n)]$ ; donde  $\mathbf{u}(n)$  es el vector de muestras de entrada al filtro de estimación:

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T \quad (1.19)$$

Cabe señalar que esta solución al problema de la igualación requiere de la inversión de la matriz  $\mathbf{R}$ , sin embargo como se verá en la parte II de este trabajo existen algoritmos recursivos que evitan el cálculo directo de la inversa de  $\mathbf{R}$ .

## 1.11 Conclusiones

En este capítulo se dio un panorama del problema de la igualación de canal. Dentro de este panorama se mencionaron brevemente las distintas soluciones que se han propuesto para resolver el problema de la igualación, de la necesidad de igualar el canal, de la caracterización del mismo como un "tapped delay line filter", de la

necesidad de un igualador adaptivo, de las diferentes estructuras de los igualadores lineales y, finalmente se dio una solución "global" al problema de la igualación de canal. Pero jamás se entro en los detalles de la algoritmica de la igualación de canal; pues a pesar de todo es el meollo central de este trabajo y de la igualación de canal; pues sin un procedimiento para determinar los coeficientes del igualador de acuerdo a algún criterio de minimización, sólo se dara "palos de ciego" logrando una igualación subóptima en la cual no se podrán lograr altas tasas de transmisión.

En la siguiente parte de este trabajo se abordarán en detalle los algoritmos utilizados para determinar los coeficientes del filtro igualador. Como se ha hecho mención dada la gran cantidad de algoritmos solo se abordarán los que de una u otra manera han logrado una aceptación universal sin por ello no dejar de hacer una breve mención de algunos de los algoritmos que no han logrado dicha popularidad. En el capítulo dos se abordarán los coeficientes basados en el gradiente estocástico, mientras que en el capítulo tres se mencionarán los algoritmos de los mínimos cuadrados los que hoy por hoy son los más populares y; finalmente en el capítulo cuatro se tratarán los algoritmos que utilizan retroalimentación.

## Parte II

# Algoritmos Recursivos Para La Igualación de Canal

## Capítulo 2

# Algoritmos De Gradiente Estocástico

En este capítulo se dan las ideas fundamentales de los igualadores lineales de canal de respuesta al impulso finita. Primeramente se desarrolla el método del "steepest descent" para el ajuste recursivo de los taps del filtro de Wiener. A continuación se utiliza el método del "steepest descent" para derivar el algoritmo de los mínimos cuadrados (MC).

Cabe señalar que estos dos métodos basados en el gradiente estocástico no son los únicos para resolver el problema de la igualación, sin embargo sí son convenientes dado que evitan la inversión de la matriz de correlación que se presenta al solucionar el problema de la igualación mediante un filtro de Wiener.

La existencia de métodos no recursivos o "globales" para resolver el problema de la igualación de canal es abundante en la literatura y dado que el tratamiento de cada uno de estos métodos en detalle sale del objetivo de este trabajo se prefiere hacer sólo una breve mención de los mismos. En estos algoritmos generalmente se fija un modelo ya sea autoregresivo, de promedio móvil ó una combinación de ambos ARMA para el filtro igualador. Una vez hecho esto, generalmente se requiere de la inversión matricial para obtener los parámetros del modelo fijado. Como ejemplos de estos algoritmos tenemos: el algoritmo de Levinson Durbin [PRLN92], el filtro de Wiener [Hay91] y, los algoritmos cicloestacionarios propuestos por Zhi Ding [LD94].

### 2.1 Algoritmo del Steepest Descent

Este algoritmo es básico para el entendimiento de las diversas maneras en que son implementados los algoritmos de adaptación basada en el gradiente. El método del steepest descent es recursivo en el sentido de que empieza de algún valor inicial arbitrario para el vector de coeficientes del igualador, el cual se refina al incrementar el número de iteraciones. El valor final del vector de coeficientes converge a la solución de Wiener. El punto importante es que el método del steepest descent encuentra el mínimo punto de la superficie de error sin ningún conocimiento de la misma [Hay91].

Considere un filtro transversal cuya entrada es:  $u(n), u(n-1), \dots, u(n-M+1)$  y cuyos coeficientes son:  $w_0(n), w_1(n), \dots, w_{M-1}(n)$ . Las entradas son muestras generadas por un proceso estocástico estacionario en sentido amplio (WSS) con matriz de correlación  $\mathbf{R}$  y media cero. Además de estos parámetros, se le debe proporcionar al igulador una "respuesta deseada"  $d(n)$  (en nuestro caso una secuencia de entrenamiento). En la figura 3.1 se muestra un diagrama para la acción del filtro igulador hasta aquí descrito.

El vector de entrada al tiempo  $n$  se denota mediante  $\mathbf{u}(n)$ , y el estimado de la respuesta deseada se denota mediante  $\hat{d}(n|\mathcal{U}_n)$ , donde  $\mathcal{U}_n$  es el subespacio generado para las entradas al filtro  $u(n), u(n-1), \dots, u(n-M+1)$ .

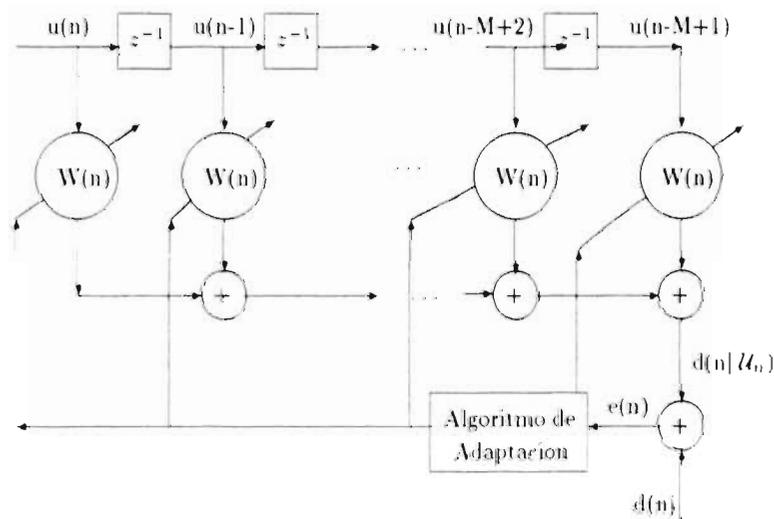


Figura 2.1: Estructura de un filtro transversal adaptivo.

El error de estimación al tiempo  $n$  se define como la diferencia entre el valor de la respuesta deseada al tiempo  $n$  y su valor estimado también al tiempo  $n$ ; matemáticamente tenemos:

$$\begin{aligned} \epsilon(n) &= d(n) - \hat{d}(n|\mathcal{U}_n) \\ &= d(n) - \mathbf{w}^H(n)\mathbf{u}(n) \end{aligned} \quad (2.1)$$

donde  $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T$  es el vector de coeficientes del filtro igulador ;  $\mathbf{w}^H(n)\mathbf{u}(n)$  es el producto interno entre vector de coeficientes del filtro

regulador y el vector de entrada. En forma expandida el vector de entrada se escribe como:  $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$ .

En caso de que el vector de entrada  $\mathbf{u}(n)$  y la respuesta deseada sean conjuntamente estacionarias, entonces el error cuadrático medio  $J(n)$  al tiempo  $n$  es una función cuadrática del vector de coeficientes, por lo que la ecuación 1.5 es válida [Hay91].

La superficie descrita por la ec. (1.5) tiene la forma de un paraboloide con un mínimo único. Esta superficie se le conoce con el nombre de superficie del error (error performance surface) del filtro adaptivo [WS85]. El proceso de adaptación consiste en buscar el punto mínimo de esta superficie; en éste punto el valor del vector de coeficientes adquiere su valor óptimo  $\mathbf{w}_o$ , el cual está definido por las ecuaciones de Wiener-Hopf:

$$\mathbf{R}\mathbf{w}_o = \mathbf{p} \quad (2.2)$$

Al evaluar el error cuadrático medio en este punto, se tiene:

$$J_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \quad (2.3)$$

### 2.1.1 El Algoritmo "Steepest Descent".

Una manera de resolver directamente la ec. de Wiener-Hopf es al invertir la matriz  $\mathbf{R}$ , otra forma es aprovechar la estructura Toeplitz de la matriz Hermiteana  $\mathbf{R}$  al aplicar el método de Levinson [Hay91]. El método del "steepest descent", encuentra el valor mínimo de la función de costo,  $J_{min}$ , mediante el siguiente procedimiento:

1. Se escoge arbitrariamente un valor inicial,  $\mathbf{W}(0)$ , para el vector de coeficientes  $\mathbf{W}(k)$ . El valor  $\mathbf{W}(0)$  proveerá una suposición inicial acerca de la localización del punto mínimo de la superficie representada por  $J(n)$ .
2. Usando el valor inicial o presente del vector de coeficientes  $\mathbf{W}$  se calcula el gradiente de la función de costo  $J(n)$ .
3. Se calcula la siguiente estimación para el valor mínimo de del vector  $\mathbf{W}$ . Este vector debe tener un sentido opuesto al gradiente calculado en el punto anterior.
4. Se regresa al paso 2 y se repite el proceso.

Matemáticamente las ideas anteriores se pueden expresar de la siguiente manera: sea  $\nabla(J(n))$  el gradiente de la función de costo al tiempo  $n$ ; con lo cual el valor actualizado al tiempo  $n+1$  del vector de coeficientes del filtro adaptivo se puede expresar como:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla(J(n))] \quad (2.4)$$

donde  $\mu$  es un parámetro conocido como el tamaño de paso de adaptación. El gradiente,  $\nabla$  de la función de costo está dado por [Hay91]:

$$\nabla(J(n)) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \quad (2.5)$$

Sustituyendo la ecuación 3.5 en la 3.4, obtenemos la siguiente recursión para el vector de coeficientes  $\mathbf{w}(n)$ :

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu [\mathbf{p} - \mathbf{R}\mathbf{w}(n)], \quad n = 0, 1, 2, \dots \quad (2.6)$$

Obsérvese que el tamaño del paso de adaptación,  $\mu$ , controla la corrección que se le aplica al vector de coeficientes del filtro al tiempo  $n$ , para obtener el valor de este vector al tiempo  $n+1$ . De acuerdo a esta última ecuación la corrección aplicada a  $\mathbf{w}(n)$  es igual a  $\mu [\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$ .

La condición necesaria y suficiente para que el algoritmo del "Steepest descent" converga es [Hay91]:

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (2.7)$$

donde  $\lambda_{max}$  es el eigen valor máximo de la matriz de autocorrelación,  $\mathbf{R}$ . Dado que  $\lambda_{max} < M \cdot \mathbf{R}(0)$ , un límite más restrictivo para  $\mu$  es [PRLN92]:

$$0 < \mu < \frac{2}{M \cdot \mathbf{R}(0)} \quad (2.8)$$

dónde  $\mathbf{R}(0) = E \{|\mathbf{u}(k)|^2\}$ . Como se señala en [Hay91] y en [WSS5], la convergencia del algoritmo "Steepest descent" es altamente sensible a las variaciones de  $\mu$  y el esparcimiento (variancia) de los eigen valores de la matriz  $\mathbf{R}$ .

### 2.1.2 Organigrama del Algoritmo Steepest Descent

Condiciones Iniciales	
$\mathbf{w}(0) = 0$	
Ecuaciones	
$\hat{d}(n) = \mathbf{w}^H(n)\mathbf{u}(n)$	(3.1)
$\epsilon(n) = d(n) - \hat{d}(n)$	(3.1)
$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu [\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$	(3.6)

Tabla 2.1: El método del "Steepest Descent"

## 2.2 Algoritmo LMS ("Least Mean Squares").

El algoritmo LMS fue desarrollado por Widrow y Hoff [WS85]. Este algoritmo como el método del Steep descent esta basado en el gradiente de la función de costo, sin embargo el algoritmo LMS ha ganado gran aceptación debido a su simplicidad, dado que no requiere del conocimiento de los estadísticos de la señal de entrada al filtro igualador y, además es recursivo de tal manera que evita la inversión de la matriz de autocorrelación de la señal de entrada,  $\mathbf{R}$ , para resolver la ecuación de Wiener-Hopff. Cabe mencionar que el algoritmo LMS sea convertido en un patrón contra el cual se comparan los algoritmos adaptivos más recientes.

### 2.2.1 Estructura del Algoritmo "Least Mean Squares"

Operacionalmente el algoritmo LMS puede ser descrito de la siguiente manera [WS85]:

1. Consta de un proceso adaptivo, el cuál involucra el ajuste automático del vector de coeficientes del filtro del igualador.
2. Un proceso de filtrado, en el cual se estima la respuesta deseada mediante el producto interno entre el vector de entrada y el vector de coeficientes del filtro; finalmente se obtiene el error de estimación al comparar la respuesta deseada con su valor estimado. Este error de estimación es a su vez utilizado para actualizar el proceso adaptable, de tal manera que exita un lazo cerrado de retroalimentación.

El algoritmo LMS no requiere de un conocimiento de los estadísticos de la señal de entrada al filtro del igualador, ya que los promedios probabilísticos requeridos por el método del Steepest descent son sustituidos por promedios temporales (asumiendo un proceso estocástico ergódico). Al efectuar dicho cambio se ha observado que tanto el gradiente como el vector de coeficientes adquieren un comportamiento ruidoso, lo que trae como consecuencia que el algoritmo LMS no nos de el valor óptimo (solución de las ecuaciones de Wiener-Hopff) para el vector de coeficientes sino un valor muy cercano al valor óptimo. Cabe hacer notar que al aumentar el numero de iteraciones el valor del vector de coeficientes estará variando de una manera ruidosa alrededor del punto mínimo de la función de costo.

### 2.2.2 Algoritmo LMS

Dado que el algoritmo LMS sustituye los promedios probabilísticos con promedios temporales (asumiendo ergodicidad), entonces en la expresión del gradiente del algoritmo steepest descent es necesario sustituir la matriz de correlación,  $\mathbf{R}$ , y el vector de la correlación cruzada entre los datos de entrada y la respuesta deseada,  $\mathbf{p}$ , por sus respectivas estimaciones:

$$\hat{\mathbf{R}} = \mathbf{u}(n)\mathbf{u}^H(n) \quad (2.9)$$

y

$$\hat{\mathbf{p}}(n) = \mathbf{u}(n)d^*(n) \quad (2.10)$$

Utilizando estas estimaciones el gradiente está dado por:

$$\hat{\nabla}(J(n)) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n) \quad (2.11)$$

Sustituyendo 3.11 en la ecuación 3.4 del algoritmo steepest descent obtenemos la siguiente expresión para el estimado del vector de coeficientes:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n) [d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \quad (2.12)$$

### 2.2.3 Organigrama del Algoritmo LMS.

Condiciones Iniciales	
$\hat{\mathbf{w}}(0) = 0$	
Ecuaciones	
$\hat{d}(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$	(3.13)
$e(n) = d(n) - \hat{d}(n)$	(3.14)
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n) \cdot e^*(n)$	(3.15)

Tabla 2.2: Algoritmo LMS

### 2.2.4 Propiedades del Algoritmo LMS.

Cuando los datos de entrada,  $\mathbf{u}(k)$ , son estacionarios, el algoritmo LMS tiene las siguientes propiedades [Hay91] [WSS5]:

1. El algoritmo es convergente en la media, esto es,  $E\{\hat{\mathbf{w}}(n)\} \rightarrow \mathbf{w}_o$  cuando  $n \rightarrow \infty$ , si el tamaño del paso  $\mu$  satisface  $0 < \mu < 2/\lambda_{max}$ .
2. El algoritmo LMS es convergente en el valor cuadrático medio:

$$E\{J(n)\} = J_{min} + E\{J_{er}(\infty)\},$$

$$\text{si } 0 < \mu < \frac{2}{\sum_{i=1}^M \lambda_i}$$

dónde  $J(n)$  está dado por (1.5),  $\{\lambda_i\}$  son los eigen valores de  $\mathbf{R}$ , y  $E\{J_{ex}(\infty)\}$  es el exceso del error cuadrático medio:

$$E\{J_{ex}(\infty)\} = \frac{\mu J_{min} \sum_{i=1}^M \lambda_i}{2 - \mu \sum_{i=1}^M \lambda_i} \quad (2.16)$$

3. Si el algoritmo es convergente en el cuadrado de la media, también lo es en la media. Esto es consecuencia de:

$$\lambda_{max} \leq \sum_{i=1}^M \lambda_i \quad (2.17)$$

1. El tamaño del paso  $\mu$  es la memoria del algoritmo LMS. Si  $\mu$  es pequeño, el algoritmo LMS se adapta lentamente y  $E\{J_{ex}(\infty)\}$  será pequeño. Por otro lado, si  $\mu$  es grande, el algoritmo se adapta rápidamente a las expensas de un gran valor de  $E\{J_{ex}(\infty)\}$ .
5. La tasa de convergencia del algoritmo es lenta si, la matriz de autocorrelación,  $\mathbf{R}$ , de los datos de entrada exhibe una gran dispersión de sus eigen valores. La dispersión de la eigen valores se define como:  $(\lambda_{max}/\lambda_{min})$ .
6. El algoritmo LMS exhibe un desajuste en el estado estable. Esto es,  $\mathcal{M} = E\{J_{ex}(\infty)\} / J_{min}$ , el cual se incrementa linealmente con  $\mu$ .

## 2.3 Conclusiones

En este capítulo se trataron los algoritmo "steepest descent" y el algoritmo LMS. En estos algoritmos no se requiere de la inversión directa de la matriz de autocorrelación de los datos de entrada sino que esta se calcula de una forma indirecta en un número menor de operaciones.

Como se vio el algoritmo "steepest descent" requiere del conocimiento de los estadísticos de la señal de entrada al receptor y de la señal deseada. Esto representa cierto inconveniente en la práctica pues no siempre se dispone de este conocimiento, sino por el contrario nop de dispone de él. De aquí que para solucionar este problema Widrow propuso el algoritmo "Least Mean Squares (LMS)", el cual no requiere del conocimiento de los estadísticos de la señal de entrada al igualador. Esta característica del algoritmo LMS aunada a su robustez numérica lo ha vuelto el "patrón" contra el cual se compara cualquier algoritmo que se propone en la literatura.

A pesar de las grandes propiedades del algoritmo LMS, sigue siendo un algoritmo de gradiente, el cual no converge a la solución óptima con cierta lentitud en comparación con los algoritmos de los mínimos cuadrados. A pesar de que esto a parentemente no representa mayor problema recordemos que la transmisión de datos cada vez se realiza a una mayor tasa, lo cual implica la constante búsqueda de

nuevos algoritmos que convergan más rápido y realicen un menor número de operaciones por muestra de salida con el fin de satisfacer de una manera más adecuada estos requerimientos. De aquí que se hallan desarrollados los algoritmos rápidos de mínimos cuadrados, los cuales convergen más rápido y también realizan un número menor de operaciones. En el siguiente capítulo se abordarán estos algoritmos tanto para una topología transversal del filtro igualador como una topología en lattice. Estos algoritmos se utilizan en situaciones que requieren tasas más altas de transmisión por lo cual resultan de extrema importancia para el tema aquí tratado.

## Capítulo 3

# Algoritmo de los Mínimos Cuadrados (MC)

En este capítulo se tratará el algoritmo de los mínimos cuadrados ("least squares (LS)") el cual para resolver el problema de la regulación de canal (filtrado) no requiere del conocimiento de los estadísticos de la señal de entrada al regulador.

Dado que el algoritmo de los mínimos cuadrados requiere de la inversión de la matriz de la autocorrelación de los datos de entrada al filtro y, dado que la inversión de matrices es un proceso computacionalmente costoso se prefiere recurrir a algoritmos recursivos en los que no se invierte dicha matriz directamente sino de una manera recursiva. Así el algoritmo MC nos conduce de una manera natural hacia su solución recursiva, dando por resultado el algoritmo recursivo de los mínimos cuadrados (MCR).

A partir de este último algoritmo y debido a la necesidad de realizar un procesamiento en tiempo real, se han desarrollado los algoritmos rápidos de los cuales en este capítulo se analizará el "Fast Transversal Filter (FTF)".

Finalmente, se analizará el algoritmo "recursive least squares "lattice" algorithm (LSL)" que en lugar de utilizar una estructura transversal para el filtro adaptivo utiliza una estructura en "lattice".

### 3.1 Desarrollo del Algoritmo de los Mínimos Cuadrados (MC)

La idea básica del algoritmo de los mínimos cuadrados es la siguiente: supongáse que se dispone del siguiente conjunto de datos obtenidos de la observación de un experimento o fenómeno determinado:  $u(1), u(2), \dots, u(n)$  donde estas observaciones fueron realizadas en los tiempos:  $t_1, t_2, \dots, t_n$  respectivamente. Dadas estos datos el algoritmo de los mínimos cuadrados obtiene la curva que mejor los aproxime en el sentido de minimizar la suma de los cuadrados de la diferencia entre la señal deseada y su estimación respectiva a cada instante de tiempo  $n$  [PRLN92].

### 3.1.1 La Función de Costo en el Algoritmo de los Mínimos Cuadrados

Spongase que se desea estudiar un fenómeno determinado el cual no deseamos modelar de una forma directa, es decir, no se desea establecer un modelo del sistema a través del estudio de su estructura interna; más bien lo que se pretende es establecer un modelo a través de la observación de la relación que existe entre las señales de entrada y las de salida. Para este fin, definamos la variable  $d(i)$  como la respuesta de este fenómeno o sistema cuando se aplica el siguiente conjunto de entradas:  $u(1), u(2), \dots, u(i - M + 1)$ . Por simplicidad se asume que la relación que guarda la salida con la(s) entrada(s) es lineal, por lo que podemos expresar a  $d(i)$  como un modelo de promedio móvil o de respuesta impulsional finita (ó FIR de sus siglas en Inglés):

$$d(i) = \sum_{k=0}^{M-1} w_{o,k}^* u(i - k) + \epsilon_o(i) \quad (3.1)$$

dónde  $w_{o,k}^*$  son los parámetros del sistema y,  $\epsilon_o$  representa el error de modelado. Este error es una variable aleatoria no observable que se introduce en el modelo con fines de cuantificar la exactitud con que se ajusta a la "realidad". Generalmente algunos autores prefieren evitar las complicaciones que pudieran surgir de un análisis más profundo de la exactitud de este tipo de modelado y asumen que el error de modelado es un proceso "blanco" con media cero y variancia  $\sigma^2$ , es decir:

$$E[\epsilon_o(i)] = 0 \quad (3.2)$$

y

$$E[\epsilon_o(i)\epsilon_o(k)] = \begin{cases} \sigma^2, & i = k \\ 0, & i \neq k \end{cases} \quad (3.3)$$

Esta suposición implica que al aplicar el operador valor esperado a la ec. 3.18 obtengamos:

$$E[d(i)] = \sum_{k=0}^{M-1} w_{o,k}^* u(i - k) \quad (3.4)$$

Nuestro problema es el encontrar los coeficientes del modelo que mejor se ajusten a los datos experimentalmente observados: sin embargo esta labor es imposible si no definimos un criterio basado en las mediciones del error entre los valores observados y los estimados: una vez definido este criterio se le debe aplicar una técnica de minimización para encontrar los parámetros de nuestro modelo,  $w_o$ . Por tanto, se define el error de estimación como la diferencia entre la respuesta deseada y la salida del filtro:

$$\epsilon(i) = d(i) - y(i) \quad (3.5)$$

dónde  $y(i)$  esta dada por:

$$y(i) = \sum_{k=0}^{M-1} w_k^* u(i - k) \quad (3.6)$$

por lo que el error de estimación,  $e(i)$ , se puede expresar como:

$$\epsilon(i) = d(i) - \sum_{k=0}^{M-1} w_k^* u(i-k) \quad (3.7)$$

De esta manera, el algoritmo de los mínimos cuadrados minimiza la siguiente función de costo para derivar los coeficientes del filtro:

$$\mathcal{E}(w_0, w_1, \dots, w_{M-1}) = \sum_{i=i_1}^{i_2} |\epsilon(i)|^2 \quad (3.8)$$

dónde  $i_1$  e  $i_2$  son los índices que definen los límites sobre los que la minimización es llevada a cabo. El valor que se le asigne a estos límites depende del tipo de ventana empleada para los datos de entrada [Alc86, Hay91].

## 3.1.2 Derivación de las Ecuaciones Normales

### 3.1.2.1 Principio de la Ortogonalidad

La función de costo, ec. (2.25) se puede expresar como:

$$\mathcal{E}(w_0, \dots, w_{M-1}) = \sum_{i=M}^N \epsilon(i)\epsilon^*(i) \quad (3.9)$$

donde el error de estimación está dado por la ec. (2.24). Si en ésta última ecuación se sustituyen los coeficientes del filtro,  $w_k$ , por:

$$w_k = a_k + jb_k, \quad k = 0, 1, \dots, M-1 \quad (3.10)$$

obtenemos la siguiente ecuación:

$$\epsilon(i) = d(i) - \sum_{k=0}^{M-1} (a_k - jb_k)u(i-k) \quad (3.11)$$

La  $k$ -ésima componente del gradiente de la función de costo,  $\mathcal{E}$ , para el algoritmo de los mínimos cuadrados se define como:

$$\nabla_k(\mathcal{E}) = \frac{\partial \mathcal{E}}{\partial a_k} + j \frac{\partial \mathcal{E}}{\partial b_k} \quad (3.12)$$

Efectuando las derivadas correspondientes se demuestra se obtiene [Alc86, Hay91]:

$$\nabla_k(\mathcal{E}) = -2 \sum_{i=M}^N u(i-k)\epsilon^*(i) \quad (3.13)$$

La minimización de la función de costo,  $\mathcal{E}$ , con respecto a los coeficientes del filtro  $w_0, w_1, \dots, w_{M-1}$  requiere que:

$$\nabla_k(\mathcal{E}) = 0, \quad k = 0, 1, 2, \dots, M-1 \quad (3.14)$$

Cuando este sistema de ecuaciones lineales es satisfecho, el error de estimación es mínimo,  $\epsilon_{min}$ . Por tanto cuando la función de costo es minimizada la ec. (2.39) se puede expresar como:

$$\sum_{i=M}^N u(i-k)\epsilon_{min}^*(i) = 0, \quad k = 0, 1, \dots, M-1 \quad (3.15)$$

Esta última ecuación es el principio de la ortogonalidad en su versión temporal. Este principio establece lo siguiente:

1. El error de estimación,  $\epsilon_{min}(i)$ , resultante del proceso de optimización de la función de costo,  $\mathcal{E}$ , es ortogonal al vector de entrada  $\mathbf{u}(i)$ .

### 3.1.2.2 Ecuaciones Normales

El error de estimación mínimo,  $\epsilon_{min}$ , está dado por:

$$\epsilon_{min}(i) = d(i) - \sum_{t=0}^{M-1} u(i-t) \quad (3.16)$$

dónde el vector de coeficientes del filtro,  $\hat{w}_t^*$ , es la solución al problema de los mínimos cuadrados. Sustituyendo la ec.(2.32) en ésta última ecuación, tenemos:

$$\sum_{t=0}^{M-1} \hat{w}_t \sum_{i=M}^N u(i-k)u^*(i-t) = \sum_{i=M}^N u(i-k)d^*(i), \quad k = 0, \dots, M-1 \quad (3.17)$$

Definiendo:

$$\phi(t, k) = \sum_{i=M}^N u(i-k)u^*(i-t), \quad 0 \leq t, k \leq M-1 \quad (3.18)$$

y

$$\theta(-k) = \sum_{i=M}^N u(i-k)d^*(i), \quad 0 \leq k \leq M-1 \quad (3.19)$$

podemos expresar 2.34 como [Alc86, Hay91]:

$$\sum_{t=0}^{M-1} \hat{w}_t \phi(t, k) = \theta(-k), \quad k = 0, 1, \dots, M-1 \quad (3.20)$$

Este último sistema de ecuaciones representa el sistema expandido de las ecuaciones normales para un filtro lineal de mínimos cuadrados [Alc86, Hay91]. En forma matricial tenemos:

$$\Phi \hat{\mathbf{w}} = \theta \quad (3.21)$$

donde:

$$\Phi = \begin{bmatrix} \phi(0, 0) & \phi(1, 0) & \cdots & \phi(M-1, 0) \\ \phi(0, 1) & \phi(1, 1) & \cdots & \phi(M-1, 1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(0, M-1) & \phi(1, M-1) & \cdots & \phi(M-1, M-1) \end{bmatrix} \quad (3.22)$$

es la matriz de autocorrelación de las muestras de entrada,  $u(i)$ ,  $u(i-1)$ , ...,  $u(i-M-1)$  al filtro; además:

$$\theta = [\theta(0), \theta(-1), \dots, \theta(-M+1)]^T \quad (3.23)$$

es el vector de la correlación cruzada de las muestras de entrada,  $u(i)$ ,  $u(i-1)$ , ...,  $u(i-M-1)$ , al filtro y la respuesta deseada  $d(i)$ . Finalmente:

$$\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1, \dots, \hat{w}_{M-1}]^T \quad (3.24)$$

La ecuación 2.38 es la forma matricial de las ecuaciones normales para los filtros lineales de mínimos cuadrados.

### 3.1.3 Propiedades del Algoritmo de Mínimos Cuadrados

A continuación se presenta un resumen de las propiedades del algoritmo de los mínimos cuadrados, para las demostraciones consulte la referencia: [Alo86, Hay91]:

1. La estimación del vector de coeficientes del filtro,  $\hat{\mathbf{w}}$  es no sesgado, si el proceso estocástico del error de modelado  $\{e_o(i)\}$  tiene media cero.
2. Cuando el error de modelado  $\{e_o(i)\}$  es "blanco" con media cero y variancia  $\sigma^2$ , la matriz de covariancia del estimado  $\hat{\mathbf{w}}$  es igual a  $\sigma^2 \Phi^{-1}$ .
3. Cuando el error de modelado  $\{e_o(i)\}$  es "blanco" con media cero, el estimado  $\hat{\mathbf{w}}$  es el mejor estimador lineal no sesgado.
4. Cuando el error de modelado  $\{e_o(i)\}$  es blanco, gaussiano y con media cero, el estimado  $\hat{\mathbf{w}}$  es igual al límite inferior "Cramér-Rao" para estimadores no sesgados.

## 3.2 Algoritmo Adaptable de los Mínimos Cuadrados

El sistema expandido de las ecuaciones normales para un filtro lineal de mínimos cuadrados, ec. 3.38, puede ser resuelto mediante la inversión de la matriz de autocorrelación de las muestras de entrada al filtro,  $\Phi$ , siempre y cuando la inversa

exista. Sin embargo, la inversión de matrices es computacionalmente cara, por lo que se prefiere resolver el sistema de ecuaciones representado por la ec. 2.38 de una manera recursiva.

El algoritmo que en esta sección se presenta es un algoritmo recursivo en el sentido de que el cálculo de la estimación del vector de coeficientes del filtro al tiempo  $n$ , se calcula en base a la estimación de éste mismo vector al tiempo  $n - 1$  y a las nuevas muestras que van entrando al filtro; por tanto, nos referimos a éste algoritmo como el algoritmo de los mínimos cuadrados recursivos.

### 3.2.1 Desarrollo del Algoritmo MCR

En este algoritmo así como en los algoritmos rápidos basados en la estimación de mínimos cuadrados se incluye un factor de ponderación ó "factor de olvido" en la función de costo dada por la ec. 3.8 con el objeto de que el algoritmo pueda hacer un seguimiento de los estadísticos de la señales en un ambiente no estacionario, de esta manera tenemos:

$$\mathcal{E}(n) = \sum_{i=1}^n \beta(n, i) |e(i)|^2 \quad (3.25)$$

dónde  $e(i)$  es el error de estimación definido por la ecuación 3.5, la cual se puede expresar vectorialmente como:

$$e(i) = d(i) - \mathbf{w}^H(n)\mathbf{u}(i) \quad (3.26)$$

donde  $\mathbf{u}(i)$  es el vector de muestras de entrada al filtro:

$$\mathbf{u}(i) = [u(i), u(i - 1), \dots, u(i - M + 1)]^T \quad (3.27)$$

y  $\mathbf{w}(n)$  es el vector de coeficientes del filtro al tiempo  $n$ :

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad (3.28)$$

El factor de olvido,  $\beta$ , introducido en la ecuación 3.25 tiene la siguiente propiedad [Alc86, Hay91]:

$$0 < \beta(n, i) \leq 1, \quad i = 1, 2, \dots, n \quad (3.29)$$

Este factor tiene el objetivo de ponderar el error de estimación, de tal manera que al tiempo presente se tenga un factor de ponderación cercano a uno, mientras que para tiempos anteriores este factor decrezca de tal manera que el "pasado lejano" afecte de una manera menos significativa a las estimaciones en el tiempo presente. Esto da como resultado que los algoritmos de mínimos cuadrados puedan seguir las variaciones en los estadísticos de la señal de entrada al filtro, cuando ésta no es estacionaria [Alc86, Hay91]. De aquí que, una forma comúnmente aceptada para el factor de olvido sea la siguiente:

$$\beta(n, i) = \lambda^{n-i}, \quad i = 1, 2, \dots, n \quad (3.30)$$

donde  $\lambda$  es una constante cercana pero menor que uno. Cuando  $\lambda$  es igual a uno tenemos el algoritmo ordinario de los mínimos cuadrados. El inverso de  $1-\lambda$  es una medida burda de la memoria del algoritmo [Alc86, Hay91], por lo que cuando  $\lambda$  es igual a uno, la memoria del filtro es infinita. Si se suastituye la ec. 3.30 en la 3.25 tenemos el algoritmo de los mínimos cuadrados con una ponderación exponencial, cuya función de costo es:

$$\mathcal{E}(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (3.31)$$

Recuérdese que el algoritmo de los mínimos cuadrados recursivos (MCR) resuelve el sistema expandido de ecuaciones normales para un filtro lineal de estimación por mínimos cuadrados, por lo que el valor óptimo para el vector de coeficientes del filtro está dado por la solución a la ecuación 3.21. Por tanto, para poder desarrollar el algoritmo MCR debemos deducir una ecuación recursiva para el cálculo del vector de coeficientes, para tal efecto defínase la matriz de autocorrelación de las muestras de entrada al filtro como:

$$\Phi(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}(i)\mathbf{u}^H(i) \quad (3.32)$$

y defínase el vector de correlación cruzada de las muestras de entrada al filtro y la respuesta deseada como:

$$\theta(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}(i)d^*(i) \quad (3.33)$$

dónde el asterisco denota el complejo conjugado. La recursión para la matriz de autocorrelación de las muestras de entrada al filtro se haya al aislar el término para el cual  $i = n$ , en la ecuación (3.32):

$$\Phi = \lambda \left[ \sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{u}(i)\mathbf{u}^H(i) \right] + \mathbf{u}(n)\mathbf{u}^H(n) \quad (3.34)$$

En esta última ecuación, la expresión entre paréntesis cuadrados es igual a la matriz de autocorrelación de las muestras de entrada al filtro al tiempo  $n-1$ , por lo tanto:

$$\Phi(n) = \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n) \quad (3.35)$$

dónde  $\Phi(n-1)$  es el valor de la matriz de autocorrelación de las muestras de entrada al tiempo  $n-1$ , y el producto matricial  $\mathbf{u}(n)\mathbf{u}^H(n)$  es la corrección que se le aplica a  $\mathbf{u}(n)\mathbf{u}^H(n)$  para obtener  $\Phi(n)$ .

Similarmente se desarrolla una recursión para el vector de la correlación cruzada de las muestras de entrada al filtro y la respuesta deseada. Partiendo de la ec. 3.33 se puede demostrar que [Alc86, Hay91]:

$$\theta(n) = \lambda\theta(n-1) + \mathbf{u}(n)d^*(n) \quad (3.36)$$

La siguiente recursión que se desarrolla es la recursión para la matriz inversa de  $\Phi$ . El desarrollo de esta recursión consiste simplemente en aplicar el lema de inversión matricial a la ecuación 3.35, con lo cual obtenemos (ver Apéndice A):

$$\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\Phi^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{u}^H(n)\Phi^{-1}(n-1)\mathbf{u}(n)} \quad (3.37)$$

Por conveniencia se define:

$$\mathbf{P}(n) = \Phi^{-1}(n) \quad (3.38)$$

y

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)} \quad (3.39)$$

Al vector  $\mathbf{k}(n)$  de dimensión  $M \times 1$  algunos autores lo llaman "vector de ganancia" pero por otro lado también se le conoce como ganancia de Kalman [Alc86]. Sustituyendo (3.39) y (3.38) en la ecuación (3.37) obtenemos:

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{K}(n)\mathbf{u}^H(n)\mathbf{P}(n-1) \quad (3.40)$$

A continuación se desarrollará una expresión alternativa para el vector de ganancia,  $\mathbf{k}(n)$ , que nos será de utilidad en nuestro desarrollo posterior del algoritmo de mínimos cuadrados recursivos. Nótese que la ecuación 3.39 se puede expresar como:

$$\begin{aligned} \mathbf{k}(n) &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n) \\ &= \left[ \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1) \right] \mathbf{u}(n) \end{aligned} \quad (3.41)$$

De la ec. 3.40 se observa que la expresión entre paréntesis es igual a  $\mathbf{P}(n)$ , por lo que tenemos:

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{u}(n) \quad (3.42)$$

La última ecuación junto con la ec. 3.38 da por resultado:

$$\mathbf{k}(n) = \Phi^{-1}(n)\mathbf{u}(n) \quad (3.43)$$

Esta ecuación nos dice que el vector de ganancia es igual al vector de muestras de entrada transformado por la inversa de la matriz de autocorrelación de las muestras de entrada al filtro.

### 3.2.1.1 Recursión para el Vector de Coeficientes del Filtro

Recuerdese qué deseamos resolver las ecuaciones normales para un filtro lineal que utiliza la estimación por mínimos cuadrados de una manera recursiva, por lo que resulta lógico sustituir las recursiones desarrolladas en las secciones anteriores en la

expresión matricial para las ecuaciones normales. Por tanto, sustituyendo 3.36 y 3.38 en 3.21 tenemos:

$$\begin{aligned}
 \hat{\mathbf{w}}(n) &= \Phi^{-1}(n)\theta(n) \\
 &= \mathbf{P}(n)\theta(n) \\
 &= \lambda\mathbf{P}(n)\theta(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n)
 \end{aligned} \tag{3.44}$$

Sustituyendo 3.40 sólo en el lado izquierdo de 3.44, tenemos:

$$\begin{aligned}
 \hat{\mathbf{w}}(n) &= \mathbf{P}(n-1)\theta(n-1) - \mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)\theta(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^* \\
 &= \Phi^{-1}(n-1)\theta(n-1) - \mathbf{k}(n)\mathbf{u}^H(n)\Phi^{-1}(n-1)\theta(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^* \\
 &= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n)
 \end{aligned} \tag{3.45}$$

Finalmente utilizando la ecuación 3.42 en la anterior, tenemos:

$$\begin{aligned}
 \hat{\mathbf{w}} &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n) \left[ d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n-1) \right] \\
 &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)
 \end{aligned} \tag{3.46}$$

dónde  $\alpha(n)$  se le conoce con el nombre de inovación y, está dada por la expresión entre paréntesis cuadrados en la ecuación anterior:

$$\begin{aligned}
 \alpha(n) &= d(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}^*(n-1) \\
 &= d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n)
 \end{aligned} \tag{3.47}$$

El producto interno dado por  $\hat{\mathbf{w}}^H(n-1)\mathbf{u}(n)$  representa la estimación de la respuesta deseada,  $d(n)$ , tomando como base la estimación del vector de coeficientes del filtro al tiempo anterior. De acuerdo a esto, a  $\alpha(n)$  también se le llama el error de estimación a priori. No se debe confundir este error con el error de estimación a posteriori dado por:

$$\epsilon(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \tag{3.48}$$

### 3.2.2 Organigrama del Algoritmo de Mínimos Cuadrados Recursivos

Condiciones Iniciales	
$\mathbf{P}(0) = \delta^{-1}\mathbf{I}, \quad \delta = \text{cte. positiva pequeña}$ $\hat{\mathbf{w}}(0) = \mathbf{0}$	
Para $n = 1, 2, \dots$ , calcule	
$\pi(n) = \mathbf{u}^H(n)\mathbf{P}(n-1)$	(3.49)
$\kappa(n) = \lambda + \pi(n)\mathbf{u}(n)$	(3.50)
$\mathbf{k}(n) = \frac{\pi(n)}{\kappa(n)}$	(3.51)
$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n)$	(3.47)
$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$	(3.46)
$\mathbf{P}'(n-1) = \mathbf{k}(n)\pi(n)$ (3.52)	
$\mathbf{P}(n) = \frac{1}{\kappa} \left( \mathbf{P}(n-1) - \mathbf{P}'(n-1) \right)$	(3.53)

Tabla 3.1: El algoritmo de Mínimos Cuadrados Recursivos (MCR).

### 3.3 Algoritmos Rápidos para La Igualación de Canal

En la sección anterior se trató el algoritmo de los mínimos cuadrados recursivos, el cual resuelve las ecuaciones normales de una manera recursiva, es decir, calcula el estimado del vector de coeficientes del filtro al tiempo  $n$  basándose en la estimación de dicho vector al tiempo  $n-1$ . Este algoritmo tiene una complejidad computacional del orden de  $M^2$  [Ale86, Hay91], donde  $M$  es el número de coeficientes del filtro de igualación. Cuando el tiempo de procesamiento es un factor primordial en determinada aplicación, tal como generalmente ocurre en el caso de la igualación de canal, debemos recurrir a un algoritmo rápido. En la literatura se entiende por un algoritmo rápido como aquel algoritmo cuya complejidad computacional se incrementa de una forma lineal con el número de variables libres o parámetros a ajustar [Ale86, Hay91].

En la actualidad con el incremento de las aplicaciones de tipo multimedia donde se requiere la transmisión en tiempo real tanto de datos, imágenes, audio y video a través de redes de comunicaciones; es indispensable el uso de algoritmos rápidos para la igualación de canal. Por tanto, los algoritmos rápidos de filtrado adaptable han experimentado un creciente interés en el ámbito de la investigación y en del desarrollo tecnológico.

En esta sección se presentan dos temas fundamentales para el desarrollo de los algoritmos rápidos del filtrado adaptable: la predicción adaptiva hacia adelante y la predicción adaptiva hacia atrás.

#### 3.3.1 Principios Básicos

##### 3.3.1.1 Predicción Adaptiva Lineal Hacia Adelante

Considere el predictor lineal hacia adelante de orden  $M$  mostrado en la figura 3.1, donde la estimación del vector de coeficientes del filtro,  $\hat{\mathbf{w}}$ , alcanza su valor óptimo (de acuerdo al criterio de minimización de los mínimos cuadrados) durante el intervalo de observación  $1 \leq i \leq n$ . Denotemos con  $f_M(i)$  el error de predicción hacia adelante producido por el predictor al tiempo  $i$  en respuesta al vector de muestras de entrada al filtro  $\mathbf{u}_M(i-1)$ :

$$f_M(i) = u(i) - \hat{\mathbf{w}}^H(n)\mathbf{u}_M(i-1), \quad 1 \leq i \leq n \quad (3.54)$$

donde  $u(i)$  representa la respuesta deseada y,  $\mathbf{u}_M(i-1)$  en forma extendida está dada por:

$$\mathbf{u}_M^T(i-1) = [u(i-1), u(i-2), \dots, u(i-M)] \quad (3.55)$$

Usualmente en la literatura se refiere a  $f_M(i)$  como el error de predicción a posteriori, ya que su cálculo se basa en el valor de la estimación del vector de coeficientes del filtro al tiempo  $n$ .

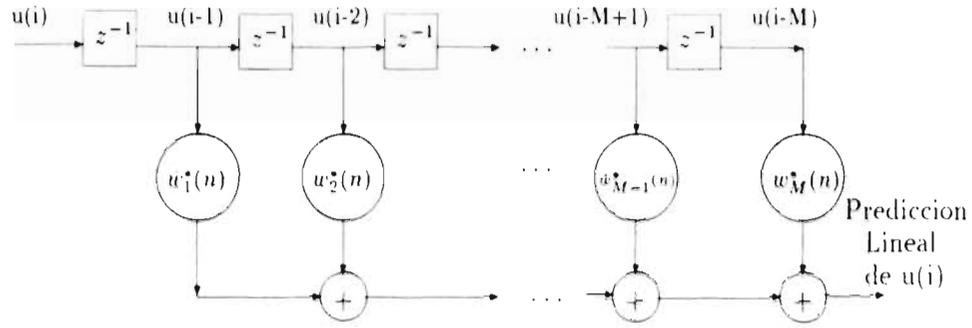


Figura 3.1: Predictor hacia adelante de orden  $M$ .

El problema de la predicción lineal adaptiva hacia adelante también se puede formular al especificar el filtro de predicción del error que se muestra en la figura 3.2. Sea  $\mathbf{a}_M(n)$  el vector de coeficientes del filtro de orden  $M$  de predicción del error al tiempo  $n$ . Nótese que éste vector guarda la siguiente relación con el vector de coeficientes del filtro  $\hat{\mathbf{w}}(n)$ :

$$\mathbf{a}_M(n) = \begin{bmatrix} 1 \\ -\hat{\mathbf{w}}(n) \end{bmatrix} \quad (3.56)$$

Se define al vector de muestras de entrada al filtro  $\mathbf{u}_{M+1}$ , al orden  $M+1$  como:

$$\mathbf{u}_{M+1}(i) = \begin{bmatrix} u(i) \\ \mathbf{u}_M(i-1) \end{bmatrix} \quad (3.57)$$

Con lo cual el error de predicción a posteriori hacia adelante,  $f_M(i)$ , también se puede expresar mediante:

$$f_M(i) = \mathbf{a}_M^H(n) \mathbf{u}_{M+1}(i), \quad 1 \leq i \leq n \quad (3.58)$$

La estimación del vector de coeficientes,  $\hat{\mathbf{w}}$ , del predictor forward es la solución obtenida del problema de minimizar la suma del cuadrado de los errores de predicción hacia adelante para el intervalo de observación:  $1 \leq i \leq n$ . De manera análoga, el vector de coeficientes,  $\mathbf{a}_M(n)$ , del filtro de predicción del error es la solución al mismo problema de minimización sólo que formulado en términos del filtro de predicción del error, es decir, sujeto a la restricción de que el primer elemento de  $\mathbf{a}_M(n)$  es igual a la unidad [Ale86, Hay91]. En cualquier caso, la solución al problema de los mínimos cuadrados se puede expresar en términos de las ecuaciones normales extendidas para la predicción lineal hacia adelante:

$$\Phi_{M+1}(n) \mathbf{a}_M(n) = \begin{bmatrix} \mathcal{F}_M(n) \\ \mathbf{0} \end{bmatrix} \quad (3.59)$$

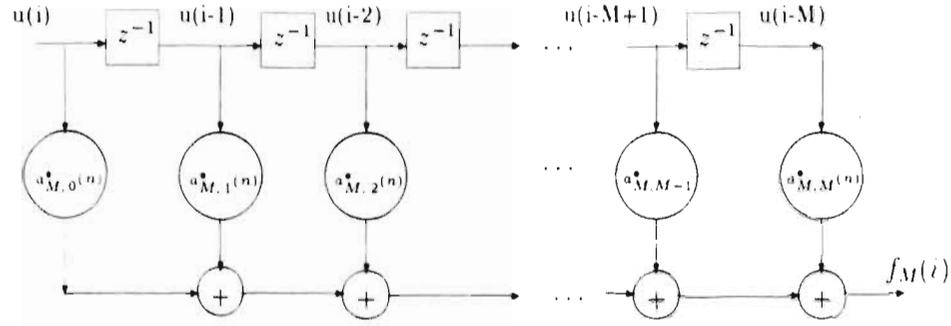


Figura 3.2: Filtro de Predicción del Error.

donde  $\mathbf{0}$  es el vector nulo de  $M \times 1$ . El escalar  $\mathcal{F}_M(n)$  es el valor mínimo de la suma ponderada del cuadrado de los errores a posteriori de predicción hacia adelante:

$$\mathcal{F}_M(n) = \sum_{i=1}^n \lambda^{n-i} |f_M(i)|^2 \quad (3.60)$$

La matriz  $\Phi_{M+1}(n)$  de  $(M+1)$  por  $(M+1)$  es la matriz de autocorrelación del vector de muestras de entrada al filtro de predicción del error  $\mathbf{u}_{M+1}(i)$ :

$$\Phi_{M+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{M+1}(i) \mathbf{u}_{M+1}^H(i) \quad (3.61)$$

Utilizando la ec. 3.57 podemos expresar a  $\Phi_{M+1}(n)$  como:

$$\Phi_{M+1}(n) = \begin{bmatrix} \mathcal{U}_1(n) & \theta_1^H(n) \\ \theta_1(n) & \Phi_M(n-1) \end{bmatrix} \quad (3.62)$$

donde  $\mathcal{U}_1(n)$  es la suma ponderada del cuadrado de la "respuesta deseada" utilizada para la predicción hacia adelante:

$$\mathcal{U}_1(n) = \sum_{i=1}^n \lambda^{n-i} |u(i)|^2 \quad (3.63)$$

$\theta_1(n)$  es el vector de  $M \times 1$  de la correlación cruzada del vector de muestras de entrada al predictor (ver fig. 3.1), y la respuesta deseada  $u(i)$ :

$$\theta_1(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i-1) u^*(i) \quad (3.64)$$

Finalmente,  $\Phi_M(n-1)$  es la matriz de autocorrelación del vector de muestras de entrada,  $\mathbf{u}_M(i-1)$ , al filtro predictor de la figura 3.1:

$$\begin{aligned} \Phi_M(n-1) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i-1) \mathbf{u}_M^H(i-1) \\ &= \sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{u}_M(i) \mathbf{u}_M^H(i) \end{aligned} \quad (3.65)$$

A continuación debemos resolver el problema de la predicción lineal hacia adelante de una manera adaptiva utilizando el algoritmo de mínimos cuadrados recursivos (MCR) tratado en la sección anterior. Para este efecto algunos autores utilizan la analogía entre los parámetros del algoritmo MCR y de la predicción lineal hacia adelante mostrada en la tabla 3.1[Alc86, Hay91]. De las ecuaciones (3.43), (3.46) y (3.47) se deduce la siguiente recursión para el vector de coeficientes del predictor:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}_M(n-1)\eta_M^*(n) \quad (3.66)$$

donde  $\eta_M(n)$  es el error a priori de predicción hacia adelante, esto es la diferencia entre la respuesta deseada y su estimación:

$$\eta_M(n) = u(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}_M(n-1) \quad (3.67)$$

y  $\mathbf{k}_M(n-1)$  es el vector de ganancia para la predicción lineal hacia adelante de orden M:

$$\mathbf{k}_M(n-1) = \Phi_M^{-1}(n-1)\mathbf{u}_M(n-1) \quad (3.68)$$

donde el índice M se utiliza para denotar el orden del predictor. Sustituyendo la ecuación 3.66 en la ec. 3.56 obtenemos la recursión para el vector de coeficientes del filtro de predicción del error:

$$\mathbf{a}_M(n) = \mathbf{a}_M(n-1) - \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} \eta_M^*(n) \quad (3.69)$$

donde:

$$\begin{aligned} \eta_M(n) &= [1, -\hat{\mathbf{w}}^H(n-1)] \begin{bmatrix} u(n) \\ \mathbf{u}_M(n-1) \end{bmatrix} \\ &= \mathbf{a}_M^H(n-1)\mathbf{u}_{M+1}(n) \end{aligned} \quad (3.70)$$

Finalmente, utilizando la siguiente ecuación correspondiente al algoritmo de mínimos cuadrados recursivos [Alc86, Hay91]:

$$\mathcal{E}_{min}(n) = \lambda \mathcal{E}_{min}(n-1) + \alpha^*(n)c(n) \quad (3.71)$$

tenemos la siguiente recursión para el mínimo valor de la suma del cuadrado de los errores ponderados de predicción hacia adelante:

$$\mathcal{F}_M(n) = \lambda \mathcal{F}_M(n-1) + \eta_M(n)f_M^*(n) \quad (3.72)$$

□.

### 3.3.1.2 Predicción Lineal Adaptiva Hacia Atras

Considere el predictor lineal hacia atrás de orden M que se muestra en la figura 3.3, cuyo vector de coeficientes,  $\mathbf{g}(n)$ , se ha optimizado de acuerdo al criterio de

Parámetro	MCR	Predictor hacia adelante	Predictor hacia atrás
vector de entrada	$\mathbf{u}(n)$	$\mathbf{u}_M(n - 1)$	$\mathbf{u}_M(n)$
respuesta deseada	$d(n)$	$u(n)$	$u(n - M)$
vector de coeficientes	$\hat{\mathbf{w}}(n)$	$\hat{\mathbf{w}}(n)$	$\mathbf{g}(n)$
error a posteriori de estimación	$e(n)$	$f_M(n)$	$b_M(n)$
error a priori de estimación	$\alpha(n)$	$\eta_M(n)$	$v_M(n)$
vector de ganancia	$\mathbf{k}(n)$	$\mathbf{k}_M(n - 1)$	$\mathbf{k}_M(n)$
Valor mínimo de la suma de errores al cuad.	$\mathcal{E}_{min}(n)$	$\mathcal{F}_M(n)$	$\mathcal{B}_M(n)$

Tabla 3.2: Analogía de parámetros entre el algoritmo de mínimos cuadrados recursivos, la predicción lineal hacia adelante y la predicción lineal hacia atrás.

minimización del error del algoritmo de MC, en el intervalo de observación:  $1 \leq i \leq n$ . Sea  $b_M(i)$  el error de predicción hacia atrás producido por el predictor de la fig. 3.3 al tiempo  $i$  en respuesta al vector de muestras de entrada  $\mathbf{u}_M(i)$ :

$$b_M(i) = u(i - M) - \mathbf{g}^H(n)\mathbf{u}_M(i) \quad (3.73)$$

donde  $u(i - M)$  es la "respuesta deseada" y:

$$\mathbf{u}_M^T(i) = [u(i), u(i - 1), \dots, u(i - M + 1)] \quad (3.74)$$

Usualmente en la literatura  $b_M(i)$  se le conoce como el error a posteriori de predicción hacia atrás, ya que su cálculo se basa en el valor del vector de coeficientes del predictor de la figura 3.3 al tiempo  $n$ , es decir en  $\mathbf{g}(n)$ .

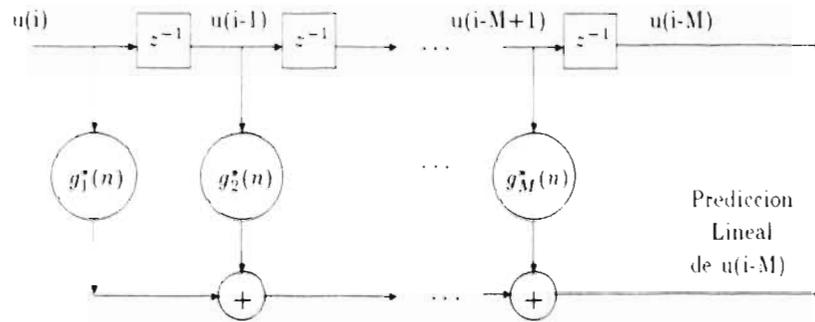


Figura 3.3: Predictor hacia atrás de orden M.

De una manera análoga al caso de la predicción lineal hacia adelante, el problema de la predicción lineal hacia atrás puede plantearse en términos del filtro de predicción del error que se muestra en la fig. 3.1. Sea  $\mathbf{c}_M(n)$  el vector de  $M \times 1$  de coeficientes del filtro de orden M de predicción del error. El vector  $\mathbf{c}_M(n)$  guarda la siguiente relación con el vector de coeficientes,  $b_M(i)$ , del predictor lineal hacia atrás:

$$\mathbf{c}_M(n) = \begin{bmatrix} -\mathbf{g}(n) \\ 1 \end{bmatrix} \quad (3.75)$$

Sea  $\mathbf{u}_{M+1}(i)$  el vector, de  $(M + 1) \times 1$ , que tiene la respuesta deseada  $u(i - M)$  como su último elemento y a los elementos del vector  $\mathbf{u}_M(i)$  como sus elementos restantes:

$$\mathbf{u}_{M+1}(i) = \begin{bmatrix} \mathbf{u}_M(i) \\ u(i - M) \end{bmatrix} \quad (3.76)$$

Entonces el error de predicción hacia atrás puede ser redefinido como:

$$b_M(i) = \mathbf{c}_M^H(n)\mathbf{u}_{M+1}(i), \quad 1 \leq i \leq n \quad (3.77)$$

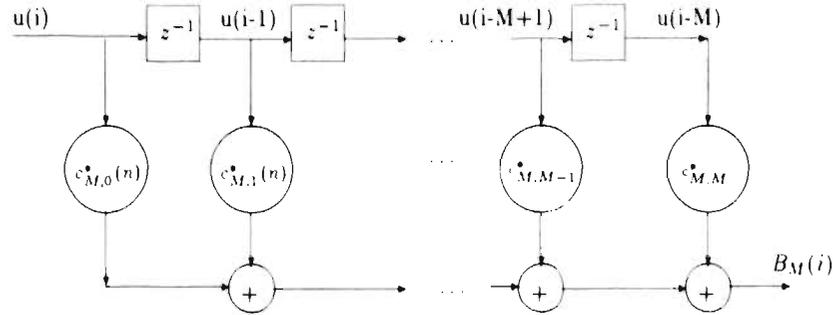


Figura 3.4: Filtro de Predicción hacia atrás del error.

El vector de coeficientes del predictor lineal hacia atrás  $\mathbf{g}(n)$  es la solución que resulta de la minimización de la suma del cuadrado de los errores a posteriori de predicción ponderados para el intervalo de observación:  $1 \leq i \leq n$ . De una manera análoga, el vector de coeficientes del filtro de predicción del error  $\mathbf{c}_M(n)$  es la solución al mismo problema de minimización sólo que sujeto a la restricción adicional de que el último elemento del vector  $\mathbf{c}_M(n)$  debe ser igual a la unidad [Alc86, Hay91]. En cualquier caso la solución al problema de los mínimos cuadrados puede ser escrita en términos de las ecuaciones normales extendidas para la predicción lineal hacia atrás:

$$\Phi_{M+1}(n)\mathbf{c}_M(n) = \begin{bmatrix} \mathbf{0} \\ \mathcal{B}_M(n) \end{bmatrix} \quad (3.78)$$

donde  $\mathbf{0}$  es el vector nulo de  $M \times 1$ . El escalar  $\mathcal{B}(n)$  es el mínimo valor de la suma del cuadrado de los errores a posteriori de predicción ponderados:

$$\mathcal{B}_M(n) = \sum_{i=1}^n \lambda^{n-i} |b_M(i)|^2 \quad (3.79)$$

La matriz  $\Phi_{M+1}(n)$  de  $(M+1) \times (M+1)$  es la matriz de autocorrelación del vector de muestras de entrada al filtro de predicción del error  $\mathbf{u}_{M+1}(i)$ . Utilizando la ec. 3.76 la matriz  $\Phi_{M+1}(n)$  se puede expresar en forma particionada como [Alc86, Hay91]:

$$\Phi_{M+1}(n) = \begin{bmatrix} \Phi_M(n) & \theta_2(n) \\ \theta_2^H(n) & \mathcal{U}_2(n) \end{bmatrix} \quad (3.80)$$

donde  $\mathcal{U}_2(n)$  es la suma del cuadrado de los valores ponderados de la respuesta deseada para la predicción lineal hacia atrás:

$$\begin{aligned} \mathcal{U}_2(n) &= \sum_{i=1}^n \lambda^{n-i} |u(i-M)|^2 \\ &= \sum_{i=1}^{n-M} \lambda^{(n-M)-i} |u(i)|^2 \end{aligned} \quad (3.81)$$

$\theta_2(n)$  es el vector de  $M \times 1$  de la correlación cruzada de las muestras de entrada al predictor de la fig. 3.3 y la respuesta deseada  $u(n - M)$ :

$$\theta_2(n) = \sum_{i=1}^N \lambda^{n-i} \mathbf{u}_M(i) \mathbf{u}_M^H(i) \quad (3.82)$$

De una manera análoga al caso de la predicción lineal adaptiva hacia adelante para poder desarrollar la teoría correspondiente a la predicción lineal adaptiva hacia atrás se recurre a la analogía entre los parámetros del algoritmo recursivo de mínimos cuadrados y los de la predicción lineal hacia atrás (ver tabla 3.1). De esta manera utilizando las ecs. (2.60), (2.63) y (2.64) podemos deducir la siguiente recursión para el vector de coeficientes del predictor hacia atrás y el vector de coeficientes del predictor hacia atrás [Alc86, Hay91]:

$$\mathbf{g}(n) = \mathbf{g}(n - 1) + \mathbf{k}_M(n) \psi_M^*(n) \quad (3.83)$$

donde  $\psi_M(n)$  es el error a priori de predicción hacia atrás:

$$\psi_M(n) = u(n - M) - \mathbf{g}^H(n - 1) \mathbf{u}_M(n) \quad (3.84)$$

y  $\mathbf{k}_M(n)$  es el vector de ganancia para la predicción lineal hacia atrás de orden  $M$ :

$$\mathbf{k}_M(n) = \Phi_M^{-1}(n) \mathbf{u}_M(n) \quad (3.85)$$

Sustituyendo 3.83 en 3.75 podemos expresar la recursión para el vector de coeficientes del filtro del error de predicción hacia atrás como:

$$\mathbf{c}_M(n) = \mathbf{c}_M(n - 1) - \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} \psi_M^*(n) \quad (3.86)$$

donde:

$$\begin{aligned} \psi_M(n) &= \begin{bmatrix} -\mathbf{g}^H(n - 1), 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_M(n) \\ u(n - M) \end{bmatrix} \\ &= \mathbf{c}_M^H(n - 1) \mathbf{u}_{M-1}(n) \end{aligned} \quad (3.87)$$

Finalmente de la ec. 3.71 se deduce la siguiente recursión para el mínimo valor de la suma del cuadrado de los errores a posteriori ponderados de predicción:

$$\mathcal{B}_M(n) = \lambda \mathcal{B}_M(n - 1) + \psi_M(n) \psi_M^*(n) \quad (3.88)$$

□.

### 3.3.1.3 Recursión para el Vector de Ganancia

Por extensión de la ecuación 3.85 se puede definir el vector de ganancia al orden  $M+1$  como:

$$\mathbf{k}_{M+1}(n) = \Phi^{-1}(n)\mathbf{u}_{M+1}(n) \quad (3.89)$$

donde  $\Phi_{M+1}^{-1}(n)$  esta dada por la ecuación 3.62. La inversa de 3.62 se halla aplicando el lema de inversión matricial (ver apéndice A):

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n-1) \end{bmatrix} + \frac{1}{\mathcal{F}_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n) \quad (3.90)$$

donde  $\mathbf{0}_M$  es el vector nulo de  $M \times 1$ ,  $\mathbf{a}_M(n)$  es el vector de  $(M+1) \times 1$  de coeficientes del filtro de orden  $M$  de predicción hacia adelante del error, y  $\mathcal{F}_M(n)$  es el valor mínimo de la suma del cuadrado de los errores ponderados a posteriori de predicción hacia adelante.

Posmultiplicando ambos lados de la ecuación 3.90 por la partición de  $\mathbf{u}_{M+1}(n)$  dada por la ec. 3.57 (en la cual se reemplaza  $i$  por  $n$ ), y utilizando la ecuación 3.58 (para  $i = n$ ) tenemos la siguiente recursión para el vector de ganancia extendido [Alc86, Hay91]:

$$\mathbf{k}_{M+1}(n) = \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \frac{f_M(n)}{\mathcal{F}_M(n)} \mathbf{a}_M(n) \quad (3.91)$$

De una manera análoga se puede desarrollar la recursión para el vector de ganancia al orden  $M+1$  para el caso de la predicción lineal adaptiva hacia atrás [Alc86, Hay91]:

$$\mathbf{k}_{M+1}(n) = \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} + \frac{b_M(n)}{\mathcal{B}_M(n)} \mathbf{c}_M(n) \quad (3.92)$$

### 3.3.1.4 Factor de Conversión (Angulo Variable)

El vector de ganancia puede ser visto como la solución de las ecuaciones normales cuando la respuesta deseada esta dada por [Alc86, Hay91]:

$$d(i) = \begin{cases} 1, & i = n \\ 0, & i = 1, 2, \dots, n-1 \end{cases} \quad (3.93)$$

El error de estimación se define como la diferencia entre la respuesta deseada y su estimación:

$$\begin{aligned} \gamma_M(n) &= 1 - \mathbf{k}_M^H(n) \mathbf{u}_M(n) \\ &= 1 - \mathbf{u}_M^H(n) \Phi_M^{-1}(n) \mathbf{u}_M(n) \end{aligned} \quad (3.94)$$

El error de estimación representa la salida de un filtro transversal cuyo vector de coeficientes es igual al vector de ganancia  $\mathbf{k}_M(n)$ , y cuya entrada es el vector de muestras de entrada  $\mathbf{u}_M(n)$  (ver fig. 5.3).

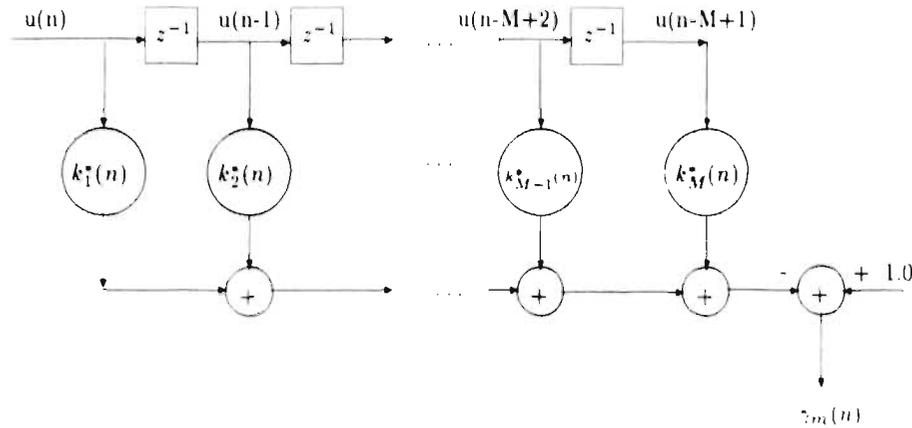


Figura 3.5: Filtro transversal utilizado para definir el error de estimación  $\gamma_M(n)$ .

El error de estimación,  $\gamma_M(n)$  tiene la siguiente propiedad [Alc86, Hay91]:

$$0 \leq \gamma_M(n) \leq 1 \quad (3.95)$$

El parámetro  $\gamma_M$  puede ser visto como un factor de conversión [GMK83]. De acuerdo a esta interpretación,  $\gamma_M$  nos sirve para determinar el valor del error a posteriori de estimación a partir del valor correspondiente del error a priori de estimación.  $\gamma_M$  tiene las siguientes interpretaciones como un factor de conversión:

1. Para una estimación recursiva por mínimos cuadrados tenemos:

$$\gamma_M(n) = \frac{\epsilon_M(n)}{\alpha_M(n)} \quad (3.96)$$

donde  $\epsilon_M(n)$  es el error a posteriori de estimación y  $\alpha_M(n)$  es el error a priori de estimación. Esta última ecuación establece que dado el error a priori de estimación  $\alpha_M(n)$ , podemos determinar el valor correspondiente del error a posteriori de estimación  $\epsilon_M(n)$  al multiplicar  $\alpha_M(n)$  por  $\gamma_M(n)$ . Por lo tanto podemos ver a  $\alpha_M(n)$  como un valor tentativo del error de estimación  $\epsilon_M(n)$  y a  $\gamma_M(n)$  como un factor correctivo multiplicativo.

2. Para la predicción lineal adaptiva hacia adelante, tenemos:

$$\gamma_M(n-1) = \frac{f_M(n)}{\eta_M(n)} \quad (3.97)$$

Por lo tanto podemos ver a  $\eta_M(n)$  como un valor tentativo del error a priori de predicción hacia adelante  $f_M(n)$  y a  $\gamma_M(n-1)$  como el factor de corrección.

3. Para la predicción lineal adaptiva hacia atrás, tenemos:

$$\gamma_M(n) = \frac{b_M(n)}{\psi_M(n)} \quad (3.98)$$

4. Por lo tanto, se puede ver a  $\psi_M(n)$  como un valor tentativo para el error de predicción hacia atrás  $b_M(n)$  y a  $\gamma_M(n)$  como el factor de corrección.

Concluimos que a través del uso de este factor de conversión podemos calcular los errores a posteriori  $\epsilon_M$ ,  $f_M(n)$  y  $b_M(n)$  al tiempo  $n$  antes de que los vectores de coeficientes de los filtros pertinentes ( esto es:  $\hat{\mathbf{w}}_M(n)$ ,  $\mathbf{a}_M(n)$ , y  $\mathbf{c}_M(n)$  ) sean calculados [Alc86, Hay91].

**Diferentes Recursiones Para El Factor de Conversión.** A continuación se presentan cuatro diferentes recursiones para el factor de conversión, las cuales probarán su utilidad en el desarrollo de las distintas versiones de los algoritmos rápidos [Alc86, Hay91]:

- La primera recursión involucra tanto al orden como al tiempo:

$$\gamma_{M+1}(n) = \gamma_M(n-1) - \frac{|f_M(n)|^2}{\mathcal{F}_M(n)} \quad (3.99)$$

- La siguiente recursión es sobre el orden del factor de conversión:

$$\gamma_{M+1}(n) = \gamma_M(n) - \frac{|b_M(n)|^2}{\mathcal{B}_M(n)} \quad (3.100)$$

- Finalmente se obtienen las dos siguientes recursiones al manipular las dos anteriores [Alc86, Hay91]:

$$\gamma_{M+1}(n) = \lambda \frac{\mathcal{F}_M(n-1)}{\mathcal{F}_M(n)} \gamma_M(n-1) \quad (3.101)$$

$$\gamma_{M+1}(n) = \lambda \frac{\mathcal{B}_M(n-1)}{\mathcal{B}_M(n)} \gamma_M(n) \quad (3.102)$$

### 3.3.1.5 Resumen de Fórmulas

Predicción Lineal Adaptiva Hacia Adelante	
$\mathbf{a}_M(n) = \mathbf{a}_M(n-1) - \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} \eta_M^*(n)$	(3.69)
$\eta_M(n) = \mathbf{a}_M^H(n-1) \mathbf{u}_{M+1}(n)$	(3.70)
$\mathcal{F}_M(n) = \lambda \mathcal{F}_M(n-1) + \eta_M(n) f_M^*(n)$	(3.72)

Tabla 3.3: Predicción lineal Adaptiva hacia adelante.

Predicción Lineal Adaptiva Hacia Atrás	
$\mathbf{c}_M(n) = \mathbf{c}_M(n-1) - \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} \psi_M^*(n)$	(3.86)
$\psi_M(n) = \mathbf{c}_M^H(n-1) \mathbf{u}_{M+1}(n)$	(3.87)
$\mathcal{B}_M(n) = \lambda \mathcal{B}_M(n-1) + \psi_M(n) b_M^*(n)$	(3.88)

Tabla 3.4: Predicción lineal Adaptiva hacia atrás.

Vector de Ganancia	
$\mathbf{k}_{M+1}(n) = \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \frac{f_M(n)}{\mathcal{F}_M(n)} \mathbf{a}_M(n)$	(3.91)
$\mathbf{k}_{M+1}(n) = \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} + \frac{b_M(n)}{\mathcal{B}_M(n)} \mathbf{c}_M(n)$	(3.92)

Tabla 3.5: Vector de Ganancia.

Factor de Conversión	
Definiciones del Factor de Conversión	
$\gamma_M(n) = \frac{e_M(n)}{\alpha_M(n)}$	(3.96)
$\gamma_M(n-1) = \frac{f_M(n)}{\eta_M(n)}$	(3.97)
$\gamma_M(n) = \frac{b_M(n)}{\nu_M(n)}$	(3.98)
Formulas de recursión:	
$\gamma_{M+1}(n) = \gamma_M(n-1) - \frac{ f_M(n) ^2}{\mathcal{F}_M(n)}$	(3.99)
$\gamma_{M+1}(n) = \gamma_M(n) - \frac{ b_M(n) ^2}{\mathcal{B}_M(n)}$	(3.100)
$\gamma_{M+1}(n) = \lambda \frac{\mathcal{F}_M(n-1)}{\mathcal{F}_M(n)} \gamma_M(n-1)$	(3.101)
$\gamma_{M+1}(n) = \lambda \frac{\mathcal{B}_M(n-1)}{\mathcal{B}_M(n)} \gamma_M(n)$	(3.102)

Tabla 3.6: Factor de Conversión

### 3.3.2 Algoritmos Rápidos: Estructura Transversal

El algoritmo de filtrado adaptable que se presenta en esta sección, se conoce como "fast transversal filters (FTF) algorithm" [GMK83] [CK84]. Este algoritmo se basa en los siguientes cuatro filtros transversales que fueron tratados en la sección anterior: el filtro de predicción del error hacia adelante con coeficientes  $\mathbf{a}_M$ , que para el desarrollo del algoritmo FTF y, con fines de simplicidad lo llamaremos simplemente filtro transversal A; el filtro de predicción del error hacia atrás con coeficientes  $\mathbf{c}_M$ , que lo llamaremos simplemente filtro transversal B; el filtro transversal de predicción del error de estimación con coeficientes  $\mathbf{K}_M$ , que lo llamaremos simplemente filtro transversal C y; el predictor hacia adelante de orden M con coeficientes  $\mathbf{w}_M$ , que lo llamaremos simplemente filtro transversal D.

#### 3.3.2.1 Desarrollo del Algoritmo FTF

El desarrollo del algoritmo FTF es como sigue: recordemos que al aplicar el vector  $\mathbf{u}_{M+1}(n)$  a la entrada del filtro A, obtenemos el error a priori de predicción hacia adelante (ver ec. 3.70):

$$\eta_M(n) = \mathbf{a}_M^H(n-1)\mathbf{u}_{M+1}(n) \quad (3.103)$$

Al multiplicar  $\eta_M(n)$  por el error de estimación  $\gamma_M(n-1)$  obtenemos el error a posteriori de predicción hacia adelante (ver ec. 3.97):

$$f_M(n) = \gamma_M(n-1)\eta_M(n) \quad (3.104)$$

La recursión para la suma del cuadrado de los errores ponderados de predicción hacia adelante es la siguiente (ver ec. 3.72):

$$\mathcal{F}_M(n) = \lambda\mathcal{F}_M(n-1) + \eta_M(n)f_M^*(n) \quad (3.105)$$

A continuación usamos la ec. 3.101 para actualizar el factor de conversión:

$$\gamma_{M+1}(n) = \lambda \frac{\mathcal{F}_M(n-1)}{\mathcal{F}_M(n)} \gamma_M(n-1) \quad (3.106)$$

Para obtener la recursión para el vector de ganancia procedemos de la siguiente manera: sustituimos la ec. 3.69 en la ec. 3.91 con lo cual obtenemos:

$$\mathbf{k}_{M+1}(n) = \left(1 - \frac{\eta_M^*(n)f_M(n)}{\mathcal{F}_M(n)}\right) \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \frac{f_M(n)}{\mathcal{F}_M(n)} \mathbf{a}_M(n-1) \quad (3.107)$$

Usando la ec. 3.71 y luego la 3.101 tenemos:

$$\begin{aligned} 1 - \frac{\eta_M^*(n)f_M(n)}{\mathcal{F}_M(n)} &= \lambda \frac{\mathcal{F}_M(n-1)}{\mathcal{F}_M(n)} \\ &= \frac{\gamma_{M+1}(n)}{\gamma_M(n-1)} \end{aligned} \quad (3.108)$$

Usando la ec. 3.96 y a continuación la ec. 3.101 tenemos:

$$\begin{aligned} \frac{f_M(n)}{\mathcal{F}_M(n)} &= \frac{\eta_M(n)\gamma_M(n-1)}{\mathcal{F}_M(n)} \\ &= \lambda^{-1} \frac{\eta_M(n)\gamma_{M+1}(n)}{\mathcal{F}_M(n-1)} \end{aligned} \quad (3.109)$$

Finalmente la recursión para el vector de ganancia se obtiene sustituyendo la ec. 3.109 y la 3.108 en la 3.107:

$$\mathbf{k}_{M+1}(n) = \frac{\gamma_{M+1}(n)}{\gamma_M(n-1)} \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \lambda^{-1} \frac{\eta_M(n)\gamma_{M+1}(n)}{\mathcal{F}_M(n-1)} \mathbf{a}_M(n-1) \quad (3.110)$$

Por conveniencia en esta presentación del desarrollo del algoritmo FTF definimos el vector de ganancia normalizado:

$$\tilde{\mathbf{k}}_M(n) = \frac{\mathbf{k}_M(n)}{\gamma_M(n)} \quad (3.111)$$

Con lo cual la recursión 3.110 puede expresarse como:

$$\tilde{\mathbf{k}}_{M+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_M(n-1) \end{bmatrix} + \lambda^{-1} \frac{\eta_M(n)}{\mathcal{F}_M(n-1)} \mathbf{a}_M(n-1) \quad (3.112)$$

Esta última ecuación es la recursión que utiliza el algoritmo FTF para actualizar el vector de ganancia normalizado.

A continuación se utiliza la ec. 3.111 y la ec. 3.96 con lo cual podemos redefinir la recursión del vector de coeficientes del filtro de predicción del error hacia adelante dada por la ec. 3.69 como:

$$\begin{aligned} \mathbf{a}_M(n) &= \mathbf{a}_M(n-1) - \gamma_M(n-1)\eta_M^*(n) \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_M(n-1) \end{bmatrix} \\ &= \mathbf{a}_M(n-1) - f_M^*(n) \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_M(n-1) \end{bmatrix} \end{aligned} \quad (3.113)$$

Hasta este punto hemos actualizado temporalmente el filtro de predicción del error hacia adelante (filtro A) y el vector de ganancia (filtro C) no sólo en tiempo sino también en orden.

Nuestra siguiente tarea consiste en : (1) corregir el incremento en el orden del filtro C de M a M+1 causado por el uso de la ec. 3.106, y (2) encontrar la recursión para actualizar temporalmente el filtro de predicción del error hacia atrás (filtro B). Para realizar estas tareas, primero sustituimos la ec. 3.86 en la ec. 3.92 con lo cual después de agrupar términos semejantes obtenemos:

$$\mathbf{k}_{M+1}(n) = \left(1 - \frac{v_M^*(n)b_M(n)}{\mathcal{B}_M(n)}\right) \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} + \frac{b_M(n)}{\mathcal{B}_M(n)} \mathbf{c}_M(n-1) \quad (3.114)$$

Considerando los últimos elementos de los vectores de la ecuación anterior, hallamos que [Alc86, Hay91]:

$$k_{M+1,M+1}(n) = \frac{b_M(n)}{\mathcal{B}_M(n)} \quad (3.115)$$

Normalizando con respecto a  $\gamma_{M+1}(n)$  podemos expresar el último elemento de  $\tilde{\mathbf{K}}_{M+1}(n)$  como:

$$\begin{aligned} \hat{k}_{M+1,M+1} &= \frac{k_{M+1,M+1}(n)}{\gamma_{M+1}(n)} \\ &= \frac{b_M(n)}{\gamma_{M+1}(n)\mathcal{B}_M(n)} \end{aligned} \quad (3.116)$$

Utilizando las ecs. 3.98, 3.102 en 3.116, tenemos:

$$\hat{k}_{M+1,M+1}(n) = \frac{v_M(n)}{\gamma_{M+1}(n)\mathcal{B}_M(n)} \quad (3.117)$$

Se puede expresar el error a priori de predicción hacia atrás  $v_M(n)$  en términos del último elemento del vector de ganancia normalizado  $\tilde{\mathbf{k}}_{M+1}(n)$  como:

$$v_M(n) = \lambda \mathcal{B}_M(n-1) \hat{k}_{M+1,M+1}(n) \quad (3.118)$$

Resolviendo la ec. 3.102 para  $\gamma_M(n)$ , tenemos:

$$\gamma_M(n) = \frac{\mathcal{B}_M(n)}{\lambda \mathcal{B}_M(n-1)} \quad (3.119)$$

Usando la ec. 3.88 y luego la 3.116, podemos escribir:

$$\begin{aligned} \lambda \frac{\mathcal{B}_M(n-1)}{\mathcal{B}_M(n)} &= 1 - \frac{v_M(n) \hat{k}_{M+1,M+1}(n)}{\mathcal{B}_M(n)} \\ &= 1 - v_M(n) \gamma_{M+1}(n) \hat{k}_{M+1,M+1}(n) \end{aligned} \quad (3.120)$$

Sustituyendo la ec. 3.120 en la 3.119 tenemos:

$$\gamma_M(n) = \left[ 1 - v_M(n) \gamma_{M+1}(n) \hat{k}_{M+1,M+1}(n) \right]^{-1} \gamma_{M+1}(n) \quad (3.121)$$

Nótese que:

$$v_M(n) \hat{k}_{M+1,M+1}(n) = v_M^*(n) k_{M+1,M+1}(n) \quad (3.122)$$

La ecuación 3.121 corrige el incremento en el orden del filtro  $C$  de  $M$  a  $M+1$  que resulta del uso de la ecuación 3.106. Una vez que se han calculado los valores de  $\gamma_M(n)$  y de  $psi_M(n)$ , podemos utilizar la ec. 3.98 para calcular el error a posteriori de predicción hacia atrás:

$$b_M(n) = \gamma_M(n) v_M(n) \quad (3.123)$$

Además podemos usar la recursión expresada por la ec. 3.88 para actualizar el valor mínimo de la suma ponderada del cuadrado de los errores a posteriori de predicción hacia atrás:

$$\mathcal{B}_M(n) = \lambda \mathcal{B}_M(n-1) + v_M(n) b_M^*(n) \quad (3.124)$$

Mediante las ecs. 3.120, 3.102, 3.111 y 3.116 la ec. 3.111 puede expresarse como [Ale86, Hay91]:

$$\begin{bmatrix} \tilde{\mathbf{k}}_M(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{k}}_{M+1}(n) - \tilde{k}_{M+1,M+1}(n)\mathbf{c}_M(n-1) \quad (3.125)$$

A continuación se utiliza la ecuación 3.86 para actualizar el vector de coeficientes del filtro de predicción hacia atrás del error:

$$\begin{aligned} \mathbf{c}_M(n) &= \mathbf{c}_M(n-1) - \epsilon_M^*(n) \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} \\ &= \mathbf{c}_M(n-1) - b_M^*(n) \begin{bmatrix} \tilde{\mathbf{k}}_M(n) \\ 0 \end{bmatrix} \end{aligned} \quad (3.126)$$

donde en la última línea de la ecuación anterior se ha utilizado 3.98 y 3.111.

Hasta este punto hemos hallado la recursión para el vector de ganancia (filtro C) y para el filtro de predicción hacia atrás del error (filtro B). Ahora sólo hace falta encontrar la recursión para el vector de coeficientes del filtro D; lo cual se hace a continuación:

Se utiliza la ec. 3.79 para calcular el error a posteriori de estimación:

$$\alpha_M(n) \doteq d(n) - \hat{\mathbf{w}}_M^H(n-1)\mathbf{u}_M(n) \quad (3.127)$$

A continuación se usa la ec. 3.96 para calcular el error de estimación:

$$\epsilon_M(n) = \gamma_M(n)\text{pha}_M(n) \quad (3.128)$$

Finalmente se usa la ec. 3.78 para actualizar el vector de coeficientes del filtro adaptivo (filtro D):

$$\begin{aligned} \hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}_M(n-1) + \mathbf{k}_M(n)\alpha_M^*(n) \\ &= \hat{\mathbf{w}}_M(n-1) + \tilde{\mathbf{k}}_M\epsilon_M^*(n) \end{aligned} \quad (3.129)$$

donde en la última línea de la ecuación anterior se han utilizado las ecs. 3.128 y la 3.111.

### 3.3.2.2 Organigrama del Algoritmo FTF

En la siguiente tabla se muestra el organigrama del algoritmo FTF, donde se han recolectado las recursiones desarrolladas en la presente sección. Además se muestra la manera de inicializar el algoritmo [Alc86, Hay91]:

$n = 1$
$\mathbf{a}_0(1) = \mathbf{c}_0(1) = \mathbf{1} \quad \tilde{\mathbf{k}}_0(1) = 0$
$\hat{\mathbf{w}}_1(1) = d^*(1)/u^*(1) \quad \gamma_0(1) = 1$
$\mathcal{F}_0(1) =  u(1) ^2, \quad u(1) \neq 0$
Para $2 \leq n \leq M + 1$ :
$\eta_{n-2}(n) = \mathbf{a}_{n-2}^H(n-1)\mathbf{u}_{n-1}(n)$
$\mathbf{a}_{n-1}(n) = \begin{bmatrix} \mathbf{a}_{n-2}(n-1) \\ -\eta_{n-2}^*(n)/u^*(1) \end{bmatrix}$
$f_{n-2}(n) = \gamma_{n-2}(n-1)\eta_{n-2}(n)$
$\mathcal{F}_{n-1}(n) = \lambda \mathcal{F}_{n-2}(n-1)$
$\mathcal{F}_{n-2}(n) = \mathcal{F}_{n-1}(n) + \eta_{n-2}(n)f_{n-2}^*(n)$
$\gamma_{n-1}(n) = \frac{\mathcal{F}_{n-1}(n)}{\mathcal{F}_{n-2}(n)} \gamma_{n-2}(n-1)$
$\tilde{\mathbf{k}}_{n-1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{n-2}(n-1) \end{bmatrix} + \frac{\eta_{n-2}(n)}{\lambda \mathcal{F}_{n-2}(n-1)} \mathbf{a}_{n-2}(n-1)$
$\mathbf{c}_{n-1}(n) = \begin{bmatrix} -u^*(1)\gamma_{n-1}(n)\tilde{\mathbf{k}}_{n-1}(n) \end{bmatrix}$ , cálculese cuando $n = M + 1$
$\mathcal{E}_{n-1}(n) = \gamma_{n-1}(n) u(1) ^2$ , cálculese cuando $n = M + 1$
$\alpha_{n-1}(n) = d(n) - \hat{\mathbf{w}}_{n-1}^H(n-1)\mathbf{u}_{n-1}(n)$
$\hat{\alpha}_{n-1}(n) = d(n) - \hat{\mathbf{w}}_{n-1}^H(n-1)\mathbf{u}_{n-1}(n)$
$\epsilon_{n-1}(n) = \gamma_{n-1}(n)\alpha_{n-1}(n)$
Si: $n \leq M$ , $\hat{\mathbf{w}}_n(n) = \begin{bmatrix} \hat{\mathbf{w}}_{n-1}(n-1) \\ \hat{\alpha}_{n-1}^*(n)/u^*(1) \end{bmatrix}$
Si $n = M + 1$ , $\hat{\mathbf{w}}_{n-1}(n) = \hat{\mathbf{w}}_{n-1}(n-1) + \hat{\mathbf{k}}_{n-1}(n)\epsilon_{n-1}^*(n)$

Tabla 3.7: Organigrama del algoritmo FTF

### 3.3.3 Algoritmos Rápidos: Estructura "Lattice"

En esta sección se desarrollará la segunda clase de algoritmos rápidos basados en un filtro "lattice". Los algoritmos aquí tratados son conocidos como: "recursive least

squares "lattice" (LSL) algorithms", los cuales involucran recursiones en el tiempo y en el orden de los parámetros que manejan. Su eficiencia es la misma que la del algoritmo FTF en el sentido de que logran la misma tasa de convergencia a expensas de que el tiempo de procesamiento se incremente linealmente con el número de parámetros ajustables [Alc86, Hay91]. También los algoritmos LSL poseen la misma insensibilidad a la dispersión de los eigen valores de la matriz de correlación de datos de entrada al filtro adaptivo.

### 3.3.3.1 Antecedentes.

Considere un filtro de predicción hacia adelante del error (ver fig. 3.2) cuyos coeficientes denotaremos con  $\mathbf{a}_m(n)$ . Por definición el primer elemento de  $\mathbf{a}_m(n)$  es igual a la unidad. Sea  $\mathbf{u}_{m+1}(i)$  el vector de  $M \times 1$  de muestras de entrada al filtro al tiempo  $i$ , donde  $1 \leq i \leq n$ . El error a posteriori de predicción hacia adelante producido a la salida del filtro como respuesta a la entrada  $\mathbf{u}_{m+1}(i)$ , está dado por:

$$f_M(i) = \mathbf{a}_m^H(n) \mathbf{u}_{m+1}(i), \quad 1 \leq i \leq n \quad (3.130)$$

Se asume el uso del "prewindowing", de tal manera que podemos hacer  $u(i) = 0$  para  $i \leq 0$ . La suma ponderada del cuadrado de los errores a posteriori de predicción hacia adelante es igual a:

$$\mathcal{F}_M(n) = \sum_{i=1}^n \lambda^{n-i} |f_m(i)|^2 \quad (3.131)$$

donde  $\lambda$  es el factor exponencial de ponderación. El vector de coeficientes del filtro  $\mathbf{a}_m(n)$  es el resultado de minimizar la función de costo  $\mathcal{F}_m(n)$ .

De una manera análoga a la que se procedió en el caso de la predicción hacia adelante, podemos derivar los resultados de la predicción hacia atrás: considere el filtro de predicción del error hacia atrás (ver fig. 3.4) cuyo vector de coeficientes denotaremos mediante  $\mathbf{c}_M(n)$ . Por definición el último elemento de  $\mathbf{c}_m(n)$  es igual a la unidad. El error a posteriori de predicción hacia atrás producido a la salida del filtro como respuesta a  $\mathbf{u}_m(n)$  es igual a:

$$b_m(i) = \mathbf{c}_m^H(n) \mathbf{u}_{M+1}(i), \quad 1 \leq i \leq n \quad (3.132)$$

La suma del cuadrado de los errores ponderados a posteriori de predicción hacia atrás esta dada por:

$$\mathcal{B}_M(n) = \sum_{i=1}^n \lambda^{n-i} |b_m(i)|^2 \quad (3.133)$$

El vector de coeficientes  $\mathbf{c}_m(n)$  es el resultado de minimizar la función de costo  $\mathcal{B}_m(n)$  sujeta a la restricción de que el último elemento de  $\mathbf{c}_m(n)$  sea igual a la unidad.

### 3.3.3.2 Recursiones en el Orden

Sea  $\Phi_{m+1}(n)$  la matriz, de  $(m+1) \times (m+1)$ , de correlación del vector de muestras de entrada  $\mathbf{u}_{m+1}(i)$ , al filtro (de orden  $m$ ) de predicción del error hacia adelante: donde  $1 \leq i \leq n$ . Se puede caracterizar al filtro de predicción del error hacia adelante mediante las ecuaciones normales extendidas (ver ec. 3.59):

$$\Phi_{m+1}(n)\mathbf{a}_m(n) = \begin{bmatrix} \mathcal{F}_m(n) \\ \mathbf{0}_m \end{bmatrix} \quad (3.134)$$

donde  $\mathbf{0}_m$  es el vector nulo de  $m \times 1$ .

A continuación utilizamos una partición de la matriz  $\Phi_{m+1}$  que nos permita relacionar el vector de coeficientes  $\mathbf{a}_m(n)$  relativo a la predicción de orden  $m$ , con el vector de coeficientes  $\mathbf{a}_{m-1}(n)$  relativo al predictor de orden  $m-1$ ; es decir utilizamos [Alc86, Hay91]:

$$\Phi_{m+1}(n) = \left[ \begin{array}{c|c} \Phi_m(n) & \theta_2(n) \\ \hline \theta_2^H(n) & \mathcal{U}_2(n) \end{array} \right] \quad (3.135)$$

donde  $\Phi_m(n)$  es la matriz (de  $m \times m$ ) de la autocorrelación del vector de muestras de entrada  $\mathbf{u}_m(i)$ ,  $\theta_2(n)$  es el vector (de  $m \times 1$ ) de la correlación cruzada de  $\mathbf{u}_m(i)$  y  $u(i-m)$ , y  $\mathcal{U}_2(n)$  es la suma ponderada del cuadrado de los valores de la entrada  $u(i-m)$  para  $1 \leq i \leq n$ . Multiplicando a 3.135 por un vector de  $(m+1) \times 1$  cuyos primeros  $m$  elementos son los elementos del vector  $\mathbf{a}_{m-1}(n)$  y cuyos últimos elementos son cero, tenemos: [Alc86, Hay91]:

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} &= \left[ \begin{array}{c|c} \Phi_m(n) & \theta_2(n) \\ \hline \theta_2^H(n) & \mathcal{U}_2(n) \end{array} \right] \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \Phi_m(n)\mathbf{a}_{m-1}(n) \\ \theta_2^H(n)\mathbf{a}_{m-1}(n) \end{bmatrix} \end{aligned} \quad (3.136)$$

Para un filtro de orden  $m-1$  de predicción del error hacia adelante, las ecuaciones normales extendidas al tiempo  $n$  están dadas por:

$$\Phi_m(n)\mathbf{a}_{m-1}(n) = \begin{bmatrix} \mathcal{F}_{m-1}(n) \\ \mathbf{0}_{m-1} \end{bmatrix} \quad (3.137)$$

donde  $\mathcal{F}_{m-1}(n)$  es la suma ponderada del cuadrado de los errores a posteriori de predicción hacia adelante, y  $\mathbf{0}_{m-1}$  es el vector nulo de  $(m-1) \times 1$ . Definamos el escalar:

$$\Delta_{m-1}(n) = \theta_2^H(n)\mathbf{a}_{m-1}(n) \quad (3.138)$$

Con lo cual podemos expresar la ec. 3.136 como:

$$\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{m-1}(n) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n) \end{bmatrix} \quad (3.139)$$

Considere enseguida el filtro de orden  $m$  de predicción hacia atrás del error que está caracterizado por las ecuaciones normales extendidas en su forma matricial:

$$\Phi_{m+1}(n)\mathbf{c}_m(n) = \begin{bmatrix} \mathbf{0}_m \\ \mathcal{B}_m(n) \end{bmatrix} \quad (3.140)$$

donde  $\mathbf{0}_m$  es el vector nulo de  $m \times 1$ . Para el caso backward se usa la siguiente partición de la matriz  $\Phi_m(n)$  [Alc86, Hay91]:

$$\Phi_{m+1}(n) = \left[ \begin{array}{c|c} \mathcal{U}_1(n) & \theta_1^H(n) \\ \hline \theta_1(n) & \Phi_m(n-1) \end{array} \right] \quad (3.141)$$

donde  $\mathcal{U}_1(n)$  es la suma ponderada de los valores al cuadrado de la entrada  $u(i)$  para el intervalo de tiempo:  $1 \leq i \leq n$ ,  $\theta_1(n)$  es el vector, de  $m \times 1$ , de la correlación cruzada de  $\mathbf{u}_m(i-1)$  y  $\Phi_m(n-1)$  es la matriz, de  $m \times m$ , de la autocorrelación de  $\mathbf{u}_m(i-1)$ . De una manera análoga al caso forward, posmultiplicamos a  $\Phi_{m+1}(n)$  por un vector de  $(m+1) \times 1$  cuyo primer elemento es cero y cuyos  $m$  elementos restantes son los elementos del vector  $\mathbf{c}_{m-1}(n-1)$ ; por lo tanto:

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} &= \begin{bmatrix} \mathcal{U}_1(n) & \theta_1^H(n) \\ \theta_1(n) & \Phi_m(n-1) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \\ &= \begin{bmatrix} \theta_1^H(n)\mathbf{c}_{m-1}(n-1) \\ \Phi_m(n-1)\mathbf{c}_{m-1}(n-1) \end{bmatrix} \end{aligned} \quad (3.142)$$

Para un filtro de orden  $m-1$  de predicción hacia atrás del error, las ecuaciones normales extendidas en forma matricial al tiempo  $n-1$  están dadas por:

$$\Phi_m(n-1)\mathbf{c}_{m-1}(n-1) = \begin{bmatrix} \mathbf{0}_{m-1} \\ \mathcal{B}_{m-1}(n-1) \end{bmatrix} \quad (3.143)$$

donde  $\mathcal{B}_{m-1}(n-1)$  es la suma ponderada del cuadrado de los errores a posteriori de predicción hacia atrás al tiempo  $n-1$ . Definamos el escalar:

$$\Delta'_{m-1}(n) = \theta_1^H(n)\mathbf{c}_{m-1}(n-1) \quad (3.144)$$

Entonces la ec. 3.136 se puede expresar como:

$$\Phi_{M+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} = \begin{bmatrix} \Delta'_{m-1}(n) \\ \mathbf{0}_{m-1} \\ \mathcal{B}_{m-1}(n-1) \end{bmatrix} \quad (3.145)$$

Multiplicando la ecuación anterior por  $\Delta_{m-1}(n)/\mathcal{B}_{m-1}(n-1)$  y restando el resultado de la ec. 3.139 tenemos [Alc86, Hay91]:

$$\begin{aligned} \Phi_{m+1}(n) \left[ \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \right] \\ = \begin{bmatrix} \mathcal{F}_{m-1}(n) - \frac{\Delta_{m-1}(n)\Delta'_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)} \\ \mathbf{0}_m \end{bmatrix} \end{aligned} \quad (3.146)$$

Mediante la comparación de 3.134 y 3.146 podemos deducir las dos siguientes recursiones en el orden:

$$\mathbf{a}_m(n) = \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} . \quad (3.147)$$

y:

$$\mathcal{F}_m(n) = \mathcal{F}_{m-1}(n) - \frac{\Delta_{m-1}(n)\Delta'_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)} \quad (3.148)$$

De una manera análoga que para la predicción hacia adelante, las recursiones en el orden para  $\mathbf{c}_m(n)$  y  $\mathcal{B}_m(n)$  pueden ser halladas de la siguiente manera: multiplicando la ec. 3.139 por  $\Delta'_{m-1}(n)/\mathcal{F}_{m-1}(n)$  y restando el resultado de la ec. 3.115 tenemos:

$$\begin{aligned} \Phi_{m+1}(n) & \left[ \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} - \frac{\Delta'_{m-1}(n)}{\mathcal{F}_{m-1}(n)} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \right] \\ & = \left[ \mathcal{B}_{m-1}(n-1) - \frac{\Delta_{m-1}(n)\Delta'_{m-1}(n)}{\mathcal{F}_{m-1}(n)} \right] \end{aligned} \quad (3.149)$$

Comparando las ecs. 3.140 con la ecuación anterior tenemos la siguientes recursiones en el orden:

$$\mathbf{c}_m(n) = \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} - \frac{\Delta'_{m-1}(n)}{\mathcal{F}_{m-1}(n)} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \quad (3.150)$$

y:

$$\mathcal{B}_m(n) = \mathcal{B}_{m-1}(n-1) - \frac{\Delta_{m-1}(n)\Delta'_{m-1}(n)}{\mathcal{F}_{m-1}(n)} \quad (3.151)$$

### 3.3.3.3 Relación entre los Parámetros $\Delta_{m-1}(n)$ y $\Delta'_{m-1}(n)$

Los parámetros  $\Delta_{m-1}(n)$  y  $\Delta'_{m-1}$ , definidos por las ecs. 3.138 y 3.144 respectivamente, están relacionados mediante [Ale86, Hay91]:

$$\Delta'_{m-1}(n) = \Delta_{m-1}^*(n) \quad (3.152)$$

donde  $\Delta_{m-1}^*(n)$  es el conjugado de  $\Delta_{m-1}(n)$ .

### 3.3.3.4 Recursiones en el Orden para los Errores A Posteriori de Predicción

Se define el coeficiente de reflexión hacia adelante como:

$$\Gamma_{f,m}(n) = -\frac{\Delta_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)}, \quad m = 1, 2, \dots, M \quad (3.153)$$

donde M es el orden final del predictor "lattice".

De una manera semejante se define el coeficiente de reflexión hacia atrás como:

$$\begin{aligned}\Gamma_{b,m}(n) &= -\frac{\Delta'_{m-1}(n)}{\mathcal{F}_{m-1}(n)} \\ &= -\frac{\Delta^*_{m-1}(n)}{\mathcal{F}_{m-1}(n)}, \quad m = 1, 2, \dots, M\end{aligned}\quad (3.154)$$

Utilizando estas definiciones las ecuaciones 3.147 y 3.150 pueden expresarse como sigue:

$$\mathbf{a}_m(n) = \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} + \Gamma_{f,m}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \quad (3.155)$$

y:

$$\mathbf{c}_m(n) = \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} + \Gamma_{b,m}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \quad (3.156)$$

donde  $m = 1, 2, \dots, M$ . Estas dos últimas ecuaciones pueden ser vistas como la contra parte determinística de la recursión de Levinson-Durbin.

El error a posteriori de predicción hacia adelante  $f_m(n)$  es la respuesta del filtro de orden  $m$  de predicción hacia adelante del error cuando la entrada es  $\mathbf{u}_{M+1}(n)$ :

$$f_m(n) = \mathbf{a}_m^H(n) \mathbf{u}_{m+1}(n) \quad (3.157)$$

El vector de muestras de entrada al filtro,  $\mathbf{u}_{m+1}(n)$ , se puede particionar como:

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} \mathbf{u}_m(n) \\ u(n-m) \end{bmatrix} \quad (3.158)$$

ó de una manera alternativa como:

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} u(n) \\ \mathbf{u}_m(n-1) \end{bmatrix} \quad (3.159)$$

Sustituyendo la ec. 3.155 en la 3.157 y usando la forma particionada para  $\mathbf{u}_{m+1}$  dada por la ec. 3.158 para el lado derecho de 3.157 y la ec. 3.159 para el segundo término, tenemos la siguiente recursión en el orden para el error a posteriori de predicción hacia adelante:

$$f_m(n) = f_{m-1}(n) + \Gamma_{f,m}^*(n) b_{m-1}(n-1), \quad m = 1, 2, \dots, M \quad (3.160)$$

donde  $f_{m-1}(n)$  es el error a posteriori de predicción hacia adelante al orden  $m-1$ , y  $b_{m-1}(n-1)$  es la versión retrasada del error a posteriori de predicción hacia atrás también al orden  $m-1$ .

Para el caso de la predicción hacia atrás se procede de una manera análoga al caso de la predicción hacia adelante: el error a posteriori de predicción hacia atrás  $b_m(n)$  es igual a la respuesta del filtro de orden  $m$  de predicción hacia atrás del error, cuando la entrada es  $\mathbf{u}_{m+1}(n)$ :

$$b_m(n) = \mathbf{c}_m^H(n) \mathbf{u}_{m+1}(n) \quad (3.161)$$

Sustituyendo la ec. 3.156 en la ec. 3.161, y usando la forma particionada de  $\mathbf{u}_{m+1}(n)$  dada por la ec. 3.159 para el lado derecho de 3.156 y la ec. 3.158 para el segundo término, tenemos la siguiente recursión en el orden para el error a posteriori de predicción hacia atrás:

$$b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m}^*(n)f_{m-1}(n), \quad m = 1, 2, \dots, M \quad (3.162)$$

Las ecuaciones 3.160 y 3.162 están representadas por el diagrama de bloques que se muestra en la fig. 3.6.

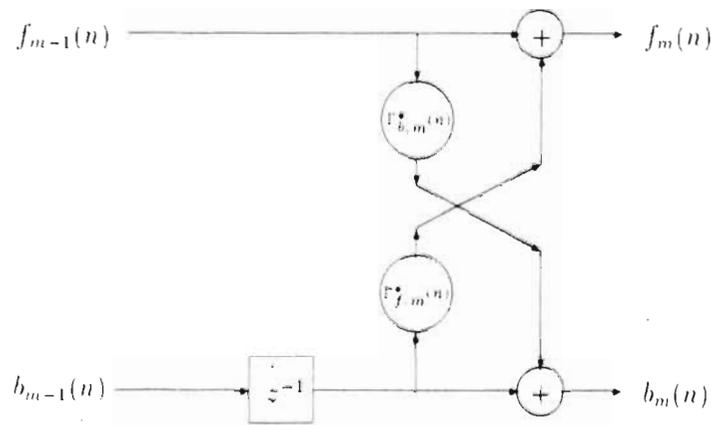


Figura 3.6: Etapa del Filtro "Lattice".

El orden,  $m$ , de predicción es un parámetro que toma los valores  $0, 1, \dots, M$ , donde  $M$  es el valor final del orden del predictor. Cuando  $m = 0$ , no se efectúa predicción alguna sobre los datos de entrada. Esta condición corresponde a los valores iniciales:

$$f_0(n) = b_0(n) = u(n) \quad (3.163)$$

donde  $u(n)$  es la muestra de entrada al filtro en el tiempo  $n$ . Por lo tanto, variamos el orden del predictor desde cero hasta  $M$ , para obtener el predictor en "lattice" que se muestra en la figura 3.7.

### 3.3.3.5 Recursiones en el Orden para las Sumas Ponderadas del Cuadrado de los Errores A Posteriori de Predicción

Las recursiones para la suma ponderada de los errores a posteriori tanto de predicción hacia adelante como de predicción hacia atrás están dadas por las ecuaciones 3.148 y 3.151, respectivamente. Usando la relación entre  $\Delta'_{m-1}(n)$  y  $\Delta_{m-1}(n)$  dada por la ecuación 3.152, podemos expresar estas dos recursiones como:

$$\mathcal{F}_m(n) = \mathcal{F}_{m-1}(n) - \frac{|\Delta_{m-1}(n)|^2}{\mathcal{B}_{m-1}(n-1)} \quad (3.164)$$

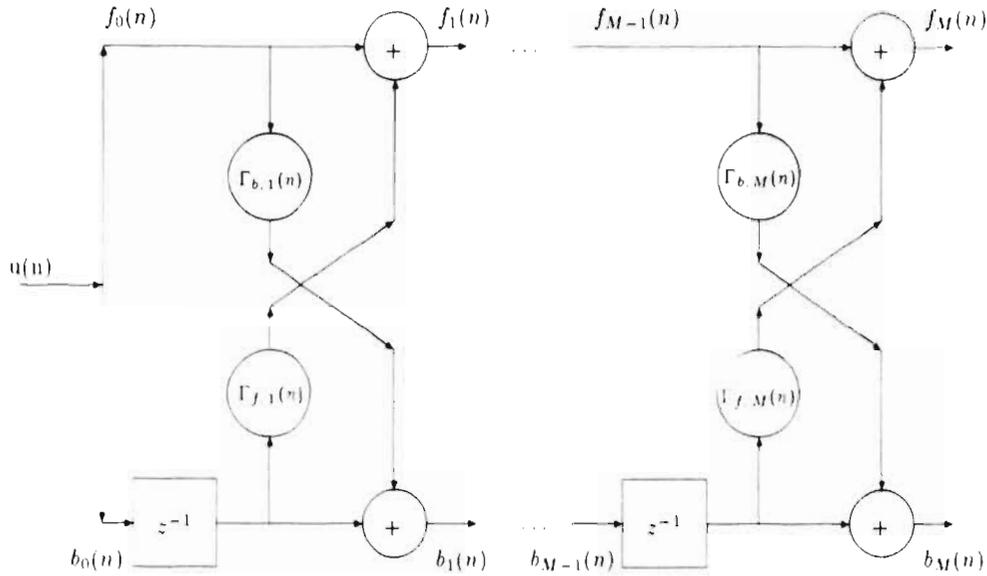


Figura 3.7: Filtro "Lattice".

y:

$$\mathcal{B}_m(n) = \mathcal{B}_{m-1}(n-1) - \frac{|\Delta_{m-1}(n)|^2}{\mathcal{F}_{m-1}(n)} \quad (3.165)$$

donde  $m = 1, 2, \dots, M$ . Las ecuaciones 3.160, 3.162, 3.164 y 3.165, junto con las definiciones de los coeficientes de reflexión 3.153 y 3.154, constituyen básicamente las recursiones en el orden para el predictor "lattice" [Ale86, Hay91].

### 3.3.3.6 Recursión en el Orden para $\gamma_m(n-1)$

El error de estimación ó factor de conversión  $\gamma_m(n-1)$  resulta de aplicar el vector de muestras de entrada  $\mathbf{u}_m(n-1)$  al filtro transversal cuyo vector de coeficientes es el vector de ganancia  $\mathbf{k}_m(n-1)$ . Recordemos que este último vector se actualiza de la siguiente manera (ver ec. 3.92):

$$\mathbf{k}_m(n-1) = \begin{bmatrix} \mathbf{k}_{m-1}(n-1) \\ 0 \end{bmatrix} + \frac{b_{m-1}(n-1)}{\mathcal{B}_{m-1}(n-1)} \mathbf{c}_{m-1}(n-1) \quad (3.166)$$

donde  $m = 1, 2, \dots, M$ . Posmultiplicando la Herimiteana de la ecuación anterior por  $\mathbf{u}_m(n-1)$ , tenemos:

$$\begin{aligned} \mathbf{k}_m^H(n-1) \mathbf{u}_m(n-1) &= \left[ \mathbf{k}_{m-1}^H(n-1), 0 \right] \mathbf{u}_m(n-1) \\ &+ \frac{b_{m-1}^*(n-1)}{\mathcal{B}_{m-1}(n-1)} \mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) \end{aligned} \quad (3.167)$$

Por definición tenemos:

$$\mathbf{k}_m^H(n-1)\mathbf{u}_m(n-1) = 1 - \gamma_m(n-1) \quad (3.168)$$

$$\begin{aligned} \mathbf{k}_m^H(n-1)\mathbf{u}_m(n-1) &= \begin{bmatrix} \mathbf{k}_{m-1}^H(n-1), & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{m-1}(n-1) \\ u(n-m) \end{bmatrix} \\ &= \mathbf{k}_{m-1}^H(n-1)\mathbf{u}_{m-1}(n-1) \\ &= 1 - \gamma_{m-1}(n-1) \end{aligned} \quad (3.169)$$

y:

$$\mathbf{c}_{m-1}^H(n-1)\mathbf{u}_m(n-1) = b_{m-1}(n-1) \quad (3.170)$$

Por lo tanto usando las ecuaciones de la 3.168 a la 3.170 en la 3.167 tenemos:

$$\gamma_m(n-1) = \gamma_{m-1}(n-1) - \frac{|b_{m-1}(n-1)|^2}{\mathcal{B}_{m-1}(n-1)}, \quad m = 1, 2, \dots, M \quad (3.171)$$

Esta es la recursión en el orden para  $\gamma_m(n-1)$ .

### 3.3.3.7 Recursiones en el Tiempo

Las recursiones anteriormente desarrolladas son sobre el orden de los predictores, sin embargo debido a la naturaleza adaptiva del algoritmo aquí desarrollado es necesario que éstas recursiones también actúen sobre el tiempo, para lo cual se debe derivar una recursión en el tiempo para el parámetro  $\Delta_{m-1}(n)$ , la cuál involucre el cálculo de las recursiones en el orden [Alc86, Hay91].

Considere el vector de coeficientes  $\mathbf{a}_{m-1}(n-1)$  relativo al filtro de predicción del error hacia adelante de orden  $m-1$ , evaluado al tiempo  $n-1$ . Dado que el primer elemento de  $\mathbf{a}_{m-1}(n-1)$  es uno, podemos expresar  $\Delta_{m-1}(n)$  como sigue [Alc86, Hay91]:

$$\Delta_{m-1}(n) = \begin{bmatrix} \Delta_{m-1}(n), & \mathbf{0}_{m-1}^T, & \mathcal{B}_{m-1}(n-1) \end{bmatrix} \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (3.172)$$

Al tomar la Hermitiana de la ec. 3.172 y usando la ec. 3.152 tenemos:

$$\begin{bmatrix} 0, & \mathbf{c}_{m-1}^H(n-1) \end{bmatrix} \Phi_{m+1}(n) = \begin{bmatrix} \Delta_{m-1}(n), & \mathbf{0}_{m-1}^T, & \mathcal{B}_{m-1}(n-1) \end{bmatrix} \quad (3.173)$$

La sustitución de 3.173 en 3.172 da:

$$\Delta_{m-1}(n) = \begin{bmatrix} 0, & \mathbf{c}_{m-1}^H(n-1) \end{bmatrix} \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (3.174)$$

La recursión en el tiempo para la matriz de correlación  $\Phi_{m+1}(n)$  está dada por (ver ec. 2.52):

$$\Phi_{m+1}(n) = \lambda \Phi_{m+1}(n-1) + \mathbf{u}_{m+1}(n)\mathbf{u}_{m+1}^H(n) \quad (3.175)$$

Esta última ecuación se puede utilizar para expresar 3.174 como:

$$\begin{aligned} \Delta_{m-1}(n) &= \lambda \left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &\quad + \left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \mathbf{u}_{m+1}(n) \mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \end{aligned} \quad (3.176)$$

De la definición del error a priori de predicción hacia adelante tenemos:

$$\begin{aligned} \mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} &= \left[ \mathbf{u}_m^H(n), u^*(n-m) \right] \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &= \mathbf{u}_m^H(n) \mathbf{a}_{m-1}(n-1) \\ &= \eta_{m-1}^*(n) \end{aligned} \quad (3.177)$$

y de la definición del error a posteriori de predicción hacia atrás tenemos:

$$\begin{aligned} \left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \mathbf{u}_{m+1}(n) &= \left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \begin{bmatrix} u(n) \\ \mathbf{u}_m(n-1) \end{bmatrix} \\ &= \mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) \\ &= b_{m-1}(n-1) \end{aligned} \quad (3.178)$$

Al sustituir  $n-1$  por  $n$  en 3.139 tenemos:

$$\Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix} \quad (3.179)$$

Usando esta última ecuación y el hecho de que el último elemento del vector  $\mathbf{c}_{m-1}(n-1)$  es igual a uno; podemos escribir el primer término del lado derecho de la ec. 3.176 (excepto por el factor  $\lambda$ ) como:

$$\begin{aligned} &\left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &= \left[ 0, \mathbf{c}_{m-1}^H(n-1) \right] \begin{bmatrix} \mathcal{F}_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix} \\ &= \Delta_{m-1}(n-1) \end{aligned} \quad (3.180)$$

Sustituyendo las ecs. 3.177, 3.178 y 3.180 en 3.176, podemos expresar la recursión en el tiempo para  $\Delta_{m-1}(n)$  simplemente como [Alc86, Hay91]:

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + b_{m-1}(n-1) \eta_{m-1}^*(n) \quad (3.181)$$

El error a priori de predicción hacia adelante  $\eta_{m-1}(n)$  esta relacionado al error a posteriori de predicción hacia adelante  $f_{m-1}(n)$  mediante (ver ec. 3.97):

$$\eta_{m-1}(n) = \frac{f_{m-1}(n)}{\gamma_{m-1}(n-1)} \quad (3.182)$$

Finalmente, al utilizar la ec. 3.182 en la 3.181 da por resultado la recursión en el tiempo para el factor de conversión:

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + \frac{b_{m-1}(n-1) f_{m-1}^*(n)}{\gamma_{m-1}(n-1)} \quad (3.183)$$

El término de corrección en esta última ecuación esta amplificado por el recíproco del factor de conversión  $\gamma_{m-1}(n-1)$ . Este parámetro permite al algoritmo LSL adaptarse rápidamente a los cambios repentinos en los datos de entrada [ML78].

### 3.3.3.8 Proceso de Estimación Conjunta

Para un filtro predictor en estructura "lattice" de  $m$  etapas, el algoritmo LSL produce una secuencia de errores de predicción hacia atrás:  $b_0(n), b_1(n), \dots, b_m(n)$ , los cuales están decorrelacionados entre sí. Esto es la matriz de autocorrelación de los errores de predicción hacia atrás es una matriz diagonal. Esta propiedad del algoritmo LSL se conoce como propiedad de desacoplamiento [Alc86, Hay91].

Si se utilizan los errores de predicción hacia atrás  $b_0(n), b_1(n), \dots, b_m(n)$  como entradas a un filtro transversal cuyo vector de coeficientes esta formado por los coeficientes de regresión:  $\kappa_0, \kappa_1, \dots, \kappa_m$ , (ver fig. 3.8), obtenemos el estimado de la respuesta deseada  $d(n)$  [Alc86, Hay91]. Nos referimos a la estructura de la figura 3.8 como el estimador conjunto dado que resuelve el problema de estimación del proceso  $\{d(n)\}$  a partir de las observaciones del proceso  $\{u(n)\}$  al formar el proceso conjunto  $\{d(n), u(n)\}$ .

**Propiedad de Desacoplamiento del Algoritmo LSL** A continuación se demostrará la propiedad de desacoplamiento del algoritmo LSL.

Considere un filtro de orden  $m$  de predicción hacia atrás del error, cuyo vector de coeficientes optimizado de acuerdo al criterio de los mínimos cuadrados en el intervalo  $1 \leq i \leq n$ , se denota mediante  $\mathbf{c}_m(n)$ . De una forma explícita tenemos:

$$\mathbf{c}_m^T(n) = [c_{m,m}(n), c_{m,m-1}(n), \dots, 1] \quad (3.184)$$

Sea  $b_m(i)$ , el error a posteriori de predicción hacia atrás, la respuesta del filtro cuando la entrada es el vector de  $m \times 1$ :  $\mathbf{u}_{m+1}(i)$ . En forma explícita tenemos:

$$\mathbf{u}_{m+1}^T(i) = [u(i), u(i-1), \dots, u(i-m)], \quad i > m \quad (3.185)$$

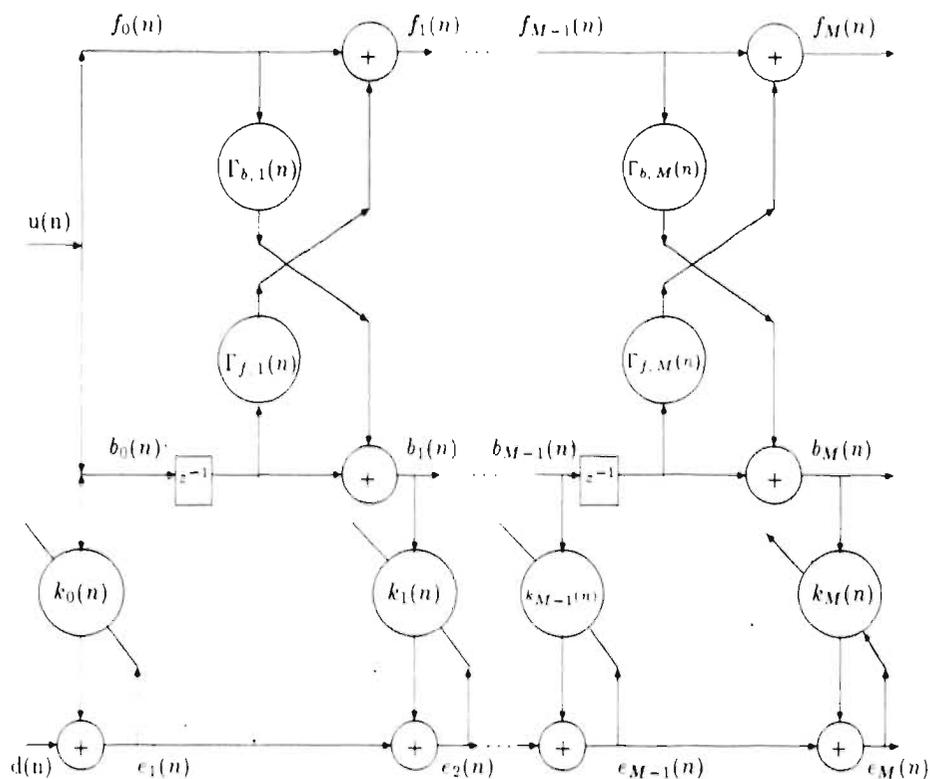


Figura 3.8: Proceso de estimación conjunta.

Por lo tanto, podemos expresar el error  $b_m(i)$  como:

$$\begin{aligned}
 b_m(i) &= \mathbf{c}_{m+1}^H(n) \mathbf{u}_{m+1}(i) \\
 &= \sum_{k=0}^m c_{m,k}^*(n) u(i - m + k), \quad \begin{array}{l} m < i \leq n, \\ m = 0, 1, 2, \dots \end{array}
 \end{aligned} \tag{3.186}$$

Sea  $\mathbf{b}_{m+1}(i)$  el vector de  $m \times 1$  cuyos elementos son los errores a posteriori de predicción hacia atrás:

$$\mathbf{b}_{m+1}^T(i) = [b_0(i), b_1(i), \dots, b_m(i)], \quad \begin{array}{l} i > m \\ m = 0, 1, 2, \dots \end{array} \tag{3.187}$$

Sustituyendo la ecuación 3.186 en la ecuación anterior, podemos expresar la transformación de los datos de entrada en su correspondiente conjunto de errores a posteriori de predicción hacia atrás como:

$$\mathbf{b}_{m+1}(i) = \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) \tag{3.188}$$

donde la matriz de transformación,  $\mathbf{L}_m(n)$ , de dimensión  $(m+1) \times (m+1)$  esta definida como:

$$\mathbf{L}_m(n) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ c_{1,1}^*(n) & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,m}^*(n) & c_{m,m-1}^*(n) & \cdots & 1 \end{bmatrix} \quad (3.189)$$

El indice  $m$  en  $\mathbf{L}_m(n)$  se refiere al orden más alto del filtro de predicción hacia atrás del error. Nótese:

- Los elementos diferentes de cero en la fila  $l$  de la matriz  $\mathbf{L}_m(n)$  son los coeficientes del filtro de predicción hacia atrás del error, cuyo orden es  $(l - 1)$ .
- Los elementos de la diagonal principal de  $\mathbf{L}_m(n)$  son iguales a uno; dado que el ultimo elemento del vector de coeficientes del filtro de predicción hacia atrás del error es uno.
- El determinante de la matriz  $\mathbf{L}_m(n)$  es igual a uno para toda  $m$ ; de aquí que la matriz inversa de  $\mathbf{L}_m(n)$  exista [Alc86, Hay91].

Se define la matriz de correlación del vector  $\mathbf{b}_{m+1}$  como:

$$\mathbf{D}_{m+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{b}_{m+1}(i) \mathbf{b}_{m+1}^H(i), \quad \begin{array}{l} m < i \leq n, \\ m = 0, 1, 2, \dots \end{array} \quad (3.190)$$

Sustituyendo la ecuación anterior en 3.188, obtenemos:

$$\begin{aligned} \mathbf{D}_{m+1}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) \mathbf{u}_{m+1}^H(i) \mathbf{L}_m^H(n) \\ &= \mathbf{L}_m(n) \left[ \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{m+1}(i) \mathbf{u}_{m+1}^H(i) \right] \mathbf{L}_m^H(n) \end{aligned} \quad (3.191)$$

En la última línea de la ecuación anterior la expresión entre paréntesis es igual a la matriz de correlación del vector de muestras de entrada  $\mathbf{u} + m + 1(i)$ :

$$\mathbf{\Phi}_{m+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{m+1}(i) \mathbf{u}_{m+1}^H(i) \quad (3.192)$$

Con lo cual podemos expresar 3.191 como:

$$\mathbf{D}_{m+1}(n) = \mathbf{L}_m(n) \mathbf{\Phi}_{m+1}(n) \mathbf{L}_m^H(n) \quad (3.193)$$

Se puede demostrar que la matriz  $\mathbf{D}_{m+1}(n)$  es diagonal y los elementos de su diagonal principal son iguales a la suma ponderada de los errores a posteriori de predicción hacia atrás:  $\mathcal{B}_0(n)$ ,  $\mathcal{B}_1(n)$ ,  $\dots$ ,  $\mathcal{B}_m(n)$  [Alc86, Hay91]; esto es:

$$\begin{aligned} \mathbf{D}_{m+1}(n) &= \mathbf{L}_m(n) \mathbf{\Phi}_{m+1}(n) \mathbf{L}_m^H(n) \\ &= \text{diag}[\mathcal{B}_0(n), \mathcal{B}_1(n), \dots, \mathcal{B}_m(n)] \end{aligned} \quad (3.194)$$

Esta última ecuación establece que los errores a posteriori de predicción hacia atrás  $b_0(n), b_1(n), \dots, b_m(n)$  producidos por las diferentes etapas de un filtro predictor de estructura "lattice" están decorrelacionados en todos los instantes de tiempo (asumimos que el proceso es ergódico).

Cabe hacer notar que la transformación que lleva a cabo el algoritmo LSL sobre la secuencia correlacionada de datos de entrada a la secuencia no correlacionada de errores a posteriori de predicción hacia atrás puede interpretarse como una forma determinística del algoritmo Gram=Schmidt (ver apéndice A).

**Transformación de la Solución de Mínimos Cuadrados Recursivos** Considere el filtro transversal que se muestra en la figura 3.9: donde las muestras de entrada  $u(n), u(n-1), \dots, u(n-m)$  son derivadas del proceso  $\{u(n)\}$  y los coeficientes del filtro se denotan por:  $\hat{w}_0(n), \hat{w}_1(n), \dots, \hat{w}_m(n)$ . Recuerdese que la solución al problema de los mínimos cuadrados está dada por:

$$\Phi_{m+1}(n)\hat{\mathbf{w}}_m(n) = \theta_{m+1}(n) \quad (3.195)$$

donde  $\Phi_{m+1}(n)$  es la matriz de correlación del vector de muestras de entrada  $\mathbf{u}_{m+1}(n)$  y  $\theta_{m+1}(n)$  es el vector, de  $(m+1) \times 1$ , de la correlación cruzada del vector de muestras de entrada al filtro  $\mathbf{u}_{m+1}(n)$  y la respuesta deseada.

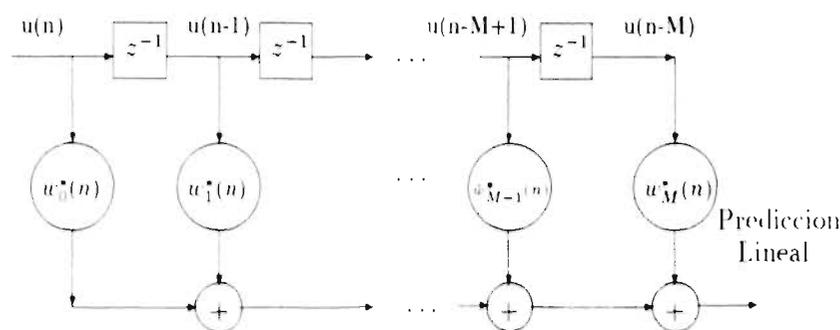


Figura 3.9: Filtro transversal.

La ecuación 3.195 se modificará mediante las dos siguientes manipulaciones algebraicas: (1) se premultiplicará por la matriz de transformación  $\mathbf{L}_m(n)$ , y (2) se intercalará la matriz identidad  $\mathbf{I} = \mathbf{L}_m^H(n)\mathbf{L}_m^{-H}(n)$  entre la matriz  $\Phi_{m+1}(n)$  y el vector  $\hat{\mathbf{w}}_m(n)$ . Efectuando estas operaciones, tenemos:

$$\mathbf{L}_m(n)\Phi_{m+1}(n)\mathbf{L}_m^H(n)\mathbf{L}_m^{-H}(n)\hat{\mathbf{w}}_m(n) = \mathbf{L}_m(n)\theta_{m+1}(n) \quad (3.196)$$

De la ecuación 3.194 sabemos que el producto matricial  $\mathbf{L}_m(n)\Phi_{m+1}(n)\mathbf{L}_m^H(n)$  es igual a la matriz diagonal  $\mathbf{D}_{m+1}(n)$ . Mientras que el producto  $\mathbf{L}_m(n)\theta_{m+1}(n)$  es

igual al vector, de  $(m+1) \times 1$ , de la correlación cruzada de los errores a posteriori de predicción hacia atrás y la respuesta deseada. Denotemos mediante  $\mathbf{t}_{m+1}(n)$  a éste vector de correlación cruzada. Entonces por definición tenemos:

$$\mathbf{t}_{m+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{b}_{m+1}(i) d^*(i) \quad (3.197)$$

donde  $d(i)$  es la respuesta deseada. Sustituyendo la ec. 3.188 en la ecuación anterior, tenemos:

$$\begin{aligned} \mathbf{t}_{m+1}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) d^*(i) \\ &= \mathbf{L}_m(n) \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{m+1}(i) d^*(i) \\ &= \mathbf{L}_m(n) \theta_{M+1}(n) \end{aligned} \quad (3.198)$$

Esta última ecuación es el resultado que buscábamos. Sustituyendo la ecuación 3.193 y la 3.198 en la 3.196, tenemos la solución transformada de los mínimos cuadrados recursivos:

$$\mathbf{D}_{m+1}(n) \mathbf{L}_m^{-H}(n) \hat{\mathbf{w}}_m(n) = \mathbf{t}_{m+1}(n) \quad (3.199)$$

Hasta aquí hemos considerado: como la aplicación de la matriz triangular inferior,  $\mathbf{L}_m(n)$ , transforma la solución de los mínimos cuadrados recursivos para el vector de coeficientes del filtro transversal que se muestra en la fig. 3.9. A continuación deseamos hallar la solución para el vector de coeficientes de regresión,  $\kappa_m(n)$  del filtro que se muestra en la fig. 3.8. El vector de coeficientes de regresión está dado por:

$$\kappa_m^T(n) = [\kappa_0, \kappa_1(n), \dots, \kappa_m(n)] \quad (3.200)$$

El vector de coeficientes de regresión se obtiene minimizando la siguiente ecuación de desempeño:

$$\sum_{i=1}^n \lambda^{n-i} |d(i) - \mathbf{b}_{m+1}^T(i) \kappa_m^*(n)|^2$$

donde  $\kappa_m(n)$  se mantiene constante durante el intervalo de observación  $1 \leq i \leq n$ . La solución a este problema de mínimos cuadrados recursivos puede expresarse como [Alc86, Hay91]:

$$\mathbf{D}_{m+1}(n) \kappa_m(n) = \mathbf{t}_{m+1}(n) \quad (3.201)$$

Al comparar la solución transformada de los mínimos cuadrados recursivos dada por la ec. 3.199 con la solución de los mínimos cuadrados recursivos dada por la ecuación anterior deducimos la siguiente relación:

$$\kappa_m(n) = \mathbf{L}_m^{-H}(n) \hat{\mathbf{w}}_m(n) \quad (3.202)$$

ó de una manera equivalente:

$$\hat{\mathbf{w}}_m(n) = \mathbf{L}_m^H(n) \kappa_m(n) \quad (3.203)$$

Nuevamente, en estas dos últimas ecuaciones la matriz de transformación triangular,  $\mathbf{L}_m(n)$ , representa la conexión entre las soluciones al problema de los mínimos cuadrados para los vectores de coeficientes de los filtros representados en las figuras 3.8 y 3.9 [Alc86, Hay91].

**Recursión para el Vector de Coeficientes de Regresión  $\kappa_m(n)$**  Resolviendo las ecuaciones 3.198 y la 3.101 en términos del vector de coeficientes de regresión, tenemos [Alc86, Hay91]:

$$\kappa_m(n) = \mathbf{D}_{m+1}^{-1}(n)\mathbf{L}_m(n)\theta_{m+1}(n) \quad (3.204)$$

Dado que la matriz  $\mathbf{D}_{m+1}(n)$  es diagonal, su inversa también lo es [Wat91]:

$$\mathbf{D}_{m+1}^{-1}(n) = \text{diag} [\mathcal{B}_0^{-1}(n), \mathcal{B}_1^{-1}(n), \dots, \mathcal{B}_m^{-1}(n)] \quad (3.205)$$

De aquí que, sólo  $m+1$  divisiones por un escalar sean requeridas en el cálculo del vector de coeficientes de regresión,  $\kappa_m(n)$ . En particular, la sustitución de las ecuaciones 3.189 y 3.205 en 3.204 nos da como resultado un sistema de  $M+1$  ecuaciones escalares:

$$\kappa_m(n) = \mathcal{B}_m^{-1}(n)\mathbf{c}_m^H(n)\theta_{m+1}(n), \quad m = 0, 1, \dots, M \quad (3.206)$$

donde  $M$  es el valor final del orden de predicción. Para desarrollar una recursión en el tiempo para el coeficiente  $\kappa_m(n)$ , definimos el escalar:

$$\rho_m(n) = \mathbf{c}_m^H(n)\theta_{m+1}(n), \quad m = 0, 1, \dots, M \quad (3.207)$$

Podemos redefinir el coeficiente de regresión dado por la ec. 3.206 en términos de  $\rho_m(n)$  como sigue:

$$\kappa_m(n) = \frac{\rho_m(n)}{\mathcal{B}_m(n)}, \quad m = 0, 1, \dots, M \quad (3.208)$$

Recordemos la siguiente recursión en el tiempo para el vector de coeficientes del filtro de predicción hacia atrás del error:

$$\mathbf{c}_m(n) = \mathbf{c}_m(n-1) - \frac{b_m^*(n)}{\gamma_m(n)} \begin{bmatrix} \mathbf{k}_m(n) \\ 0 \end{bmatrix} \quad (3.209)$$

donde  $\mathbf{k}_m(n)$  es el vector de ganancia (de  $m \times 1$ ). Además recuérdese que la recursión en el tiempo para el vector de correlación cruzada  $\theta_{m+1}(n)$  esta dada por:

$$\theta_{m+1}(n) = \lambda\theta_{m+1}(n-1) + \mathbf{u}_{m+1}(n)d^*(n) \quad (3.210)$$

Sustituyendo las ecuaciones 3.209 y 3.210 en 3.207, y utilizando el hecho de que  $\gamma_m(n)$  es real, obtenemos:

$$\begin{aligned} \rho_m(n) &= \lambda\mathbf{c}_m^H(n-1)\theta_{m+1}(n-1) + \mathbf{c}_m^H(n-1)\mathbf{u}_{m+1}(n)d^*(n) \\ &= -\frac{b_m^*(n)}{\gamma_m(n)}\mathbf{k}_m^H(n)\theta_m(n) \end{aligned} \quad (3.211)$$

En el último término de la ecuación anterior, se usó el hecho de que los primeros  $m$  elementos del vector  $\theta_{m+1}(n)$  son iguales a los elementos del vector  $\theta_m(n)$ , y dado que el último elemento del vector de dimensión  $(m+1) \times 1$  en el lado derecho de la ec. 3.209 es cero; sólo los primeros  $m$  elementos de  $\theta_{m+1}(n)$  entran en la multiplicación del término  $\mathbf{c}_m^H(n)$  por  $\theta_{m+1}(n)$ . La ec. 3.211 se puede expresar como sigue:

- De la ecuación 3.207, se deduce que [Alc86, Hay91]:

$$\rho_m(n-1) = \mathbf{c}_m^H(n-1)\theta_{m+1}(n-1)$$

- De la definición del error a priori de predicción hacia atrás, tenemos:

$$v_m(n) = \mathbf{c}_m^H(n-1)\mathbf{u}_{m+1}(n)$$

- Además, dado que  $v_m(n)$  es igual a la razón de  $b_m(n)$  a  $\gamma_m(n)$ , tenemos que:

$$\mathbf{c}_m^H(n-1)\mathbf{u}_{m+1}(n) = \frac{b_m(n)}{\gamma_m(n)}$$

- Dado que el vector de ganancia  $\mathbf{k}_m(n)$  es igual a  $\Phi_m^{-1}(n)\mathbf{u}_m(n)$  y, a que la matriz  $\Phi_m(n)$  es Hermiteana, tenemos que [Alc86, Hay91]:

$$\begin{aligned} \mathbf{k}_m^H(n)\theta_m(n) &= \mathbf{u}_m^H(n)\Phi_m^{-1}(n)\theta_m(n) \\ &= \mathbf{u}_m^H(n)\hat{\mathbf{w}}_{m-1}(n) \end{aligned} \quad (3.212)$$

donde en la última línea de la ecuación anterior se han usado las ecuaciones normales para la estimación por mínimos cuadrados. El producto interno  $\mathbf{u}_m^H(n)\hat{\mathbf{w}}_{m-1}(n)$  es igual al complejo conjugado de la estimación por mínimos cuadrados de la respuesta deseada  $d(n)$ . Sea  $\hat{d}(n | \mathcal{U}_n)$  el valor de la estimación por mínimos cuadrados de la respuesta deseada, donde  $\mathcal{U}_n$  es el espacio generado por los elementos del vector  $\mathbf{u}_m(n)$ .

De acuerdo a lo anterior la ec. 3.211 se puede expresar como:

$$\rho_m(n) = \lambda\rho_m(n-1) + \frac{b_m(n)}{\gamma_m(n)} [d^*(n) - \hat{d}^*(n | \mathcal{U}_n)] \quad (3.213)$$

Se define el error a posteriori de estimación, basándose en las  $m$  muestras de entrada, de acuerdo a:

$$\begin{aligned} \epsilon_m(n) &= d(n) - \hat{d}(n | \mathcal{U}_n) \\ &= d(n) - \hat{\mathbf{w}}_{m-1}^H(n)\mathbf{u}_m(n) \end{aligned} \quad (3.214)$$

Sustituyendo la última ecuación en 3.213, tenemos:

$$\rho_m(n) = \lambda \rho_m(n-1) + \frac{b_m(n)}{\gamma_m(n)} \epsilon_m^*(n), \quad m = 0, 1, 2, \dots, M \quad (3.215)$$

Excepto por  $\gamma_m(n)$ , podemos interpretar a  $\rho_m$  como la correlación cruzada de  $b_m(n)$  y  $\epsilon_m(n)$ .

Para completar el procedimiento de recursión para el proceso de estimación conjunta, es necesario desarrollar una recursión en el orden para el error de estimación: esto se logra reemplazando  $m$  por  $m+1$  en la ec. 3.214:

$$\epsilon_{m+1}(n) = d(n) - \hat{\mathbf{w}}_m^H(n) \mathbf{u}_{m+1}(n) \quad (3.216)$$

Dado que el producto interno  $\hat{\mathbf{w}}_m^H(n) \mathbf{u}_{m+1}(n)$  es igual a  $\kappa_m^H(n) \mathbf{b}_{m+1}(n)$  (ver ecs. 3.188 y 3.203), podemos expresar a  $\epsilon_{m+1}(n)$  como sigue:

$$\begin{aligned} \epsilon_{m+1}(n) &= d(n) - \kappa_m^H(n) \mathbf{b}_{m+1}(n) \\ &= d(n) - \sum_{l=0}^m \kappa_l^*(n) b_l(n) \end{aligned} \quad (3.217)$$

Aislando el producto interno  $\kappa_m^*(n) b_m(n)$  del resto de los términos de la suma del lado derecho de la ecuación anterior y, reconociendo que la suma es igual a:

$$d(n) - \sum_{l=0}^{m-1} \kappa_l^*(n) b_l(n) = \epsilon_m(n) \quad (3.218)$$

entonces podemos expresar la recursión en el orden para el error a posteriori de estimación como:

$$\epsilon_{m+1}(n) = \epsilon_m(n) - \kappa_m^*(n) b_m(n), \quad m = 0, 1, \dots, M \quad (3.219)$$

Nótese que el número de coeficientes de regresión excede el orden final de predicción,  $M$ , por uno.

### 3.3.3.9 Organigrama del Algoritmo LSL

<p>Predicción:</p> <p>Para <math>n = 1, 2, 3, \dots</math> calcule las sig. recursiones en el orden:</p>
$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + \frac{b_{m-1}(n-1) f_{m-1}^*(n)}{\gamma_{m-1}(n-1)}$ $\Gamma_{f,m}(n) = -\frac{\Delta_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)}$ $\Gamma_{b,m}(n) = -\frac{\Delta_{m-1}^*(n)}{\mathcal{F}_{m-1}(n)}$ $f_m(n) = f_{m-1}(n) + \Gamma_{f,m}(n) b_{m-1}(n-1)$ $b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m}(n) f_{m-1}(n)$ $\mathcal{F}_m(n) = \mathcal{F}_{m-1}(n) - \frac{ \Delta_{m-1}(n) ^2}{\mathcal{B}_{m-1}(n-1)}$ $\mathcal{B}_m(n) = \mathcal{B}_{m-1}(n-1) - \frac{ \Delta_{m-1}(n) ^2}{\mathcal{F}_{m-1}(n)}$ $\gamma_m(n-1) = \gamma_{m-1}(n-1) - \frac{ b_{m-1}(n-1) ^2}{\mathcal{B}_{m-1}(n-1)}$
<p>Filtrado:</p> <p>Para <math>n = 1, 2, 3, \dots</math> calcule las sig. recursiones en el orden:</p>
$\rho_m(n) = \lambda \rho_m(n-1) + \frac{\delta_m(n)}{\gamma_m(n)} \epsilon_m^*(n)$ $\kappa_m = \frac{\rho_m(n)}{\mathcal{B}_m(n)}$ $\epsilon_{m+1}(n) = \epsilon_m(n) - \kappa_m^*(n) b_m(n)$

Tabla 3.8: Organigrama del algoritmo LSL.

<p>Para inicializar el algoritmo al tiempo <math>n = 0</math>:</p> $\Delta_{m-1}(0) = 0$ $\mathcal{F}_{m-1}(0) = \delta, \quad 0 < \delta \ll 1$ $\mathcal{B}_{m-1}(0) = \delta$ $\gamma_0(0) = 1$ $\rho_m(0) = 0$
<p>Para cada <math>n \geq 1</math>, genere los parámetros de orden cero:</p> $f_0(n) = b_0(n) = u(n)$ $\mathcal{F}_0(n) = \mathcal{B}_0(n) = \lambda \mathcal{F}_0(n-1) +  u(n) ^2$ $\gamma_0(n-1) = 1$ $e_0(n) = d(n)$

Tabla 3.9: Inicialización del algoritmo LSL.

### 3.3.3.10 Algoritmo LSL Usando Errores A Priori de Estimación

En esta sección se describe una versión del algoritmo LSL que usa formas a priori de los errores de predicción hacia adelante, de los errores de predicción hacia atrás, y de los errores de estimación del proceso de estimación conjunto, como los parámetros de interés.

Recordemos que el error a posteriori de predicción hacia adelante,  $f_m(n)$ , y el error a priori de predicción hacia adelante,  $\eta_m(n)$ , están relacionados mediante (ver ec. 3.97):

$$f_m(n) = \gamma_m(n-1)\eta_m(n) \quad (3.220)$$

Recuérdese, también que la relación entre el error a posteriori de predicción hacia atrás  $b_m(n)$  y el error a priori de predicción hacia atrás  $\psi_m(n)$  esta dada por (ver

ec. 3.98):

$$b_m(n) = \gamma_m(n)v_m(n) \quad (3.221)$$

De la relación entre el error a posteriori de estimación,  $\epsilon_m(n)$ , y el error a priori de estimación  $\alpha_m(n)$ , tenemos (ver ec. 3.96):

$$\epsilon_m(n) = \gamma_m(n)\alpha_m(n) \quad (3.222)$$

También recuérdese la relaciones que existen para actualizar en el tiempo la suma ponderada del cuadrado de los errores tanto de predicción hacia adelante como de predicción hacia atrás (ver ecs. 3.72 y 3.88):

$$\mathcal{F}_{m-1}(n) = \lambda\mathcal{F}_{m-1}(n-1) + \eta_{m-1}(n)f_{m-1}^*(n) \quad (3.223)$$

y:

$$\mathcal{B}_{m-1}(n) = \lambda\mathcal{B}_{m-1}(n-1) + v_{m-1}(n)b_{m-1}^*(n) \quad (3.224)$$

Al usar las ecuaciones 3.153, 3.181, 3.220, 3.221 y 3.224 en 3.162, se obtiene la siguiente recursión en el orden para el error a priori de predicción hacia adelante:

$$\eta_m(n) = \eta_{m-1}(n) + \Gamma_{f,m}^*(n-1)v_{m-1}(n-1), \quad m = 1, 2, \dots, M \quad (3.225)$$

Similarmente utilizando las ecs. 3.154, 3.181, 3.220, 3.221 y 3.223 en 3.164, se obtiene la siguiente recursión en el orden para el error a priori de predicción hacia atrás:

$$v_m(n) = v_{m-1}(n-1) + \Gamma_{v,m}^*(n-1)\eta_{m-1}(n), \quad m = 1, 2, \dots, M \quad (3.226)$$

Finalmente usando las ecuaciones 3.208, 3.215, 3.221, 3.222 y 3.224 en 3.217, se obtiene la siguiente recursión en el orden para el error a priori de estimación:

$$\alpha_{m+1}(n) = \alpha_m(n) - \kappa_m^*(n-1)v_m(n), \quad m = 0, 1, 2, \dots, M \quad (3.227)$$

Combinando las recursiones anteriores en el orden con las siguientes recursiones: (1) recursión en el tiempo para correlación cruzada  $\Delta_m(n)$ , en términos de los errores a priori de predicción  $\eta_{m-1}(n)$  y  $v_{m-1}(n-1)$ ; (2) recursiones en el tiempo para las sumas ponderadas del cuadrado de los errores a posteriori de predicción  $\mathcal{F}_{m-1}(n)$  y  $\mathcal{B}_{m-1}(n)$  y; (3) recursiones en el tiempo para la correlación cruzada  $\rho_m(n)$  reformulada en términos de los errores a priori  $\alpha_m(n)$  y  $v_m(n)$ ; obtenemos la segunda versión del algoritmo LSL [Hay91, LMP85, LS83].

### 3.3.3.11 Organigrama del Algoritmo LSL Usando Errores A Priori de Estimación

<p>Predicción:</p> <p>Para <math>n = 1, 2, 3, \dots</math> calcule las sig. recursiones en el orden:</p>
$\eta_m(n) = \eta_{m-1}(n) + \Gamma_{f,m}^*(n-1)\psi_{m-1}(n-1)$ $\psi_m(n) = \psi_{m-1}(n-1) + \Gamma_{b,m}^*(n-1)\eta_{m-1}(n)$ $\Delta_{m-1}(n) = \lambda\Delta_{m-1}(n-1) + \gamma_{m-1}(n-1)\psi_{m-1}(n-1)\eta_{m-1}^*(n)$ $\mathcal{F}_{m-1}(n) = \lambda\mathcal{F}_{m-1}(n-1) + \gamma_{m-1}(n-1) \eta_{m-1}(n) ^2$ $\mathcal{B}_{m-1}(n) = \lambda\mathcal{B}_{m-1}(n-1) + \gamma_{m-1}(n) \psi_{m-1}(n) ^2$ $\Gamma_{f,m}(n) = -\frac{\Delta_{m-1}(n)}{\mathcal{B}_{m-1}(n-1)}$ $\Gamma_{b,m}(n) = -\frac{\Delta_{m-1}^*(n)}{\mathcal{F}_{m-1}(n)}$ $\gamma_m(n) = \gamma_{m-1}(n) - \frac{\gamma_{m-1}^*(n) \psi_{m-1}(n) ^2}{\mathcal{B}_{m-1}(n)}$
<p>Filtrado:</p> <p>Para <math>n = 1, 2, 3, \dots</math> calcule las sig. recursiones en el orden:</p>
$\rho_m(n) = \lambda\rho_m(n-1) + \gamma_m(n)\psi_m(n)\alpha_m^*(n)$ $\alpha_{m+1}(n) = \alpha_m(n) - \kappa_m^*(n-1)\psi_m(n)$ $\mathcal{B}_m(n) = \lambda\mathcal{B}_m(n) + \gamma_m(n) \psi_m(n) ^2$ $\kappa_m(n) = \frac{\rho_m(n)}{\mathcal{B}_m(n)}$

Tabla 3.10: Organigrama del algoritmo LSL usando errores a priori de estimación.

<p>Para inicializar el algoritmo al tiempo <math>n = 0</math>:</p> $\Delta_{m-1}(0) = 0$ $\mathcal{F}_{m-1}(0) = \delta, \quad 0 < \delta \ll 1$ $\mathcal{B}_{m-1}(0) = \delta$ $\Gamma_{f,m}(0) = \Gamma_{b,m}(0) = 0$ $\gamma_0(0) = 1$ $\rho_m(0) = 0$
<p>Para cada <math>n \geq 1</math>, genere los parámetros de orden cero:</p> $\eta_0(n) = \epsilon_0(n) = u(n)$ $\mathcal{F}_0(n) = \mathcal{B}_0(n) = \lambda \mathcal{F}_0(n-1) +  u(n) ^2$ $\gamma_0(n) = 1$ $\alpha_0(n) = d(n)$

Tabla 3.11: Inicialización del algoritmo LSL usando errores a priori de estimación.

### 3.4 Conclusiones

En este capítulo se planteo el problema de los mínimos cuadrados cuya solución directa requiere de la inversión de la matriz de autocorrelación de los datos de entrada, lo cual no es conveniente computacionalmente debido al elevado número de operaciones necesarias para invertir esta matriz. De aquí que se halla buscado la manera de resolver este problema de forma recursiva dando origen al algoritmo recursivo de los mínimos cuadrados.

A pesar de que el algoritmo recursivo de los mínimos cuadrados requiere de un número menor de operaciones que la inversión directa de la matriz de autocorrelación de los datos de entrada al receptor: se ha demostrado que existe la manera de realizar menos operaciones dando de esta manera paso a los algoritmos rápidos. Como se recordará se dice que un algoritmo es rápido si al aumentar el número de coeficientes a ajustar, el número de operaciones a realizar aumenta de una manera lineal. De aquí la gran importancia de estos algoritmos para igualar canales en los que se requiere de un número considerable de coeficientes para lograr un resultado satisfactorio. A pesar de las bondades de estos algoritmos, no son la panacea universal para el problema del filtrado adaptable pues como se verá en el siguiente capítulo si el igualador ideal es un filtro IIR que sea estable, entonces los algoritmos de filtrado adaptable que adaptan los coeficientes de un filtro IIR se desempeñan mucho mejor, ya que logran converger en un número menor de iteraciones que los algoritmos aquí tratados. Además dichos algoritmos requeriran de un número menor de coeficientes a ajustar que si se emplea un filtro FIR, el cual requerirá de un número muy elevado de coeficientes para aproximar la respuesta impulsional infinita del igualador ideal.

## Capítulo 4

# Algoritmos Adaptables Para Filtros A Respuesta Impulsional Infinita

Los filtros adaptables a respuesta impulsional finita (FIR) han gozado de una gran aceptación dado que la formulación matemática del problema del filtrado adaptable en terminos de estos filtros es sencilla y, además da como resultado que la superficie del error cuadrático medio sea igual a un paraboloide [Hay91, WS85]; por lo tanto, la minimización del error cuadrático medio siempre dará el mismo resultado sin importar el valor que se le asigne como condición inicial al vector de coeficientes del filtro. De aquí proviene la gran aceptación de los algoritmos de filtrado adaptable para filtros a respuesta impulsional finita, sin embargo existen casos en los cuales la aplicación de un filtro adaptable a respuesta impulsional finita no resuelve el problema de una manera satisfactoria [EJ86, CRJ81, ETCRJ80], esto se debe principalmente a que el filtro adaptable a respuesta impulsional finita requiere de un gran número de coeficientes para poder llevar a cabo su tarea de una manera eficiente en el sentido de converger a la solución óptima mas no en un tiempo relativamente corto, lo cual se debe al excesivo número de operaciones resultado del gran número de coeficientes requeridos por el algoritmo de filtrado adaptable. El requerimiento de un gran número de coeficientes para producir una solución óptima con respecto a algún criterio de minimización es resultado de que el fenómeno bajo estudio no se puede modelar de una manera conveniente mediante un modelo todo cero (también llamado de promedio móvil) sino que se requiere en general de un modelo que contemple polos y ceros; lo cuál, como se verá más adelante, además de complicar el desarrollo matemático en lo que respecta a la minimización del criterio del error, ocasionará problemas de inestabilidad en lo que se refiere a las actualizaciones del vector de coeficientes que dan por resultado un filtro inestable; aunado a todos estos problemas, la superficie descrita por el criterio del error no posee un sólo mínimo absoluto como en el caso de los filtros adaptables a respuesta impulsional finita sino que dicha superficie poseerá varios mínimos en general, por lo cuál el

algoritmo no siempre convergerá a la solución óptima sino a alguna otra solución, la cual no siempre será la adecuada para el problema que se esté resolviendo. El hecho de que la superficie que describe el criterio del error tenga varios mínimos implica que la elección de las condiciones iniciales juega un papel crítico en la convergencia del algoritmo a la solución óptima.

A pesar de todos estos problemas, la necesidad de desarrollar algoritmos adaptables para filtros a respuesta impulsional infinita tiene dos motivaciones principales, la primera es el deseo de reducir el número de iteraciones necesarias para converger a la solución óptima en aquellos casos en que la aplicación de un filtro de respuesta impulsional finita sea impráctica debido al gran número de coeficientes requeridos para obtener la solución óptima de acuerdo a algún criterio de minimización del error y, la segunda proviene del deseo de identificar o ajustar un modelo a un sistema desconocido, el cual generalmente y debido a su naturaleza será mejor representado por un modelo que contemple la posibilidad de la coexistencia de polos y ceros (llamado autoregresivo y de promedio móvil ARMA ).

## 4.1 Antecedentes Históricos

Los primeros trabajos sobre algoritmos adaptables para filtros a respuesta impulsional infinita se centran en la idea de extrapolar el algoritmo LMS propuesto por Widrow para el caso de los filtros adaptables a respuesta impulsional infinita basándose en las técnicas del gradiente estocástico [CRJL77, PA78, SEA76, Whi75, WM77]. Sin embargo, como se verá más adelante estos algoritmos sufren de ciertas limitaciones lo que restringe su uso práctico [FJ86].

Después de estos intentos, se decidió dar un giro a la dirección de los esfuerzos por formular algoritmos de filtrado para filtros a respuesta impulsional infinita con propiedades de convergencia en principio susceptibles de demostrar matemáticamente. Este nuevo giro consistió en reformular el problema del filtrado adaptable para filtros a respuesta impulsional infinita en términos de un problema más estudiado y comprendido hasta aquél entonces, que es el problema de la identificación de sistemas [GS84, LS83]. Este cambio de enfoque abrió nuevas perspectivas desde las cuales atacar el problema del filtrado adaptable para filtros a respuesta impulsional infinita, un ejemplo de ello lo constituyen una familia de algoritmos basados en las ideas de la hiperestabilidad [JT79, CRJ78, CRJ84, CRJT80, CRJLTD81, PSA79, LTCRJ80]. Entre éstos algoritmos se ha demostrado que el "hyperstable adaptive recursive filter" (HARF) es asintóticamente convergente bajo la condición de "strict positive reality" (SPR) [LTCRJ80]. Esta última condición constituye un obstáculo mayor en la aplicación práctica del algoritmo HARF [CRJ84, CRJT80, CRJLTD81, PSA79].

## 4.2 Formulación del Problema del Filtrado Adaptable IIR

Fundamentalmente existen dos vertientes para proponer soluciones al problema del filtrado adaptable, estas son: la formulación de la ecuación de error y la formulación del error de salida. Estos métodos tienen su origen en dos interpretaciones diferentes del error de predicción como veremos más adelante.

A pesar de los esfuerzos realizados, ambos métodos presentan problemas que impiden su aceptación universal tal y como lo han logrado los algoritmos FIR de filtrado adaptable. Estos problemas son: 1) el método de la ecuación de error puede dar como resultado estimaciones sesgadas del vector de coeficientes que caracteriza al filtro adaptable [Shy89]; 2) el método del error de la salida no presenta el problema anterior, sin embargo dado que es un algoritmo recursivo sobre la salida si presenta problemas de estabilidad sobre todo si los polos están cercanos al círculo unitario; además puede converger a un mínimo local en lugar de hacerlo al mínimo global y, finalmente es difícil predecir su convergencia [Shy89].

### 4.2.1 Método de la Ecuación de Error

Considere el filtro adaptivo caracterizado por la siguiente ecuación:

$$y_e(n) = \sum_{m=1}^{N-1} a_m(n)d(n-m) + \sum_{m=0}^{M-1} b_m(n)x(n-m) \quad (4.1)$$

Donde  $a_m$  y  $b_m$  son los coeficientes del filtro adaptivo,  $x$  es la señal de entrada,  $d$  es la respuesta deseada y  $y_e$  es la salida del filtro adaptivo para el método de la ecuación del error. Obsérvese que la ec. 4.1 representa un filtro de dos entradas y una salida, la cuál depende de las muestras de la señal de entrada  $x(n-m)$ ,  $m = 0, \dots, M-1$ , y de las muestras de la respuesta deseada  $d(n-m)$ ,  $m = 0, \dots, N-1$ . Dado que el filtro no es recursivo sobre la salida, éste no presenta retroalimentación, la salida es una función lineal de los coeficientes del filtro. Esta propiedad simplifica la formulación de los algoritmos de filtrado adaptable basados en el gradiente estocástico, ya que la salida del filtro  $y_e(n)$  no depende de sus valores pasados, los cuales a su vez dependen de los valores pasados de los parámetros del filtro. En cambio  $y_e$  sólo depende de los parámetros del filtro al igual que  $x(n)$  y  $d(n)$ .

La ecuación (4.1) se puede escribir de un modo más conveniente mediante:

$$y_e(n) = A(n, q)d(n) + B(n, q)x(n) \quad (4.2)$$

donde los polinomios en  $q$  representan filtros variantes en el tiempo:

$$A(n, q) = \sum_{m=1}^{N-1} a_m(n)q^{-m} \quad y \quad B(n, q) = \sum_{m=0}^{M-1} b_m(n)q^{-m} \quad (4.3)$$

donde  $q$  representa la operación de atrasar una señal en el tiempo, es decir si  $q^{-m}x(n) = x(n - m)$ .

Obsérvese que la salida del filtro  $y_\epsilon$  sólo depende de las muestras anteriores de  $d$ , lo cual nos revela la idea básica del método de la ecuación de error, la cuál consiste en estimar la muestra presente de la respuesta deseada a partir de sus muestras pasadas y de las muestras de la señal de entrada. Esencialmente opera como un filtro FIR de dos entradas y una salida de tal manera que el polinomio asociado a los polos del filtro IIR  $1 - A(q)$ , se adapta como si fuese un filtro FIR. Después de cada actualización de los coeficientes del filtro, el inverso del polinomio  $1 - A(q)$  es copiado al filtro todo polar, el cuál a su vez esta conectado en cascada con  $B(n, q)$ . De esta manera se logra adaptar el filtro IIR como si se tratase de un filtro FIR, sin embargo obsérvese que si los ceros del polinomio caen fuera del círculo unitario entonces el filtro IIR es inestable, por lo que éste método no resuelve el problema de las actualizaciones inestables. Así de esta manera cuando los ceros de  $1 - A(q)$  caen fuera del círculo unitario se requerirá de un método para proyectarlos dentro de éste último.

La ecuación de error está dada por:

$$\epsilon_\epsilon(n) = d(n) - y_\epsilon(n) \quad (4.4)$$

Se le llama ecuación de error porque es generada al substraer dos ecuaciones de diferencias:  $[1 - A(q)]d(n)$  y  $B(n, q)x(n)$ . Nótese que  $\epsilon_\epsilon$  también es una función lineal de los coeficientes del filtro; por lo tanto el error cuadrático para el método de la ecuación de error (MSEE) es una función cuadrática con un sólo mínimo.

Obsérvese que la ecuación (4.4) puede escribirse como:

$$y_\epsilon(n) = \theta^T(n)\phi_\epsilon(n) \quad (4.5)$$

donde  $\theta$  es el vector de coeficientes, y el vector de señales esta dado por  $\phi_\epsilon$ , cada uno de longitud  $M + N - 1$ , es decir:

$$\begin{aligned} \theta(n) &= [a_1(n), a_2(n), \dots, a_{N-1}(n), b_0(n), \dots, b_{M-1}(n)]^T \\ \phi_\epsilon(n) &= [d(n-1), \dots, d(n-N+1), x(n), \dots, x(n-M+1)]^T \end{aligned} \quad (4.6)$$

Obsérvese que la ecuación 4.5 tiene la forma de una regresión lineal, la cual es comúnmente utilizada en estadística [MT77], donde  $\theta$  corresponde a los parámetros estimados y  $\phi_\epsilon$  es el vector de regresión (vector de datos). El regresor es independiente de los coeficientes dado que  $d(n)$  y  $x(n)$  no son funciones de  $A(n, q)$  y de  $B(n, q)$ . Muchos de los algoritmos y técnicas utilizadas en la inferencia estadística de parámetros pueden ser utilizados para encontrar el conjunto de parámetros óptimos. Algunos de estos métodos son: el algoritmo de verosimilitud máxima [Men87], máximo a posteriori [Men87], mínimos cuadrados [Hay91] y error cuadrático medio [WS85].

## 4.2.2 Método del Error de Salida

El método del error de salida está caracterizado por la siguiente ecuación cuando  $v(n) = 0$  (ver fig. 4.1):

$$y_o(n) = \sum_{m=1}^{N-1} a_m(n)y_o(n-m) + \sum_{m=0}^{M-1} b_m(n)x(n-m) \quad (4.7)$$

la cual depende en las muestras pasadas de  $y_o$ , las cuales a su vez dependen de los valores pasados de los parámetros del filtro, lo cual hace muy complicada la obtención del gradiente del error como veremos más adelante. Esta última ecuación nos indica que el problema aquí tratado es reformulado restringiendo la generalidad del mismo: pues anteriormente la señal deseada  $y(n)$  era cualquier señal, mientras que en esta nueva reformulación del problema se considerará a la señal deseada como la salida de una planta, la cual se modela mediante un proceso autoregresivo y de promedio móvil (ARMA). Obsérvese que esta "restricción" no es de ninguna manera excesiva ya que el modelo más general para una señal es el modelo ARMA y, dado que estamos en libertad de fijar tanto el número de polos como el número de ceros se considera que podemos generar cualquier señal. Además si hemos de considerar señales producidas tanto por la naturaleza como por el hombre no tenemos ningún motivo de preocupación por la restricción impuesta por este nuevo enfoque del problema tratado, pues toda señal de este tipo puede ser modelada por un proceso ARMA. La ecuación 4.7 se puede escribir en términos de la ecuación 4.3:

$$y_o(n) = \left( \frac{B(n, q)}{1 - A(n, q)} \right) x(n) \quad (4.8)$$

También la ecuación 4.7 se puede escribir de una manera compacta como:

$$y_o(n) = \theta^T(n)\phi(n) \quad (4.9)$$

donde el vector de coeficientes  $\theta$  está dado por la ecuación (4.6) pero el vector de señales está dado por:

$$\phi(n) = [y_o(n-1), \dots, y_o(n-N+1), x(n), \dots, x(n-M+1)]^T \quad (4.10)$$

La ecuación 4.9 no es una regresión lineal pero dado que tiene la forma de la ec. 4.5 se le conoce como regresión pseudolineal [LS83]. Se pueden aplicar técnicas similares al caso de la regresión lineal pero la solución obtenida no será la óptima a menos que se cumpla con la condición SPR [Lau79], la cual abordaremos posteriormente en este mismo capítulo.

El error de salida está dado por  $\epsilon_o(n) = d(n) - y_o(n)$ : recibe éste nombre debido a que se genera al restar  $d(n)$  de la ecuación 4.8. Claramente,  $\epsilon_o(n)$  es una función no lineal de  $\theta$ , por lo tanto, el error cuadrático medio del método del error de salida (MSEOE) puede tener varios mínimos locales ya que no es una función cuadrática [Ste81]. Matemáticamente  $\epsilon_o$  se puede expresar como (ver fig. 4.1):

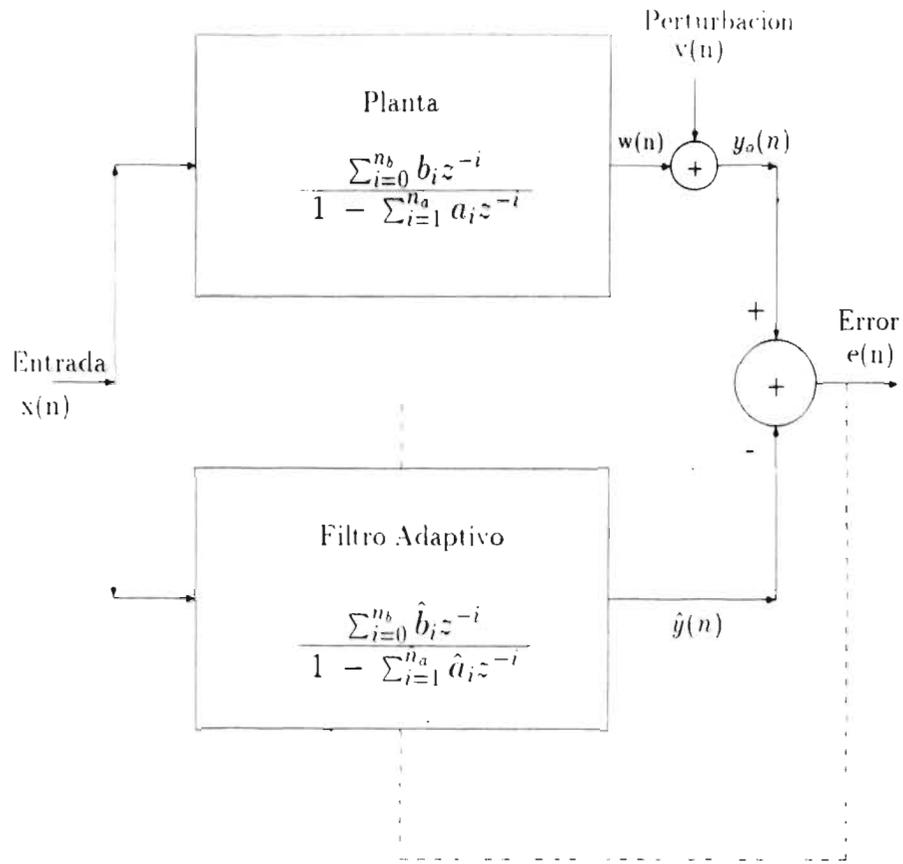


Figura 4.1: Método del Error de Salida.

$$\epsilon_o(n) \equiv \sum_{i=0}^{\hat{n}_b} \{b_i - \hat{b}_i(n)\} x(n-i) + \sum_{i=1}^{\hat{n}_a} \{a_i y_o(n-i) - \hat{a}_i(n) \hat{y}_o(n-i)\} + v(n) \quad (4.11)$$

dónde  $\hat{n}_a \geq n_a$  y  $\hat{n}_b \geq n_b$ . De aquí en adelante se asumirá sin pérdida de generalidad que  $\hat{n}_a = n_a$  y que  $\hat{n}_b = n_b$ . Para minimizar la función de costo:

$$J = E\{\epsilon_o^2(n)\} \quad (4.12)$$

basta con hacer  $\hat{b}_i \equiv b_i$  y  $\hat{a}_i \equiv a_i$ , de esta manera tenemos:

$$\epsilon_o(n) \equiv \sum_{i=1}^N a_i \epsilon(n-i) + v(n) \quad (4.13)$$

debido a la estabilidad de (4.11) tenemos:

$$\lim_{n \rightarrow \infty} \epsilon_o(n) = v(n) \quad (4.14)$$

Antes de proseguir cabe hacer una distinción entre el problema de la identificación de sistemas y el problema aquí tratado: en el primer caso el error resultante de la diferencia de la señal deseada de su respectiva estimada, se utiliza sólo como un medio para lograr un error pequeño de los parámetros estimados con respecto a los parámetros "reales"; mientras que en el problema del filtrado el fin es lograr que el error en cuestión sea lo más cercanamente posible a cero cuando  $v(n) = 0$ , ó en caso contrario que  $e_c(n)$  sea igual a  $v(n)$ [LTCRJ80].

#### 4.2.2.1 Superficies de Error para el Método del Error de Salida

Una vez que hemos establecido el método del error de salida para el problema del filtrado adaptable para filtros a respuesta impulsional infinita (IIR) podemos referirnos al estudio de las superficies de error para dicha modalidad del problema aquí tratado.

Recordemos de los capítulos anteriores que las funciones de costo para el caso del filtrado adaptable para filtros a respuesta impulsional finita (FIR) son unimodales con respecto al vector de coeficientes del filtro  $\theta$  y, además están representadas por hiperparaboloides de dimensión  $n$ . Por tanto resulta lógico la utilización de la función gradiente para hallar el valor óptimo del vector de coeficientes del filtro, el cual está dado por el mínimo de la función de costo. Sin embargo para el presente problema estas superficies carecen de estas dos valiosas características que contribuirían enormemente a la simplificación del problema del filtrado adaptable utilizando filtros a respuesta impulsional finita, por tanto la solución del presente problema se complica de sobre manera si consideramos que la superficie que representa a la función de costo tiene varios mínimos. Esto implica que el algoritmo puede converger a una solución que no es la óptima; por ello se han hecho varios esfuerzos por estudiar el comportamiento de las superficies de error, entre estos estudios uno de los más destacados es el realizado por Stearns [Ste81]. En este estudio se analiza el comportamiento de las superficies de error mediante simulaciones por computadora, es decir, de una forma numérica más no analítica. Stearns llega a la siguiente conclusión: "si el filtro adaptable es de suficiente orden (esto es, si  $\hat{n}_1 \geq n_1$  y  $\hat{n}_k \geq n_k$  véase la fig. 4.1) y además, si la señal de entrada al filtro  $x(n)$  es ruido blanco entonces la superficie del error dada por el valor esperado de  $e^2(n)$  es unimodal. Tomando como base esta conclusión, las superficies de error para el método del error de salida de los filtros adaptables a respuesta impulsional infinita que trabajan en un ambiente estacionario se pueden clasificar aproximadamente de la siguiente manera:

- 1) de suficiente orden con ruido blanco como excitación.
- 2) de suficiente orden con ruido coloreado como excitación.
- 3) de orden reducido con ruido blanco como excitación.
- 4) de orden reducido con ruido coloreado como excitación.

Las superficies de error para el caso 1 son unimodales de acuerdo con Stearns [Ste81]. En lo que respecta a los tres casos restantes, las superficies de error son multimodales

en general. A continuación se citan algunas fuentes donde se muestra la multimodalidad de las superficies de error: la multimodalidad para el caso dos queda mostrada en [S75] y para el caso tres en [CRJL77, Ste81]. Finalmente la multimodalidad para el caso cuatro se puede demostrar generalizando el ejemplo que se da en [S75] como se muestra en [FJ86]. Por último cabe hacer mención de que apesar de los estudios realizados, se tiene un conocimiento bastante limitado de las superficies de error [FJ86].

### 4.2.3 Problemas Relativos a los Métodos de la Ecuación de Error y el Error de Salida

Como se mencionó anteriormente los algoritmos basados en el método del error pueden converger a un resultado, el cual sufre de un sesgo con respecto a la solución óptima. Dentro del contexto de la identificación de sistemas, este problema se manifiesta en estimaciones incorrectas del vector de parámetros  $\theta$ , es decir el límite de  $E\{\theta(n)\}$  cuando  $n$  tiende a infinito es igual a  $\theta + \text{sesgo}$ . Se puede demostrar que este sesgo es igual a cero si cualquiera de las dos siguientes condiciones se cumple: que no exista ruido o bien que  $A(n, q) = 0$  (lo que corresponde a un filtro FIR). Nótese que la ecuación de error puede ser expresada como una versión filtrada del error de salida [Shy89]:  $e_e(n) = [1 - A(n, q)]\epsilon_o(n)$ .

Nótese que la magnitud del sesgo es directamente proporcional a la potencia del ruido dado que éste también es filtrado por  $1 - A(n, q)$ . En efecto, el sesgo se debe a que el filtro adaptivo intenta minimizar la potencia del ruido que afecta a  $e_e(n)$ , mientras que por otro lado trata de identificar los polos del sistema. A pesar de que los métodos basados en la ecuación del error convergen rápidamente, su desempeño puede ser completamente insatisfactorio si el sesgo es significativo.

Los algoritmos basados en el método del error de salida tienen una complejidad mayor que los algoritmos basados en el método de la ecuación del error pero no presentan el problema de la solución sesgada. Sin embargo, pueden converger a un mínimo local de la superficie de error (MSOE) [Ste81]. Es por ello, que se han realizado estudios para investigar las condiciones suficientes para la no existencia de mínimos locales en la superficie del error MSOE [Nay88, S75, SSS2]: (1) la función de transferencia del filtro adaptivo debe de ser al menos del orden suficiente (mismo número de polos y ceros) para modelar de una manera adecuada al sistema que se desea identificar, (2) la señal de entrada  $x(n)$  debe ser una secuencia de ruido blanco, y (3) el orden del numerador del filtro adaptivo debe ser mayor que el orden del denominador del sistema que se desea modelar. Nótese que éstas son condiciones para el caso de la identificación de sistemas, para el problema del filtrado deben de existir condiciones similares para la no existencia de mínimos locales, sin embargo existen pocos estudios al respecto [Shy89].

En lo que concierne al problema de los mínimos locales, la condición inicial para el vector de parámetros influye en tanto en la rapidez de la convergencia así como al mínimo al cual el algoritmo converge: por lo tanto  $\theta(0)$  se debe escoger

de tal forma que este cerca del mínimo global ó que este en alguna trayectoria que conduzca a éste último.[Shy89].

Claramente existe un compromiso entre los dos métodos arriba mencionados. Por un lado, la ecuación del error es una función lineal de los coeficientes de tal manera que la superficie del MSEE sólo tiene un mínimo. Los algoritmos derivados de la formulación de la ecuación del error convergen rápidamente pero convergerán inevitablemente a una solución sesgada siempre que exista ruido. Por otra parte, el error de salida es una función no lineal de los parámetros del filtro y de la superficie del error MSOE, la cual puede presentar mínimos locales. Los algoritmos derivados de este método tiene una convergencia más lenta con respecto a los derivados mediante el método de la ecuación de error y además pueden converger a una solución no óptima (mínimo local).

### 4.3 Forma General de los Algoritmos para Filtrado Adaptable a Respuesta Impulsional Infinita

Los algoritmos de filtrado adaptable basados ya sea en el método de la ecuación de error ó en el método del error de salida se pueden expresar de acuerdo a la siguiente forma genérica conocida como el algoritmo recursivo de Gauss-Newton [LS83]:

$$\theta(n + 1) = \theta(n) + \alpha R^{-1}(n + 1) \phi_f(n) \epsilon_f(n) \quad (4.15)$$

donde  $\phi_f$  y  $\epsilon_f(n)$  son versiones filtradas de  $\phi(n)$  y  $e(n)$ , es decir  $\phi_f(n) = F(n, q)\phi(n)$  y  $\epsilon_f(n) = G(n, q)e(n)$ , respectivamente. Nótese que aquí no se ha utilizado el subíndice  $e$  u  $o$  para  $\phi$  y  $\epsilon(n)$  dado que éstas variables nos dan una representación genérica de los algoritmos de filtrado adaptable IIR. El escalar positivo  $\alpha$  es el paso de adaptación y,  $R(n)$  es el estimado de la matrix Hessiana, la cual se actualiza de acuerdo a:

$$R(n + 1) = \lambda R(n) + \alpha \phi_f(n) \phi_f^T(n) \quad (4.16)$$

donde  $\lambda = 1 - \alpha$  es el factor de olvido. Dado que el cálculo de la inversa de una matrix, en éste caso  $R(n)$ , es computacionalmente costoso y complejo, se prefiere hacer el cálculo de una forma recursiva. De esta manera, para desarrollar la recursión para la inversa de  $R(n)$  se aplica el lemma de la inversión matricial (ver apendie A):

$$R^{-1}(n + 1) = \frac{1}{\lambda} \left( R^{-1}(n) - \frac{R^{-1}(n) \phi_f(n) \phi_f^T(n) R^{-1}(n)}{\lambda/\alpha + \phi_f^T(n) R^{-1}(n) \phi_f(n)} \right) \quad (4.17)$$

La estimación de la matrix Hessiana se incorpora para mejorar la tasa de convergencia del algoritmo a costa de un incremento en la carga computacional del mismo. De otra manera, la actualización en 4.12 ó en 4.13 no se lleva a cabo y  $R(n + 1)$  en 4.11 es igual a la matrix identidad  $I$ . A los algoritmos de la forma dada por la ec. 4.11

se les conoce como algoritmos del gradiente estocástico [Hay91], los cuales tienen una tasa de convergencia pequeña. Sin embargo, la complejidad de estos algoritmos es del orden de  $M + N$  operaciones, en lugar de  $(M + N)^2$  que les corresponde a los algoritmos de la forma dada por la ecuación 4.11 en donde la estimación de la matriz Hessiana es diferente de la identidad.

Para que 4.11 converga es necesario que la matriz  $R$  siempre sea definida positiva (positive definite, esto es que exista la inversa) y, que los ceros de  $1 - A(z)$  caigan dentro del círculo unitario.

Se puede demostrar fácilmente para el método de la ecuación de error que  $F(n, q) = G(n, q) = 1$ , es decir, no se filtra el regresor de la ecuación de error. Al algoritmo correspondiente se le conoce como el algoritmo RLS de ganancia normalizada [Lju87], y para el caso de  $R = I$  se le llama algoritmo LMS.

## 4.4 El Algoritmo HARF

En esta sección se presenta el algoritmo HARF, el cuál se fundamenta en la teoría de la Hyperestabilidad [CRJ78, TLJ78], la cuál es una poderosa herramienta que fue originalmente desarrollada para el análisis de sistemas no lineales y variantes en el tiempo [Pop73]. La importancia del algoritmo HARF radica en que es el primer algoritmo propuesto para filtros adaptables a respuesta impulsional infinita que tiene propiedades de convergencia susceptibles de demostrar matemáticamente [CRJ78].

El primero en presentar un algoritmo para estimar los parámetros de un proceso ARMA tomando como base las ideas de la teoría de la hyperestabilidad fue Landau [Lan76], cuyo trabajo lleva por título: "hyperstable output error identifier". Este trabajo es la base del algoritmo HARF dado que éste es una modificación del primero: esta modificación tiene el objetivo de que el algoritmo pueda ser aplicado al campo del procesamiento digital de señales. En concreto las modificaciones que debe sufrir el algoritmo de Landau para su aplicación exitosa al problema del filtrado adaptable son las siguientes: 1) el algoritmo original de Landau contempla una disminución gradual de los factores de ganancia, los cuales deben de converger a cero. Esto último aunque es adecuado para identificar una planta con parámetros constantes, resulta inadecuado para el problema del filtrado estable en un ambiente no estacionario donde se requiere de un seguimiento de los estadísticos de la señal de entrada al filtro adaptable; por lo que se debe modificar el algoritmo de Landau para poder efectuar dicho seguimiento y; 2) el algoritmo de Landau no es estrictamente estable [Lan76], ya que requiere del valor presente de la señal deseada para poder calcular las estimaciones de los parámetros de la planta al tiempo presente, aunque esta condición es razonable para el problema de la identificación de sistemas no lo es para el problema del filtrado adaptable, por lo que es necesario deshacerse de esta condición [LTCRJ80].

### 4.4.1 Introducción a la Teoría de la Hyperestabilidad

Cabe mencionar que el concepto de la hyperestabilidad de los sistemas se debe a Popov [Pop73], quien utilizó este concepto dentro del área del control principalmente para analizar sistemas no lineales y variantes en el tiempo. De esta manera, a éste concepto, la comunidad científica dedicada al procesamiento de señales le presto poca atención hasta que se propuso el algoritmo HARF [LTCRJ80], con esto y debido a que este algoritmo es el primero en poseer propiedades de convergencia matemáticamente probables [CRJ78] surgió un vivo interés en la teoría de la hyperestabilidad, sin embargo a pesar de resolver el problema de la convergencia, el algoritmo HARF, como veremos más adelante plantea el reto de eliminar la condición de "strictly positive real (SPR)" [JT79], la cuál es un serio obstáculo para la aplicación práctica de este algoritmo [CRJT80, PSA79, CRJLTD81].

La hyperestabilidad se define para el caso discreto como sigue [And68, Lar79]: sea  $G(z)$  una función de transferencia racional y escalar de un sistema lineal e

invariante en el tiempo cuya entrada se denota con  $u(k)$  y su salida con  $y(k)$ . Se dice que dicho sistema es hyperestable si su vector de estado mantiene una magnitud finita para todas las posibles señales de entrada con sus correspondientes señales de salida que satisfagan [CG85]:

$$\sum_{l=0}^{K_0} u(l)y(l) < K^2, \quad \forall K_0 \quad (4.18)$$

Los algoritmos discutidos en [JT79, CG85, LTCRJ80] no utilizan esta definición tal cual sino una variación de la misma conocida como Hyperestabilidad asintótica, la cual requiere que todos los elementos del vector de estado convergan a cero para la clase de señales de entrada discutidas en el párrafo anterior y descritas matemáticamente por la ec. 4.16; de aquí en adelante al hacer referencia al término de hyperestabilidad se entenderá que se alude a esta última definición. Nótese que si la función de transferencia  $G(z)$  es racional propia, la salida  $y(n)$  también decaerá a cero [LTCRJ80].

A continuación enunciamos el teorema de la hyperestabilidad [And68, Pop73]: "El sistema anteriormente descrito es hyperestable si y sólo si la parte real de su función de transferencia es estrictamente positiva (SPR de sus siglas en inglés) [HA69], esto es:

$$\text{Re} [G(z)] > 0, \quad z = e^{j\theta} \quad (4.19)$$

Esto es, un sistema que cumpla con la condición SPR tendrá una salida finita cuando la señal de entrada satisfaga la ec. 4.16. Aquí es necesario hacer notar que esta última condición es suficiente mas no necesaria, pues pueden existir secuencias de entrada divergentes para las cuales la salida del sistema siempre tiene un valor finito [LTCRJ80].

El concepto de la hyperestabilidad es de utilidad para el filtrado adaptable cuando se considera su aplicación al sistema de malla cerrada mostrado en la figura 4.2 [CG85]; la adaptación de los parámetros de los filtros adaptables puede ser redefinida en términos de esta configuración [Lar79]. Sea  $u(n)$  una secuencia derivada como una función no lineal y variante en el tiempo de la salida del sistema; si el elemento de retroalimentación  $F(z)$  es lineal, la BIBO estabilidad del sistema puede ser fácilmente verificada en el dominio de la frecuencia, al hallar los ceros de:

$$1 + F(z)G(z) \quad (4.20)$$

Cualquier cero que se localice fuera del círculo unitario implica que el sistema de la fig. 4.2 es inestable. Una interpretación física de este hecho establece que la ganancia de lazo cerrado a cualquier frecuencia jamás debe ser igual a -1, es decir, no se debe presentar un defasamiento de  $180^\circ$ . Una condición suficiente mas no necesaria para cumplir con este último requisito, consiste en restringir la contribución a la fase del sistema de lazo cerrado tanto de  $F(z)$  como de  $G(z)$  para que cada una de estas contribuciones sea menor en magnitud a un defasamiento de  $90^\circ$ . Esto es, si tanto  $F(z)$  como  $G(z)$  cumplen con la condición SPR, el sistema de lazo cerrado será estable [CG85].

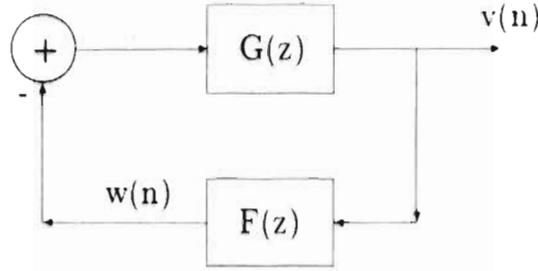


Figura 4.2: Sistema de lazo cerrado a excitación nula.

Esta condición no es útil cuando se desean analizar sistemas que involucran una retroalimentación no lineal y variante en el tiempo, por lo que para estos casos se requiere reformular esta condición en términos más generales. Sea  $\mathcal{F}$  un elemento general de retroalimentación, el cual generalmente será no lineal y variante en el tiempo; y sea  $v(n)$  la señal de entrada a este elemento. Entonces si tenemos una configuración igual a la de la figura 4.2 con la única excepción de que hay que sustituir al elemento lineal de retroalimentación,  $F(z)$ , por el elemento general de retroalimentación  $\mathcal{F}$  tendremos que la salida de este último elemento estará dada por  $w(k) = -u(k)$ . Entonces si se cumple que:

$$\sum_{l=0}^K v(l)w(l) \geq -\gamma_0^2, \quad \forall K > 0. \quad (4.21)$$

se sigue que:

$$\sum_{l=0}^K u(l)y(l) \leq \gamma_0^2, \quad \forall K > 0. \quad (4.22)$$

De esta manera, si  $G(z)$  cumple con la SPR entonces el sistema de lazo cerrado será hyperstable [CG85]. Esto representa una forma de generalizar la condición SPR al caso de sistemas de lazo cerrado con retroalimentación no lineal y variante en el tiempo. En lo que respecta a los sistemas lineales, se puede demostrar mediante el análisis de eigenfunciones que esta condición de hecho asegura que la fase a cualquier frecuencia tanto de la planta como del elemento de retroalimentación no excederá en magnitud a los  $90^\circ$  [LTCRJ80]. Desde un punto de vista más ingenieril si la energía alimentada al elemento de retroalimentación  $\mathcal{F}$ :

$$\sum_{l=0}^{K_0} v(l)w(l) \quad (4.23)$$

cumple con la ec. 4.19, entonces  $\mathcal{F}$  es disipativo [CG85]. Concluyendo, el teorema de la hyperestabilidad garantiza la estabilidad para una clase de sistemas intrínsecamente

no lineales y variantes en el tiempo. Claramente el requisito expresado por la ec. 4.16 representa una condición suficiente mas no necesaria, por lo cual es demasiado restrictiva [CG85]. Para el caso lineal citado anteriormente, la condición SPR tanto para la planta como para el elemento de retroalimentación no es necesaria para que el sistema sea estable. Por tanto, en el caso general se puede esperar que estas condiciones no sean necesarias; de aquí qué sea necesaria la formulación de un criterio menos restrictivo para el caso de los sistemas no lineales, sin embargo la teoría hasta aquí tratada es satisfactoria para el análisis de los algoritmos de filtrado adaptable abordados en este capítulo.

#### 4.4.2 El Algoritmo HARF

Considere la figura 4.3, donde  $\hat{A}$  denota el vector de coeficientes backward dado por  $[\hat{a}_1, \dots, \hat{a}_N]$ ;  $\hat{B}$  es el vector de coeficientes forward dado por  $[\hat{b}_1, \dots, \hat{b}_M]$ ;  $\hat{C}$  es el vector de coeficientes del proceso de promedio móvil (el cual se utiliza para suavizar el error dado por la diferencia de  $f(n)$  y  $d(n)$ ), explícitamente  $\hat{C} = [\hat{c}_1, \dots, \hat{c}_N]$  y; finalmente,  $n$  denota el índice temporal. De esta figura se observa inmediatamente que además del filtro adaptivo principal con salida  $y(n)$  tenemos el proceso auxiliar  $f(n)$  dado por:

$$f(n) = \sum_{i=1}^N \hat{a}_i(n+1)f(n-i) + \sum_{j=0}^M \hat{b}_j(n+1)x(n-j) \quad (4.24)$$

Mientras que  $y(n)$  esta dada por:

$$y(n) = \sum_{i=1}^N \hat{a}_i(n)f(n-i) + \sum_{j=0}^M \hat{b}_j(n)x(n-j) \quad (4.25)$$

Nótese que los parámetros en la ec. 4.22 se han adaptado un paso más con respecto a los que aparecen en la ecuación 4.21. Obsérvese que si finalmente el algoritmo ha de converger, entonces  $\hat{a}_i(n+1) = \hat{a}_i(n)$ ,  $\hat{b}_i(n+1) = \hat{b}_i(n)$  y, además  $y(n)$  converge asintóticamente a  $f(k)$ . Sin embargo, en el proceso de adaptación, la distinción entre  $y(n)$  y  $f(n)$  es necesaria.

La formulación hyperestable del algoritmo HARF continua con la adaptación de los parámetros backward y forward [CG85]:

$$\begin{aligned} \hat{a}_i(n) &= \hat{a}_i(n-1) + \frac{\mu_i}{q(n)} f(n-i-1) (\{d(n-1) - y(n-1)\} \\ &\quad + \sum_{l=1}^P c_l \{d(n-l-1) - f(n-l-1)\}) \quad 1 \leq i \leq N, \end{aligned} \quad (4.26)$$

$$\begin{aligned} \hat{b}_j(n) &= \hat{b}_j(n-1) + \frac{\mu_j}{q(n)} x(n-j-1) (\{d(n-1) - y(n-1)\} \\ &\quad + \sum_{l=1}^P c_l \{d(n-l-1) - f(n-l-1)\}) \quad 0 \leq j \leq M. \end{aligned} \quad (4.27)$$

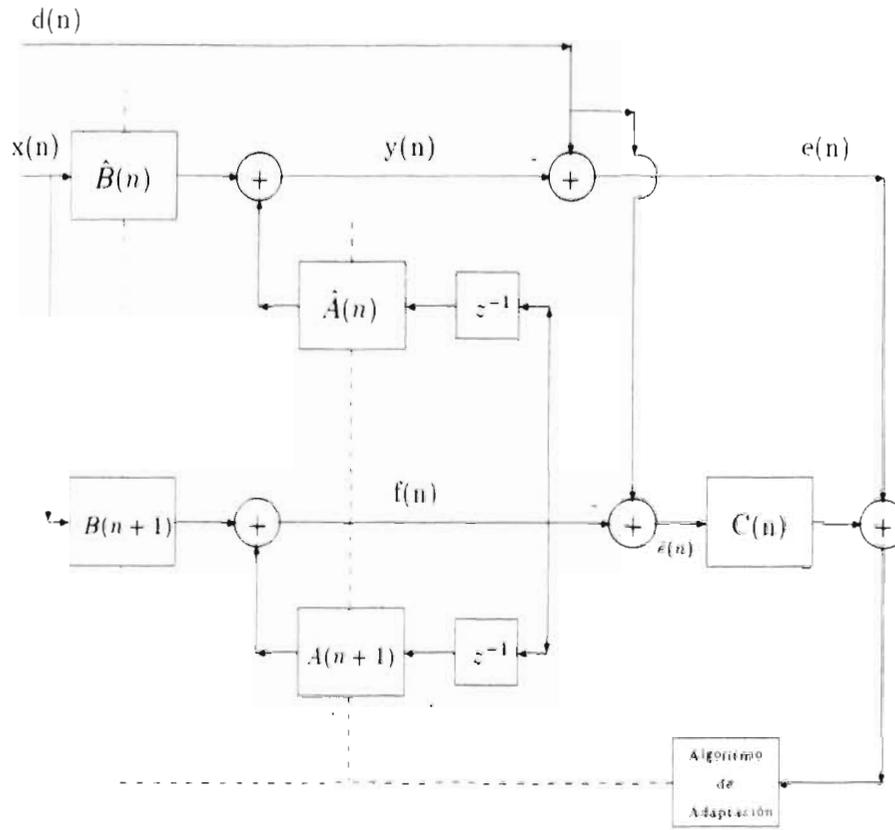


Figura 4.3: Implementación del algoritmo IIARF.

donde en las dos últimas ecuaciones  $q(k)$  es un factor de normalización mayor que la unidad.

$$q(n) = 1 + \sum_{l=1}^N \mu_l f^2(n-l-1) + \sum_{l=0}^M \rho_l x^2(n-l-1) \quad (4.28)$$

donde  $\mu_l$  y  $\rho_l$  son constantes positivas de valor arbitrario. Además las  $P$  constantes  $c_l$  deben ser escogidas por el diseñador de tal manera que la función de transferencia:

$$G(z) = \frac{1 + \sum_{l=1}^P c_l z^{-l}}{1 - \sum_{l=1}^N a_l z^{-l}} \quad (4.29)$$

cumpla con el criterio SPR. Bajo esta condición se puede demostrar [CRJ78], que la variable:

$$v(n) = (d(n) - f(n)) + \sum_{l=1}^P c_l (d(n-l) - f(n-l)) \quad (4.30)$$

converge a cero como resultado de:

$$y(n) \rightarrow f(n) \rightarrow d(n) \quad (4.31)$$

lo que representa el desempeño deseado. Finalmente, cabe mencionar que en [CRJ78] se demuestra que el algoritmo hasta aquí descrito satisface las condiciones impuestas por la teoría de la hyperestabilidad.

#### 4.4.3 Organigrama del Algoritmo HARF.

Ecuaciones	
$f(n) = \sum_{i=1}^N \hat{a}_i(n+1)f(n-i) + \sum_{j=0}^M \hat{b}_j(n+1)x(n-j)$	(4.22)
$y(n) = \sum_{i=1}^N \hat{a}_i(n)f(n-i) + \sum_{j=0}^M \hat{b}_j(n)x(n-j)$	(4.23)
$\hat{a}_i(n) = \hat{a}_i(n-1) + \frac{\mu_i}{y(n)} f(n-i-1) (\{d(n-1) - y(n-1)\} + \sum_{l=1}^P c_l \{d(n-l-1) - f(n-l-1)\}) \quad 1 \leq i \leq N.$	(4.24)
$\hat{b}_j(n) = \hat{b}_j(n-1) + \frac{\rho_j}{y(n)} x(n-j-1) (\{d(n-1) - y(n-1)\} + \sum_{l=1}^P c_l \{d(n-l-1) - f(n-l-1)\}) \quad 0 \leq j \leq M.$	(4.25)
$q(n) = 1 + \sum_{i=1}^N \mu_i f^2(n-i-1) + \sum_{j=0}^M \rho_j x^2(n-j-1)$	(4.26)

Tabla 4.1: Algoritmo HARF.

#### 4.4.4 El Algoritmo SARF

A pesar de la solución que brinda el algoritmo HARE al problema del filtrado adaptable para filtros de respuesta impulsional infinita, dicho algoritmo es computacionalmente costoso pues sufre de los dos siguientes problemas: 1) para poder calcular la salida del filtro,  $y(n)$ , se requiere del cálculo del proceso auxiliar  $f(n)$ , el cual a su vez es necesario para actualizar los parámetros del filtro adaptivo y; 2) por cada iteración del algoritmo se requiere calcular el factor de normalización  $q(n)$ . Estos dos problemas incrementan substancialmente el costo de la implementación del algoritmo, por lo que se impone la necesidad de buscar alguna simplificación al mismo que lo vuelva más susceptible de aplicar en problemas de índole práctica y no meramente teóricos. Con el objetivo de poder aplicar el algoritmo HARE a problemas de tiempo real en [LTCRJ80] se proponen la siguiente modificación: especificar las constantes  $\mu$  y  $\rho$  lo suficientemente pequeñas de tal manera que los parámetros del filtro sufran un cambio pequeño de iteración en iteración, es decir:

$$\begin{aligned} \hat{a}_i(n+1) &\cong \hat{a}_i(n) \\ \hat{b}_j(n+1) &\cong \hat{b}_j(n) \end{aligned} \quad (4.32)$$

Si se utiliza esta última aproximación en las ecuaciones (4.22) y (4.23) tenemos:

$$f(n) \cong y(n)$$

Entonces (4.23) se puede expresar como:

$$y(n) = \sum_{i=1}^N \hat{a}_i(n)y(n-i) + \sum_{j=0}^M \hat{b}_j(n)x(n-j) \quad (4.33)$$

por tanto, el proceso de promedio móvil  $v(n)$ , dado por la ec.(4.28), se puede expresar como:

$$\begin{aligned} v(n) &\cong \{d(n) - y(n)\} + \sum_{l=1}^P c_l \{d(n-l) - y(n-l)\} \\ &= e(n) + \sum_{l=1}^P c_l e(n-l). \end{aligned} \quad (4.34)$$

Finalmente, nótese que el factor de normalización  $q(n)$  controla la tasa instantánea de adaptación mediante la reducción del factor de corrección aplicado a los parámetros del filtro para grandes valores tanto de la señal de entrada como de la señal de salida del filtro. De esta manera y dado que anteriormente se asumió que tanto  $\mu$  como  $\rho$  toman valores pequeños, se puede asumir que:

$$q(n) \cong 1$$

Si se usan las aproximaciones desritas anteriormente, las ecuaciones para actualizar los parámetros del filtro se pueden expresar como:

$$\hat{a}_i(n) \cong \hat{a}_i(n-1) + \mu_i y(n-1-i)v(n-1), \quad 1 \leq i \leq N \quad (4.35)$$

$$\hat{b}_j(n) \cong \hat{b}_j(n-1) + \rho_j x(n-1-j)v(n-1), \quad 1 \leq j \leq M \quad (4.36)$$

El conjunto de ecuaciones que comprende desde la ec. 4.21 hasta la ec. 4.24 constituye la versión simplificada del algoritmo HARF, de ahí que esta versión simplificada reciba el nombre de "simple hyperstable adaptive recursive filter (SARF)".

Nótese que en el algoritmo SARF para actualizar los parámetros del filtro sólo se requiere el conocimiento del error de salida suavizado  $v(n)$ , lo cual reduce de manera significativa el tiempo de procesamiento con respecto al algoritmo HARF. Sin embargo, esta reducción en el tiempo de procesamiento se realiza a costa de no satisfacer de una manera rigurosa la condición de hyperestabilidad dada por la ec. 4.19, por lo que la convergencia no está garantizada para cualquier par de valores positivos que se le asignen a  $\mu$  y  $\rho$  [CG85]. Sin embargo, algunos autores conjeturan que para el caso de una adaptación lenta el algoritmo SARF presenta hyperestabilidad [CG85, LTRJ80].

#### 4.4.5 Organigrama del Algoritmo SARF.

Ecuaciones	
$y(n) = \sum_{i=1}^N \hat{a}_i(n)y(n-i) + \sum_{j=0}^M \hat{b}_j(n)x(n-j)$	(4.31)
$e(n) \cong \{d(n) - y(n)\} + \sum_{l=1}^P c_l \{d(n-l) - y(n-l)\}$ $= e(n) + \sum_{l=1}^P c_l e(n-l).$	(4.32)
$\hat{a}_i(n) \cong \hat{a}_i(n-1) + \mu_i y(n-1-i)v(n-1), \quad 1 \leq i \leq N$	(4.33)
$\hat{b}_j(n) \cong \hat{b}_j(n-1) + \rho_j x(n-1-j)v(n-1), \quad 1 \leq j \leq M$	(4.34)

Tabla 4.2: Algoritmo SARF.

## 4.5 Algoritmos SIM, AFM e IF

### 4.5.1 "Least Mean Square Error Equation"

Nuevamente, para el desarrollo de este algoritmo se recurre al método del error de salida, el cuál se discutió anteriormente. De la figura 4.1 tenemos:

$$w(n) = \sum_{i=1}^{n_a} a_i w(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) \quad (4.37)$$

$$y(n) = w(n) + v(n) \quad (4.38)$$

$$\hat{y}(n) = \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n) \hat{y}(n-i) + \sum_{i=0}^{\hat{n}_b} \hat{b}_i(n) x(n-i) \quad (4.39)$$

$$e(n) = y(n) - \hat{y}(n) \quad (4.40)$$

El mayor problema para la formulación de un algoritmo de filtrado adaptable, cuya estructura se ilustra en la figura 4.1 y su representación matemática esta dada por las cuatro ecuaciones anteriores, radica en las recursiones sobre las variables  $w(n)$  y  $\hat{y}(n)$ ; estas recursiones dificultan la solución matemática al problema de la minimización de la función de costo  $J = E\{e^2\}$ . En la literatura referente a la indentificación de sistemas se ha propuesto solucionar este problema mediante el filtrado de la señal de error a través de un filtro todo cero, el cuál es igual al denominador de la función de transferencia del filtro adaptivo, de tal manera que la señal filtrada del error o ecuación de error sea una función lineal de los parámetros del filtro adaptivo:  $\hat{a}_i(n)$  y de  $\hat{b}_i(n)$  [AE71]. Esto es, si consideramos por el momento a  $x'(n)$  como la entrada, a  $e'(n)$  como el error de salida y, a  $v'(n)$  como la perturbación (vease la fig. 4.2); entonces  $e(n)$  se convierte en la ecuación de error y se expresa como:

$$e(n) = e'(n) - \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n) e'(n-i) \quad (4.41)$$

dónde:

$$e'(n) = y'(n) - \hat{y}'(n) \quad (4.42)$$

$$w'(n) = \sum_{i=1}^{n_a} a_i w'(n-i) + \sum_{i=0}^{n_b} b_i x'(n-i) \quad (4.43)$$

$$y'(n) = w'(n) + v'(n) \quad (4.44)$$

$$v'(n) = v(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n) v'(n-i) \quad (4.45)$$

$$\hat{y}'(n) = \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n) \hat{y}'(n-i) + \sum_{i=0}^{\hat{n}_b} \hat{b}_i(n) x'(n-i) \quad (4.46)$$

Nótese que el conjunto de ecuaciones que comprende de la ecuación 4.40 a la 4.44

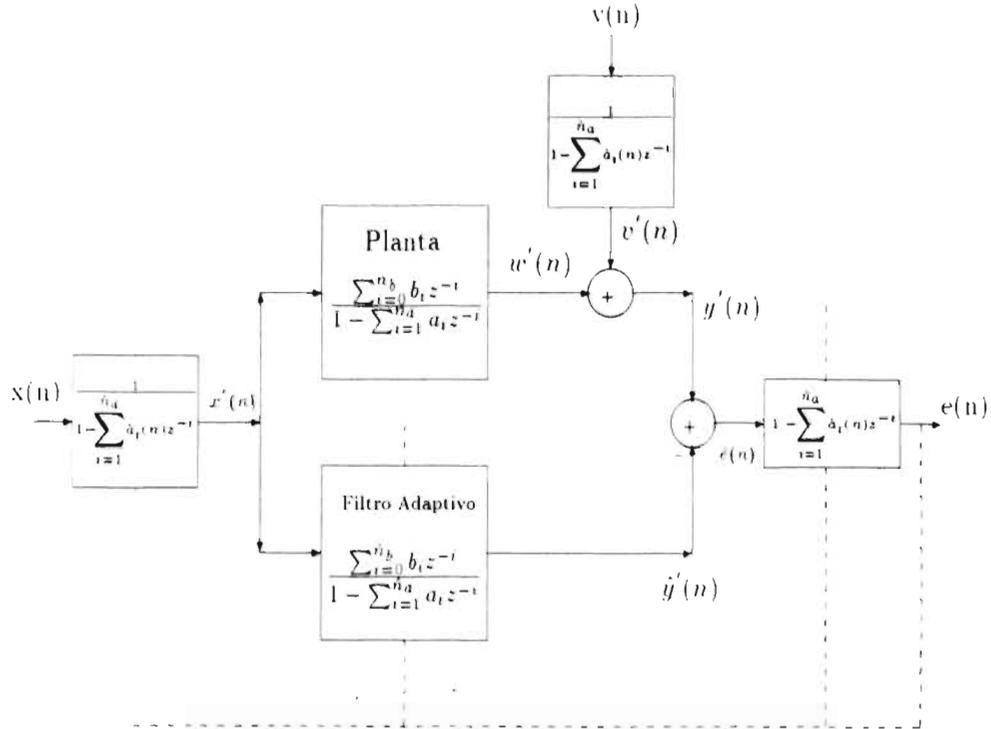


Figura 4.4: Modo de Identificación de sistemas para el algoritmo SIM.

es análogo a las ecuaciones 4.35, 4.36, 4.37 y 4.38. Sustituyendo las ecuaciones 4.40 y 4.44 en 4.39 tenemos:

$$\begin{aligned}
 \epsilon(n) &= y'(n) - \sum_{i=1}^{n_a} \hat{a}_i(n) y'(n-i) - [y'(n) - \sum_{i=1}^{n_a} \hat{a}_i(n) \hat{y}'(n-i)] \\
 &= y'(n) - \sum_{i=1}^{n_a} \hat{a}_i(n) y'(n-i) - \sum_{i=0}^{n_b} \hat{b}_i(n) x'(n-i)
 \end{aligned}
 \tag{4.47}$$

En esta última ecuación  $\epsilon(n)$  será una función lineal de los parámetros  $\hat{a}_i(n)$  y  $\hat{b}_i(n)$  solo si tanto  $y'(n)$  como  $x'(n)$  son independientes de estos mismos parámetros. Si se cumple lo anterior la minimización de la función de costo  $J = E\{\epsilon^2(n)\}$  se puede llevar a cabo mediante el método "least mean squares (LMS)" [FJ86], es decir:

$$\begin{aligned}
 \frac{\partial}{\partial \hat{a}_i(n)} E\{\epsilon^2(n)\} &\approx 2\epsilon(n) \frac{\partial}{\partial \hat{a}_i(n)} \epsilon(n) \\
 &= -2\epsilon(n) y'(n-i), \quad i = 1, 2, \dots, \hat{n}_a
 \end{aligned}
 \tag{4.48}$$

$$\begin{aligned} \frac{\partial}{\partial \hat{b}_i(n)} E\{\epsilon^2(n)\} &\approx 2\epsilon(n) \frac{\partial}{\partial \hat{b}_i(n)} \epsilon(n) \\ &= -2\epsilon(n)x'(n-i), \quad i = 0, 1, \dots, \hat{n}_b \end{aligned} \quad (4.19)$$

donde las recursiones para los parámetros están dadas por:

$$\hat{a}_i(n+1) = \hat{a}_i(n) + 2k_1\epsilon(n)y'(n-i), \quad i = 1, 2, \dots, \hat{n}_a \quad (4.50)$$

$$\hat{b}_i(n+1) = \hat{b}_i(n) + 2k_2\epsilon(n)x'(n-i), \quad i = 0, 1, \dots, \hat{n}_b \quad (4.51)$$

donde  $k_1$  y  $k_2$  son constantes. El método hasta aquí descrito recibe el nombre de "least mean square equation error (LMSEE)". Este método conduce a la minimización del cuadrado del error de salida sólo en el caso de que el mínimo valor de este error sea igual a cero, esto corresponde a los casos uno y dos de la clasificación de las superficies del error que se mencionaron en la sección tres del presente capítulo.

#### 4.5.2 Desarrollo de los Algoritmos SIM, AFM e IF

Los algoritmos aquí tratados proponen prefiltrar la señal de entrada  $x(n)$ , a través de un filtro todo polo que es igual al inverso del denominador del filtro adaptivo; tal como se muestra en la fig. 4.4. Note que la respuesta de este filtro dada por  $e(n)$  en general no es igual a la respuesta del filtro de la figura 4.1, esto debido a la varianza en el tiempo que presentan ambos filtros. Sin embargo, algunos autores conjeturan que en el caso de una adaptación lenta, el filtro de la figura 4.4 se aproxima al de la figura 4.1 [FJ86]. Nótese además que ambos filtros son equivalentes, en el caso de que convergan. Por lo tanto, la ecuación de error  $e(n)$  con respecto a  $x'(n)$  puede interpretarse como el error de salida cuando la entrada es  $x(n)$  [FJ86]. Entonces el esquema de minimización del error, dado por el conjunto de ecuaciones que comprende de la ecuación 4.39 a la ec. 4.19, puede ser aplicado si  $x'(n)$  e  $y'(n)$  son independientes de los parámetros del filtro adaptivo, es decir, de  $\hat{a}_i(n)$  y de  $\hat{b}_i(n)$ . Esto aparentemente no es cierto debido a la operación de prefiltrado. Sin embargo, se observa que la ecuación 4.17 es válida dado que tanto  $x'(n)$  como  $y'(n)$  son independientes de los coeficientes  $\hat{b}_i$ ; y a que la derivada de  $e(n)$  con respecto a  $\hat{a}_i(n)$  se puede aproximar mediante [FJ86]:

$$\frac{\partial}{\partial \hat{a}_i(n)} \epsilon(n) \approx -w'(n-i), \quad i = 1, 2, \dots, \hat{n}_a \quad (4.52)$$

Note que esta aproximación es razonable debido a que tanto  $v(n)$  como  $v'(n)$  no se pueden medir, por lo que  $y'(n-j)$  es la única variable que se puede utilizar para estimar  $w'(n-j)$ . Nótese que esta aproximación es exacta para el caso  $v(n) \equiv 0$ , y además nos permite aplicar el esquema de minimización dado por el conjunto de ecuaciones que comprende de la ecuación 4.29 a la ec. 4.39, al filtro adaptable de la figura 4.4.

Finalmente, el algoritmo en su modo de identificación de sistemas (SIM) está dado por las ecuaciones 4.39, 4.40, 4.41, 4.42, 4.43, 4.44, 4.48 y 4.49, junto con:

$$x'(n) = x(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)x'(n-i) \quad (4.53)$$

Nótese que el algoritmo SIM no es realizable en los siguientes casos:

- Dado que tanto  $v(n)$  como  $v'(n)$  son cantidades no mensurables, la ec. 4.43 nos indica que el algoritmo SIM no es realizable cuando  $v(n) \neq 0$ .
- El algoritmo no es realizable para su modo de filtrado adaptable, donde  $y(n)$  es la respuesta deseada, ya que  $y'(n)$  no se puede obtener mediante la medición de la señal de salida de la planta cuando a ésta se le alimenta con  $x'(n)$  (tal y como es descrito por las ecs. 4.11 y 4.13), dado que la función de transferencia de la planta se desconoce [FJ86]. Por tanto, se propone otra versión de este algoritmo, la cuál está dada por las siguientes ecuaciones (4.39), (4.40), (4.44), (4.48), (4.49), (4.51) y:

$$y'(n) = y(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)y'(n-i) \quad (4.54)$$

donde  $y(n)$  es la respuesta deseada. Nótese que las dos etapas de prefiltrado dadas por las ecuaciones 4.51 y 4.52 son necesarias para obtener  $x'(n)$  e  $y'(n)$ , las cuales a su vez son necesarias para actualizar los coeficientes. Sin embargo, no es necesario calcular  $e(n)$  usando (4.39), (4.40), (4.44), (4.51) y (4.52). De hecho, dado que se puede considerar que en el caso de una adaptación lenta, que la etapa de prefiltrado se cancela con la de posfiltrado,  $e(n)$  se puede calcular directamente de (4.38). Por lo tanto, el modo de filtrado adaptivo de este algoritmo (algoritmo AFM) está dado por las ecuaciones (4.37), (4.38), (4.48), (4.49), (4.51) y (4.52). En la figura 4.5 se muestra la implementación de este algoritmo, la cuál es más eficiente que la implementación mostrada en la fig. 4.4, ya que sólo utiliza dos filtros adicionales al filtro adaptivo y al mecanismo de actualización de parámetros[FJ86].

De una manera análoga a la utilizada por Setarns [CRJ84, SEA76], se puede obtener una versión más compleja del algoritmo propuesto, de la siguiente manera: en lugar de usar una versión temporalmente desplazada de  $y'(n)$  y de  $x'(n)$  como se hizo en (4.48) y (4.49), un conjunto de variables independientemente filtradas ( $x'_j(n)$  y  $y'_i(n)$ ) es utilizado. Estas variables se obtienen mediante:

$$y'_i(n) = y(n-i) + \sum_{m=1}^{\hat{n}_a} \hat{a}_m(n)y'_i(n-m), \quad i = 1, 2, \dots, \hat{n}_a \quad (4.55)$$

$$x'_j(n) = x(n-j) + \sum_{m=1}^{\hat{n}_a} \hat{a}_m(n)x'_j(n-m), \quad j = 0, 1, \dots, \hat{n}_b \quad (4.56)$$

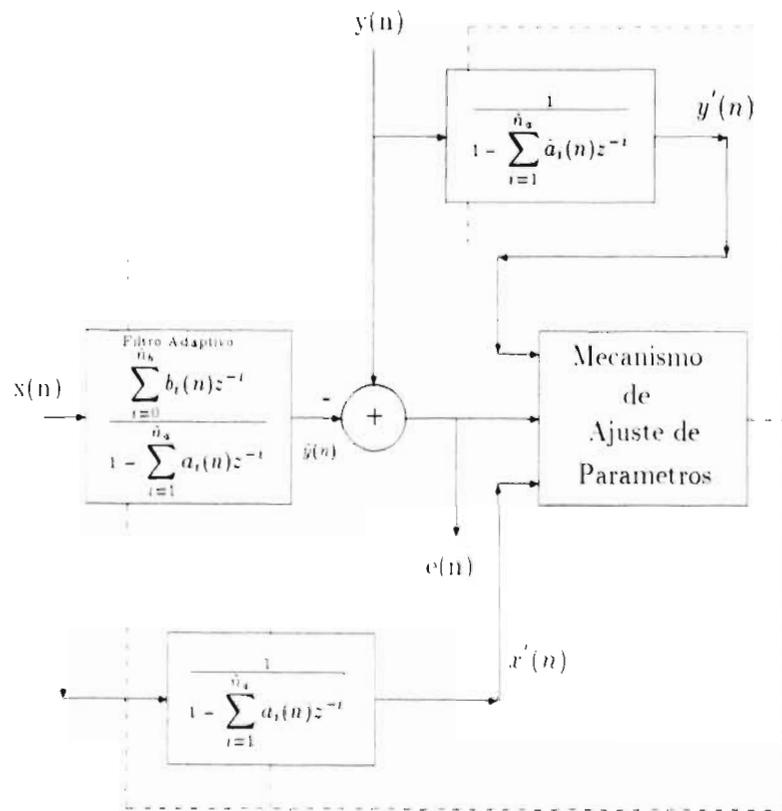


Figura 1.5: Modo de filtrado adaptivo para el algoritmo AFM.

Estas dos últimas ecuaciones pueden ser aplicadas tanto al algoritmo SIM como al IF, sin embargo dado que este último tiene una mayor aplicabilidad a casos prácticos y que además puede ser utilizado para identificar sistemas; sólo se considerará la versión correspondiente al algoritmo AFM. A esta versión del algoritmo AFM se le conoce como "independent filtering algorithm (IF)". A pesar que la literatura reporta que este último algoritmo no exhibe mayores ventajas computacionales que los algoritmos AFM y SIM [FJ86], si se ha demostrado de una manera rigurosa su convergencia [Fan85].

Por último cabe hacer notar que en los algoritmos aquí tratados, la estabilidad no esta de ninguna manera garantizada por lo que se requiere incorporar un mecanismo encargado de verificar la estabilidad del algoritmo y, en caso necesario lograr la estabilización del mismo [Fan85, CRJ84]. Sin embargo, sea ha especulado sin ninguna prueba rigurosa que para el caso de que las constantes  $k_1$  y  $k_2$  sean lo suficientemente pequeñas, no se necesita verificar la estabilidad del algoritmo [FJ86].

### 4.5.3 Organigrama de los Algoritmos SIM, AFM e IF

Ecuaciones	
$\hat{a}_i(n + 1) = \hat{a}_i(n) + 2k_1\epsilon(n)y'(n - i), \quad i = 1, 2, \dots, \hat{n}_a$	(4.48)
$\hat{b}_i(n + 1) = \hat{b}_i(n) + 2k_2\epsilon(n)x'(n - i), \quad i = 0, 1, \dots, \hat{n}_b$	(4.49)
$x'(n) = x(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)x'(n - i)$	(4.51)
$\epsilon(n) = c'(n) - \sum_{j=1}^{\hat{n}_a} \hat{a}_j(n)\epsilon'(n - j)$	(4.59)
$\epsilon'(n) = y'(n) - \hat{y}'(n)$	(4.54)
$w'(n) = \sum_{i=1}^{n_a} a_i w'(n - i) + \sum_{i=0}^{n_b} b_i x'(n - i)$	(4.41)
$y'(n) = w'(n) + v'(n)$	(4.42)
$v'(n) = v(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)v'(n - i)$	(4.43)
$\hat{y}'(n) = \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)\hat{y}'(n - i) - \sum_{i=0}^{\hat{n}_b} \hat{b}_i(n)x'(n - i)$	(4.44)

Tabla 1.3: Algoritmo SIM.

Ecuaciones	
$\hat{a}_i(n + 1) = \hat{a}_i(n) + 2k_1\epsilon(n)y'(n - i), \quad i = 1, 2, \dots, \hat{n}_a$	(4.48)
$\hat{b}_i(n + 1) = \hat{b}_i(n) + 2k_2\epsilon(n)x'(n - i), \quad i = 0, 1, \dots, \hat{n}_b$	(4.49)
$y'_i(n) = y(n - i) + \sum_{m=1}^{\hat{n}_a} \hat{a}_m(n)y'_i(n - m), \quad i = 1, 2, \dots, \hat{n}_a$	(4.52)
$x'(n) = x(n) + \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)x'(n - i)$	(4.51)
$\epsilon(n) = y(n) - \hat{y}(n)$	(4.38)
$\hat{y}'(n) = \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)\hat{y}'(n - i) + \sum_{i=0}^{\hat{n}_b} \hat{b}_i(n)x'(n - i)$	(4.37)

Tabla 4.4: Algoritmo AFM.

Ecuaciones	
$\hat{a}_i(n + 1) = \hat{a}_i(n) + 2k_1\epsilon(n)y'_i(n), \quad i = 1, 2, \dots, \hat{n}_a$	(4.55)
$\hat{b}_j(n + 1) = \hat{b}_j(n) + 2k_2\epsilon(n)x'_j(n), \quad j = 0, 1, \dots, \hat{n}_b$	(4.56)
$y'_i(n) = y(n - i) + \sum_{m=1}^{\hat{n}_a} \hat{a}_m(n)y'_i(n - m), \quad i = 1, 2, \dots, \hat{n}_a$	(4.53)
$x'_j(n) = x(n - j) + \sum_{m=1}^{\hat{n}_a} \hat{a}_m(n)x'_j(n - m), \quad j = 0, 1, \dots, \hat{n}_b$	(4.54)
$\epsilon(n) = y(n) - \hat{y}(n)$	(4.38)
$\hat{y}(n) = \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n)\hat{y}(n - i) + \sum_{j=0}^{\hat{n}_b} \hat{b}_j(n)x(n - j)$	(4.37)

Tabla 4.5: Algoritmo IF.

## 4.6 Realizaciones Alternativas

Con el objeto de resolver algunos de los problemas asociados con la forma directa de los filtros adaptivos IIR, se han propuesto algoritmos para el filtrado adaptable IIR en arquitecturas u estructuras alternativas tales como: la forma paralela, en cascada y en "lattice". Estas estructuras ofrecen un monitoreo más sencillo de la estabilidad y tienen una sensibilidad menor a los errores de tipo numérico que se presentan por efectos de cuantización, redondeo y truncamiento [Shy89].

### 4.6.1 Estructura en Paralelo

La estructura paralela [JN86] se deriva de la expansión en fracciones parciales del filtro IIR dado por la ecuación 4.8. Esta expansión trae como consecuencia que el filtro dado por la ec. 4.8 se pueda sustituir por una estructura en paralelo de  $L = (N - 1)/2$  filtros de segundo orden [Shy89]. La ventaja de este tipo de estructura es que el monitoreo de la estabilidad es un proceso trivial dado que sólo se requiere checar la estabilidad de un sistema de segundo orden  $L$  veces; además las componentes del gradiente se calculan fácilmente debido a que los diferentes filtros son independientes entre sí, es decir, el gradiente de un filtro no depende de los coeficientes de otro filtro [Shy89].

La desventaja principal de la estructura paralela radica en que existen varios mínimos globales a los que el algoritmo puede converger de acuerdo al ordenamiento ó distribución de polos que se realice para cada filtro o sección de la estructura paralela [Shy89].

### 4.6.2 Estructura "Lattice"

Al igual que en el caso de los filtros adaptables FIR, se ha propuesto el uso de una estructura "lattice". La principal ventaja de esta estructura consiste en que el monitoreo de la estabilidad se vuelve un proceso trivial dado que requiere que el valor absoluto de cada coeficiente de reflexión sea menor a uno [MAG76]. Una ventaja de la estructura "lattice" sobre la estructura paralela es que sólo existe una representación para cualquier conjunto de coeficientes de la forma directa ó del filtro de la ec. 4.8. Esto implica que no existen puntos de silla en la superficie del error como en el caso de la estructura en paralelo.

## 4.7 Conclusiones

En este capítulo se introdujeron los algoritmos que adaptan los coeficientes de un filtro IIR, como se señaló en su oportunidad el desarrolloma matemático de este tipo de algoritmos se vuelve mucho más complicado debido a la retoalimentación que existe. Estos algoritmos se pueden clasificar como en los que se basan en la ecuación de error y, los que se basan en el error de salida. Los dos métodos presentan sus

propios problemas: el método de la ecuación de error en la presencia de ruido nos da como resultado una solución sesgada, mientras que el método del error de salida puede converger a un mínimo local. Estos dos problemas son en esencia un serio obstáculo para la utilización de estos métodos ya que por un lado si sabemos que el canal es ruidoso entonces la solución que se obtenga por medio de un algoritmo que se base en el método de la ecuación de error no será la óptima sino será una solución subóptima. Mientras que por otro lado la utilización de un algoritmo basado en el método del error de salida puede converger a un mínimo local. Aunque se ha demostrado la no existencia de mínimos locales bajo ciertas circunstancias muy específicas para el caso de la identificación de sistemas esto no es de gran ayuda para el caso del filtrado dado que se requiere de un incremento en el orden del filtro lo cual desde el punto de vista computacional no es de ninguna manera eficiente. Además en este último caso no se ha demostrado ningún resultado similar aunque se especule con su existencia. Sin embargo, debemos recordar que la igualación de canal presenta problemas y retos diferentes al de la identificación de sistemas por lo que el problema aquí tratado debe ser considerado tomando en cuenta estas diferencias.

Finalmente cabe hacer mención que el problema de la igualación de canal no se ha resuelto de ninguna manera y, para mostrarlo considere que para el caso de un canal de fase no mínima no existe su inverso por lo tanto lo único que se puede hacer desde el punto de vista de los algoritmos tratados en este trabajo es aproximarnos a dicha respuesta y mediante otros medios lograr la probabilidad de bit en error requerida.

Es por ello que existen otros tipos de igualadores no lineales que se basan en el criterio de la máxima verosimilitud para determinar que símbolo fue el transmitido [BSH94], sin embargo dado que estos métodos no forman parte del objetivo del presente trabajo se prefiere sólo dar la referencia en caso de que el lector este interesado en esta clase de algoritmos.

Esta clase de algoritmos reporta un desempeño superior al de los aquí tratados, sin embargo el precio que se paga es una carga computacional considerable, por lo que la investigación continua abierta en este fascinante tema.

# **Parte III**

## **Implementación de los Algoritmos**

## Capítulo 5

# Evaluación de los Algoritmos

En este capítulo se presentan las evaluaciones y resultados obtenidos para los algoritmos adaptables aplicados al problema de la igualación de canal. Debido a la gran riqueza de la teoría del filtrado adaptable y a la gran variedad de algoritmos existentes así como a sus múltiples variantes, no es posible analizar cada uno de los algoritmos así como sus respectivas variaciones y, mantener este trabajo dentro de un límite razonable en cuanto a su extensión: es por ello que se ha decidido analizar sólo algunos algoritmos con el fin de analizar su comportamiento. Los algoritmos seleccionados para este fin son los siguientes: "Normalized Least Mean Squares", "Fast Least Squares", "Lattice Least Squares" y el algoritmo "The Simple Hyperstable Recursive Filter" (SARF).

El objetivo de estas evaluaciones lo constituye el análisis del comportamiento así como del desempeño de estos algoritmos para distintos tipos de canales, los cuales a su vez pueden ser clasificados de acuerdo a su fase [PRLN92]. Supongamos que tenemos un canal cuya función de transferencia es racional, esto es:

$$H(z) = \frac{B(z)}{A(z)} \quad (5.1)$$

donde  $A(z)$  y  $B(z)$  son polinomios en  $z$  y, el grado de  $A(z)$ ,  $n$ , es mayor que el grado de  $B(z)$ ,  $m$ . De esta manera tenemos un canal con respuesta impulsional infinita (IIR). Entonces bajo la suposición de que el sistema dado por  $H(z)$  es estable y causal, podemos decir que se trata de un sistema de fase mínima si todos sus ceros no triviales caen dentro del círculo unitario, por otro lado si todos los ceros caen fuera del círculo unitario se dice que el sistema es de fase máxima y, finalmente tenemos el caso de que algunos ceros caigan fuera del círculo unitario mientras que otros caigan dentro, en éste caso se dice que tenemos un sistema de fase mixta ó no mínima. Dentro de éste contexto, cabe señalar que el objetivo de la igualación de canal es encontrar el sistema inverso,  $H^{-1}(z)$ , del canal de tal manera que:

$$H(z)H(z)^{-1} = 1 \quad (5.2)$$

Siempre tendremos problemas con los canales de fase no mínima, ya que éstos no son invertibles. Una vez sea dicho de paso que cualquier sistema de fase no mínima

se puede expresar como un sistema de fase mínima en cascada con un sistema paso todas [PM96].

De la discusión anterior resulta obvio que los igualadores de estructura FIR presentarán problemas al intentar igualar un canal de fase no mínima, pues en realidad el igualador ideal será inestable, por lo que es de esperar que este tipo de igualadores necesite de un gran número de coeficientes para producir resultados aceptables. Por otro lado el uso de un "Feedback equalizer" para el caso de canales de fase no mínima no es posible ya que el inverso del canal es inestable, por lo que sólo se utilizan este tipo de algoritmos cuando se conoce con certeza que el canal es de fase mínima [CG85]. Considérese el caso de la propagación multitrayectoria: en el caso de que las señales que siguen trayectorias diferentes de la línea de vista sean más débiles que la señal que se propaga siguiendo la línea de vista, el canal es de fase mínima; por otro lado si por alguna razón una o más de las señales que siguen trayectorias diferentes a la línea de vista tiene una SNR mayor que la señal que sigue la línea de vista, la función de transferencia del canal será de fase no mínima. Por lo que en éste último caso no se podrá aplicar un esquema de igualación con retroalimentación.

## 5.1 Descripción de las Evaluaciones

En esta sección se describirá bajo que condiciones fueron realizadas las evaluaciones. Con el objeto de estandarizar las pruebas realizadas y, de esta manera evitar cualquier sesgo en la evaluación de los algoritmos aquí tratados, todos los algoritmos fueron implantados en MATLAB y, se utilizó una señal PAM de cuatro niveles generada a partir de ruido blanco gaussiano de variancia unitaria y media cero. Los canales utilizados para las distintas evaluaciones, se seleccionaron de la literatura de la igualación de canal. Sin embargo cabe mencionar que no todos los canales fueron utilizados en todas las evaluaciones, pues en el caso de los algoritmos de filtrado adaptable que utilizan retroalimentación (en este caso el algoritmo SARF), sólo pueden utilizarse para igualar canales de fase mínima; por lo tanto no se efectuaron evaluaciones del desempeño del algoritmo SARF para canales de fase no mínima. En cuanto al resto de los algoritmos aquí tratados se efectuaron las evaluaciones correspondientes tanto para el caso de canales de fase mínima, como para el caso de canales de fase no mínima, dado que estos algoritmos no presentan problemas de inestabilidad ya que adaptan los coeficientes de un filtro FIR.

### 5.1.1 Comentarios Acerca de las Evaluaciones

En esta sección, se hace un énfasis especial en que los resultados aquí presentados dependen en cierta medida del juicio que se emplee para determinar si la calidad de un resultado determinado, dado que no se utilizó un criterio fundamentado en algún índice de desempeño del sistema de comunicaciones como lo es la probabilidad de bit en error. En cambio para determinar la convergencia del algoritmo se utilizaron

varios criterios: el primero consiste en observar la magnitud al cuadrado del vector de coeficientes del filtro adaptivo, si esta se mantiene más o menos constante a través del tiempo entonces se dice que el algoritmo ha convergido de acuerdo a este criterio; el segundo criterio consiste en que la convolución de las respuestas al impulso del canal y del igualador de como resultado una secuencia de muestras igual a cero excepto para un sólo valor del tiempo discreto en el cuál la muestra correspondiente sea igual a la unidad, esto nos indicará a un nivel cualitativo que gran parte de la interferencia intersímbolos ha sido cancelada por medio del igualador. Finalmente el tercer criterio consiste en comparar la respuestas en frecuencia tanto del canal como del igualador. Es claro que a la frecuencia donde la respuesta en frecuencia del canal presente una resonancia o "pico", la respuesta en frecuencia del igualador debe presentar una "antiresonancia" ó "valle" y viceversa.

En caso de que estos tres criterios no coincidieran del todo se analizará la correlación cruzada de señal de entrada al canal con la señal de salida del igualador, la cual debe ser igual a la autocorrelación de la señal de entrada al canal. De esta manera, se puede comprender de mejor manera el desempeño del igualador: aunque cabe señalar que en este último criterio, la semejanza de dichas correlaciones no garantiza de ningún modo la fidelidad de la señal de salida del igualador [LTCRJ80].

Finalmente cabe señalar que el número de operaciones flotantes se midió mediante la función flops de MATLAB, la cuál regresa el número total de operaciones flotantes que se han realizado hasta el momento de su llamada. Esta función no siempre cuenta el número de operaciones flotantes pero de acuerdo con el manual del MATLAB sí cuenta las operaciones más importantes. Las adiciones y subtracciones se cuenta como una operación flotante si los operandos son números reales, en caso de ser complejos se cuentan como dos operaciones flotantes. Las multiplicaciones y divisiones se cuentan como una operación flotante si el resultado es real, en caso contrario se cuentan como seis operaciones flotantes.

De esta manera se obtuvo el número de operaciones flotantes, el cuál se dividió entre el número de muestras de la señal de entrada al igualador para obtener el número de operaciones flotantes por muestra de salida.

### 5.1.2 Canales Utilizados en las Evaluaciones

Los canales utilizados en las evaluaciones se tomaron de artículos en dónde éstos fueron utilizados para validar los métodos ahí propuestos. De esta manera los canales utilizados son los siguientes:

$$H_1(z) = [1 - 1.4001z^{-1} + 0.9801z^{-2}] \quad (5.3)$$

Este canal tiene sus ceros en  $0.99e^{\pm j44.9988^\circ}$ , por lo tanto se trata de un canal de fase mínima. Este canal fue tomado de [PF89]. El siguiente canal esta dado por:

$$H_2(z) = [1 - 1.1314z^{-1} + 0.6400z^{-2}] \quad (5.4)$$

Este canal tiene sus ceros en  $0.8e^{\pm j44.9985^\circ}$ , por lo tanto se trata de un canal de fase mínima. Este canal fue tomado de [PF89]. El siguiente canal esta dado por:

$$H_3(z) = \frac{1 + 1.5000z^{-1}}{1 - 1.64453z^{-1} + 0.740818z^{-2}} \quad (5.5)$$

Los polos de este canal estan dados por  $0.86071e^{\pm j17.188^\circ}$ . Este canal tiene un cero en -1.5 por lo tanto es de fase máxima. Este canal fue tomado de [LD94]. El siguiente canal esta dado por:

$$H_4(z) = \frac{z^{-1}(1 - 2.02620z^{-1} + 1.01389z^{-2})}{1 - 2.96999z^{-1} + 3.39000z^{-2} - 1.75824z^{-3} + 0.349938z^{-4}} \quad (5.6)$$

El cual es una versión sobre muestreada del siguiente canal analógico:

$$H(S) = \frac{(S + 100)(S - 100)}{(S^2 + 1800S + 2250000)(S^2 + 2400S + 2250000)} \quad (5.7)$$

Através de este canal comúnmente se transmiten señales moduladas en PAM con una duración de símbolo igual a  $T = 10^2 3s$ . Los polos de la versión discreta de este canal están dados por  $0.7988e^{\pm j17.1702^\circ}$  y  $0.7405e^{\pm j12.924^\circ}$ ; los ceros de este canal estan dados por 1.1248 y 0.9011, por lo tanto se trata de un canal de fase no mínima. Este canal fue tomado de [LD94]. El siguiente canal esta dado por:

$$H_5(z) = \frac{1 + 0.6000z^{-1} - 0.3937z^{-3}}{1 - 0.6561z^{-4}} \quad (5.8)$$

Este canal tiene cuatro polos en  $0.9000e^{\pm j45^\circ}$  y en  $0.9000e^{\pm j135^\circ}$ ; los ceros se localizan en  $0.8252e^{\pm j134.46^\circ}$  y en 0.5781, por lo tanto este canal es de fase mínima. Este canal se tomo de [LD94]. El siguiente canal esta dado por:

$$H_6(z) = [0.01 - 0.05z^{-1} + 0.07z^{-2} - 0.21z^{-3} - 0.5z^{-4} + 0.72z^{-5} + 0.36z^{-6} + 0.21z^{-7} + 0.03z^{-8} + 0.07z^{-9}] \quad (5.9)$$

Este es un canal de fase no mínima y fue tomado de [PRLN92]. El siguiente canal esta dado por:

$$H_7(z) = [0.407 - 0.815z^{-1} + 0.407z^{-2}] \quad (5.10)$$

Este canal es de fase no mínima y fue tomado de [Pro95]. El siguiente canal esta dado por:

$$H_8(z) = [0.227 - 0.460z^{-1} + 0.688z^{-2} - 0.460z^{-3} - 0.227z^{-4}] \quad (5.11)$$

Este canal es de fase no mínima y fue tomado de [Pro95].

La siguiente función de transferencia se utilizo en la calibración de los algoritmos tratados en este capítulo. Esta fue tomada de [LTCRJ80]:

$$H_9(z) = \frac{0.057}{1 - 1.645z^{-1} + 0.9025z^{-2}} \quad (5.12)$$

Los polos de  $H_9(z)$  estan dados por  $0.95e^{\pm j30^\circ}$ .

## 5.2 Presentación de Resultados

### 5.2.1 Evaluaciones con el Algoritmo SARF

Durante el proceso de validación de este algoritmo, se hicieron algunas pruebas tanto para la identificación de la función de transferencia del canal como para la igualación del mismo.

Una de las primeras cuestiones que surge está directamente relacionada con la estructura IIR del algoritmo, por lo que cabe señalar que es recomendable aplicar éste algoritmo sólo cuando el igualador ideal sea estable y además tenga una función de respuesta al impulso infinita, lo que nos conduce a que el canal sea de fase mínima. Si el canal no es de fase mínima cualquier algoritmo de filtrado adaptable IIR es inaplicable pues los polos del filtro adaptivo tendrán forzosamente que abandonar el círculo unitario antes de que el algoritmo converga. Por otro lado considere el caso de un canal cuya función de transferencia sea todo polo, entonces la función de transferencia del igualador estará dada por un filtro todo cero; en este caso la estructura IIR del algoritmo SARF se reduce a una estructura FIR; por lo tanto, para este caso, el algoritmo SARF puede ser visto como un algoritmo FIR basado en el gradiente estocástico. En este último caso, el algoritmo SARF no presenta los problemas de inestabilidad y de su posible convergencia a un mínimo local, de aquí que se pueda utilizar la teoría del filtrado adaptable para filtros FIR, la cual está mucho más desarrollada que su contraparte para filtros IIR.

Como se mencionó anteriormente uno de los principales problemas de los algoritmos de filtrado adaptable IIR basados en el método del error de salida es su posible convergencia a un mínimo local. Se ha demostrado la no existencia de mínimos locales cuando el algoritmo es configurado para operar en su modo de identificación de sistemas bajo las siguientes condiciones [S75, S882]:

- El filtro adaptivo tiene el suficiente orden para modelar al sistema desconocido.
- La entrada al sistema a modelar es ruido blanco.
- El orden del numerador del filtro adaptivo excede al orden del denominador del sistema desconocido.

Cabe mencionar que estos resultados han sido derivados para el caso de la identificación de sistemas, en lo que respecta al caso del filtrado no existe ningún resultado similar sino que sólo se especula con la existencia del mismo [Shy89]. No cabe la menor duda de que este resultado es importante para la problemática de la identificación de sistemas, sin embargo sólo nos indica cuando no existe un mínimo local en ciertas condiciones muy restrictivas las cuales presuponen cierto conocimiento a priori del sistema a identificar.

Quizas para el caso de la identificación de sistemas estas condiciones no sean demasiado restrictivas pero supóngase que se hallase un resultado similar para el problema del filtrado; esto no daría una solución definitiva a varios problemas:

pensemos en la igualación de canal para un sistema de comunicaciones móviles en donde las condiciones de la transmisión no son tan predecibles y, además como se sabe se presenta el fenómeno de la propagación multitrayectoria. En este caso como se menciono anteriormente existe la posibilidad de que alguna de las señales que no se propagan a través de la línea de vista tenga una SNR mayor que la señal que se propaga a través de la línea de vista lo cual ocasionará la introducción de un cero fuera del círculo unitario en la función de transferencia del canal; por lo que además de variar el orden de la función de transferencia, su característica relativa a la fase ha cambiado de una fase mínima a una fase no mínima para la cual la aplicación de un algoritmo adaptable con estructura IIR no es posible. De aquí que al contrario del problema de la identificación de sistemas donde es bastante razonable suponer que el orden del sistema es invariante en el tiempo, en el caso del filtrado esto no es siempre posible. De aquí que en un caso como el hasta aquí tratado habría que proceder de cualquiera de las dos siguientes maneras: 1) aumentar el orden del filtro adaptable de tal manera que siempre sea menor al del canal a igualar, lo que traería como consecuencia un aumento inevitable en el número de operaciones a realizar en el proceso de la igualación de canal, lo que se traduciría inevitablemente en un aumento del tiempo necesario tanto para comenzar la transmisión de datos como en el proceso de demodulación de los mismos en el receptor; produciéndose de esta manera un retardo en la comunicación, lo cual no es de ninguna manera aceptable para un enlace de comunicaciones en tiempo real a altas tasas de transmisión; 2) por otro lado podríamos estimar el orden del canal, de tal manera que el orden del filtro sea del mínimo requerido. Este método también traerá las mismas consecuencias citadas para el caso anterior, pues el algoritmo de estimación del orden del canal también nos conduce a un aumento del tiempo requerido para comenzar la transmisión de datos; además se requiere de un monitoreo del orden del canal para compensar las variaciones del mismo lo cual incrementará la carga computacional del receptor ocasionando los problemas citados en este mismo párrafo.

### 5.2.1.1 Evaluaciones con el Algoritmo SARF para Identificar el Canal

En el modo de identificación de sistemas se observó que no siempre existen mínimos locales, sin embargo en otros casos el algoritmo converge a un mínimo local por lo cual fue necesario incrementar el orden del filtro, lo cual trae consigo los efectos anteriormente citados.

Para el caso de identificar la función  $H_9(z)$  se observo que el algoritmo converge a la solución correcta sin necesidad de incrementar el orden del filtro, por lo que en este caso se especula que no existen mínimos locales en la superficie de la función de costo. A pesar de éste hecho se decidió aumentar el orden del filtro a pesar de las consecuencias anteriormente citadas con el objeto de estudiar el comportamiento del algoritmo. Se observo que al incrementar el orden del filtro, el algoritmo converge más rápido (vease la tabla 5.1 en donde se muestran los resultados de identificar la función  $H_9$  con el algoritmo SARF). Los pasos de adaptación utilizados para realizar

estas evaluaciones son:  $\rho = 0.001$ ,  $\mu = 0.01$  y  $c_1 = -0.8$ .

Nótese que en estas evaluaciones el número de polos siempre excede en dos al número de ceros y, además cuando el orden del filtro es mayor que el orden del canal se observa que los polos y ceros se tienden a acomodar de tal manera que los polos sobrantes se anulan con los ceros, así los dos polos restantes corresponden a los del sistema que se desea identificar. En las gráficas correspondientes a la migración de polos se puede observar que los polos describen trayectorias rectas, de tal manera que los dos polos que no se cancelan con los ceros tienden en línea recta hacia los polos del canal. Para corroborar esto último se presenta la gráfica de la figura 5.1.

Orden del Modelo	Convergencia en Iteraciones	Número de Operaciones Flotantes por Iteración
2 polos	90 000	73.86
4 polos 2 ceros	8 000	2 136.11
6 polos 4 ceros	3 000	7 115.18
8 polos 6 ceros	2 750	12 360.87

Tabla 5.1: Resultados obtenidos de aplicar el algoritmo SARF para identificar  $H_0$

Cabe señalar que este comportamiento solo se observa en el caso de que se dé la cancelación de polos y ceros anteriormente descrita, en caso contrario los polos y ceros del sistema no seguirán este comportamiento. De cualquier manera según la literatura no es muy conveniente analizar el comportamiento del algoritmo SARF mediante este tipo de gráficas [LTCRJ80], sin embargo en este caso nos ayuda a visualizar el por qué el algoritmo converge más rápido cuando se aumenta el orden del filtro. Si se compara este comportamiento de los polos con el obtenido para el

filtro del orden estrictamente necesario para identificar la función  $H_9(z)$ , se observa de la gráfica de migración de polos para este último caso, que los polos no siguen una trayectoria recta hacia los polos del canal; por lo que resulta lógico suponer que el algoritmo tardará más en converger para este último caso.

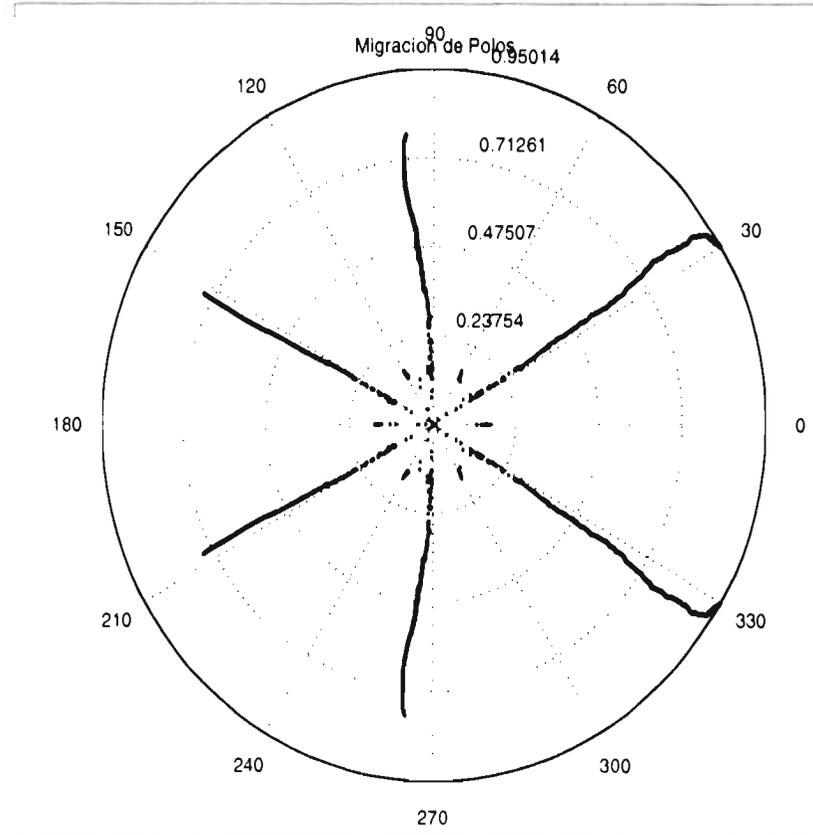


Figura 5.1: Migración de polos para el algoritmo SARF. Se utilizó un filtro de seis polos y cuatro ceros.

Para corroborar esto último se presentan las gráficas de las figuras 5.1 y 5.2, donde se puede apreciar la migración de polos para el algoritmo SARF para el caso de utilizar un filtro adaptivo de 4 polos y dos ceros y, para el caso de utilizar un filtro del mismo orden que la función de transferencia del canal que se desea identificar, respectivamente.

Cabe señalar que este comportamiento del algoritmo SARF sólo se observó para el caso en el que se dio la cancelación de polos y ceros anteriormente descrita,

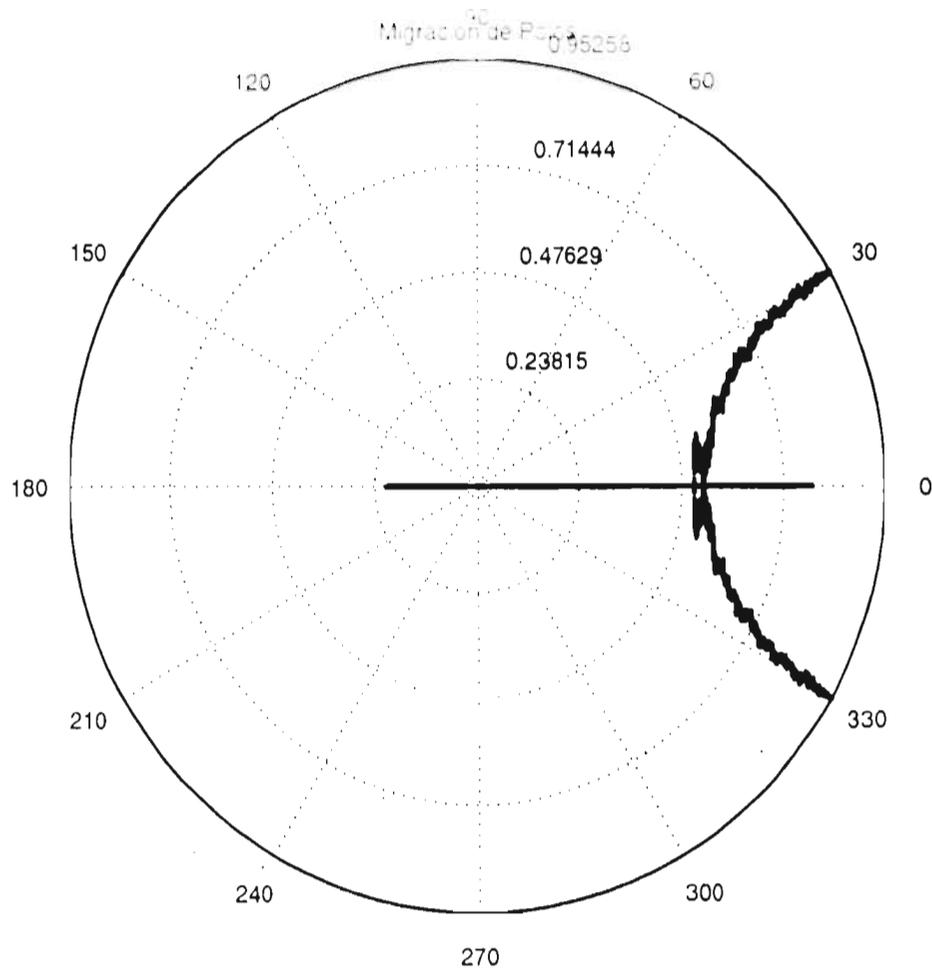


Figura 5.2: Migración de polos para el algoritmo SARF.  
Se utilizo un filtro de dos polos únicamente.

La siguiente prueba que se realizó con el algoritmo SARF, en su modo de identificación de sistemas consistió en la variación del filtrado que se lleva a cabo sobre la señal del error de salida. Como se menciono en el capítulo sobre filtrado adaptable IIR, este filtrado sobre la señal del error de salida no solamente restringe la región del círculo unitario donde se cumple la condición SPR sino que también afecta la convergencia del algoritmo. Para esta prueba se considero un filtro de un sólo coeficiente. Se utilizo un filtro adaptivo de cuatro polos y dos ceros. Los pasos de adaptación utilizados fueron de  $\mu = 0.01$  y una  $\rho$  de 0.001. Los resultados obtenidos en cuanto a la variación de este coeficiente muestran una tendencia en el aumento de la rapidez con que converge el algoritmo al aumentar el valor del coeficiente del filtro de promedio móvil que se aplico a la señal de salida del error (vease la tabla 5.2).

Parámetro del Filtro $c_1$	Convergencia en Iteraciones
-1.0	8 000
0.0	6 500
0.8	5 000
1.0	5 500
1.5	4 500

Tabla 5.2: Resultados obtenidos de variar el parámetro  $c_1$  del filtrado sobre la señal del error de salida en el algoritmo SARF, utilizado a su vez para identificar  $H_3$

Finalmente se harán algunos comentarios sobre la rapidez con la que el algoritmo converge y sobre la complejidad computacional del mismo. Como se menciona en el capítulo correspondiente, el algoritmo SARF es una versión simplificada del algoritmo HARF [LTCRJ80]. Esta simplificación según la literatura es necesaria para poder aplicar el algoritmo HARF a problemas en tiempo real, debido a la excesiva carga computacional que este impone; sin embargo se debe pagar algún precio al realizar la simplificación del algoritmo HARF, que en su momento nos condujo al algoritmo SARF: este precio consiste en que la convergencia del algoritmo SARF al contrario que la del algoritmo HARF no está totalmente garantizada para cualquier valor de constantes de adaptación, ni siquiera se ha demostrado de una manera rigurosa la convergencia del algoritmo SARF sino que se especula con el hecho de que si los pasos de adaptación son lo suficientemente pequeños el algoritmo convergerá. Por lo tanto éste hecho trae como consecuencia que el algoritmo SARF no se puede aplicar en problemas en los que se requiera de una adaptación rápida.

Experimentalmente se hicieron pruebas para obtener un límite máximo para los pasos de adaptación, estos límites resultaron en una  $\mu$  de 0.01 y en una  $\rho$  de

0.001, si se trata de aumentar los pasos de adaptación el algoritmo no converge. Por otro lado al disminuir estos parámetros la convergencia del algoritmo es demasiado lenta.

En cuanto a la carga computacional del algoritmo SARF como la de cualquier algoritmo adaptable IIR, se debe en gran parte a la necesidad de checar la estabilidad del filtro adaptivo en cada iteración. En este trabajo se comprobó que esto se debe realizar a pesar de que en la literatura [CG85, LTCRJ80, Shy89], se argumente que para pasos "pequeños de adaptación" esto no es necesario. En caso de que no se verifique la estabilidad del filtro éste no converge a la solución óptima debido a que los polos cercanos al círculo unitario provocan que en cualquier iteración el filtro se vuelva inestable por la simple y sencilla razón de que los polos abandonen el círculo unitario. Esto debido a que se utilizan estimadores "ruidosos" para evaluar los valores esperados, ya que no se conocen con exactitud las densidades de probabilidad del ruido que afecta al canal ni de la señal que se propaga a través de este. Esta falta de conocimiento de dichas densidades de probabilidad trae como consecuencia que los estimadores presenten "ruido" que se manifiesta a través de la oscilación del vector de coeficientes del filtro adaptivo alrededor de su valor óptimo, lo cual puede traer como consecuencia que los polos abandonen el círculo unitario.

Por otro lado, en la literatura se sugiere la disminución de los pasos de adaptación cada vez que se detecte que el algoritmo es inestable (mediante algún algoritmo como la prueba de estabilidad de Shür-Cohn), como una alternativa a proyectar las raíces (polos) dentro del círculo unitario [FJ86, CRJ84]. Sin embargo, esta práctica no es muy recomendable pues en algunos casos se observó que los pasos de adaptación toman un valor prácticamente igual a cero sin que el algoritmo haya convergido. Por lo tanto, el problema de la estabilidad de los filtros adaptivos IIR constituye una problemática abierta, ya que por un lado el cálculo directo de las raíces consume demasiados recursos computacionales; mientras que por otro lado la reducción de los pasos de adaptación no funciona en todos los casos.

### 5.2.1.2 Evaluaciones con el Algoritmo SARF para la Igualación de Canal

Una vez calibrado el algoritmo para la identificación de la función de transferencia del canal, se efectuaron evaluaciones para la igualación de canales de fase no mínima, ya que como recordará el lector, los algoritmos de filtrado adaptable IIR no se pueden aplicar a canales de fase no mínima. De esta manera sólo se realizaron pruebas para los canales  $H_1$ ,  $H_2$  y  $H_5$ . Cabe mencionar que para igualar los canales  $H_1$  y  $H_2$ , el algoritmo SARF tuvo un desempeño excelente con una convergencia aproximada en mil iteraciones tanto para  $H_1$  como para  $H_2$ . Sin embargo fue necesario elevar el orden de los filtros utilizados pues con el orden justo el algoritmo convergió a un mínimo local en los dos casos hasta aquí tratados. Aunque el elevar el orden del filtro es beneficioso en cuanto al aumento de la rapidez con que el algoritmo converge, el precio que se paga es el aumento en el número de operaciones flotantes por muestra de salida (véase la tabla 5.2). Por otro lado, para el canal  $H_5$

Canal	Orden del Moelo	Convergencia en Iteraciones	Número de Operaciones Flotantes por Iteración
$H_1(z)$	2 polos	10 000	73.86
$H_1(z)$	6 polos 4 ceros	1 000	7 115.18
$H_1(z)$	2 polos	10 000	73.86
$H_2(z)$	6 polos 4 ceros	1 000	7 115.18

Tabla 5.3: Resultados obtenidos de aplicar el algoritmo SARF para igualar los canales  $H_1$ ,  $H_2$

el desempeño del algoritmo no es tan bueno como en los casos anteriores, pues en las evaluaciones realizadas a pesar de que la convolución de la respuesta al impulso del canal con la respuesta al impulso del igualador es casi igual al impulso discreto salvo en algunas muestras menores a 0.1: la respuesta en frecuencia del igualador no es exactamente igual a la inversa del canal. Esta dificultad en igualar el canal puede deberse a la configuración de los polos y ceros del mismo. Si se observa con más detenimiento esta configuración se notará de inmediato que el canal tiene un par de polos complejos conjugados en  $0.9000e^{\pm j135^\circ}$  y un par de ceros complejos conjugados en  $0.8252e^{\pm j134.46^\circ}$ ; de aquí la suposición de que esta cercanía entre los polos y ceros del canal de alguna manera pueda afectar el desempeño del igualador debido a errores numéricos, los cuales puedan provocar una cancelación no deseable de polos y ceros.

Los resultados de las evaluaciones efectuadas con el canal  $H_5(z)$  se pueden resumir de la siguiente manera:

- La convolución de la respuesta al impulso del canal con la del igualador casi da un pulso, sin embargo la respuesta en frecuencia del igualador no es muy

buena (ver figura 5.3).

De aquí que el algoritmo no iguale el canal de la mejor forma, lo que queda plenamente corroborado de las figuras 5.4, 5.5 y 5.6, donde se muestra la correlación cruzada de la señal de entrada al canal con la señal de salida del igualador; la autocorrelación de la señal de entrada al canal y; la correlación cruzada de la señal de entrada al canal con la señal de salida del canal.

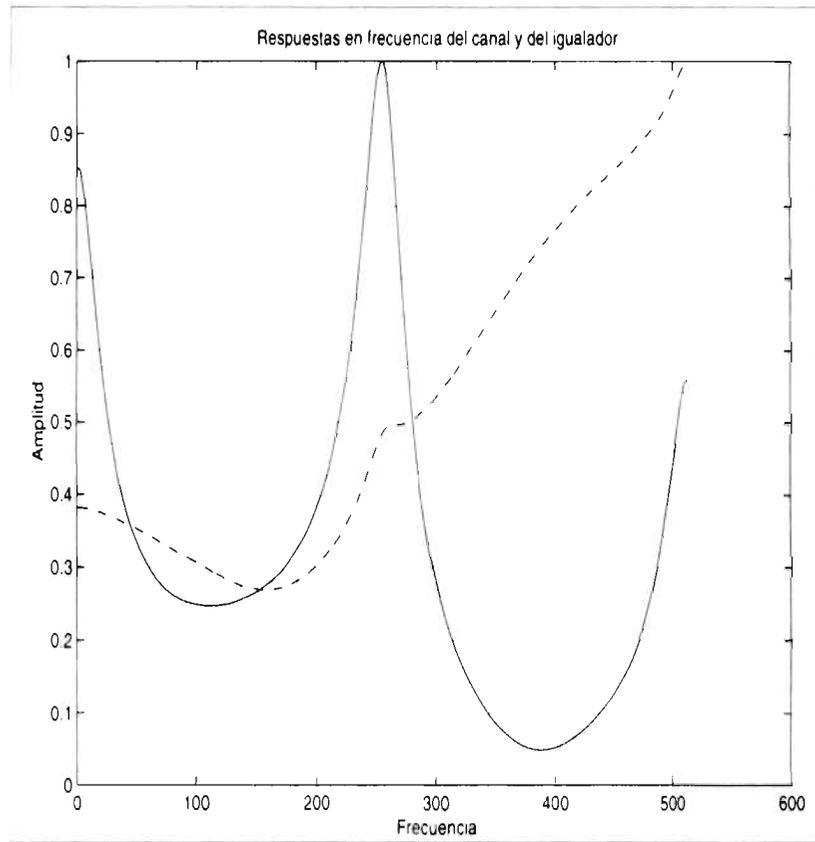


Figura 5.3: Respuestas en frecuencia del canal y del igualador como resultado de aplicar el algoritmo SARF para igualar el canal  $H_5(z)$ . Línea continua (Canal).

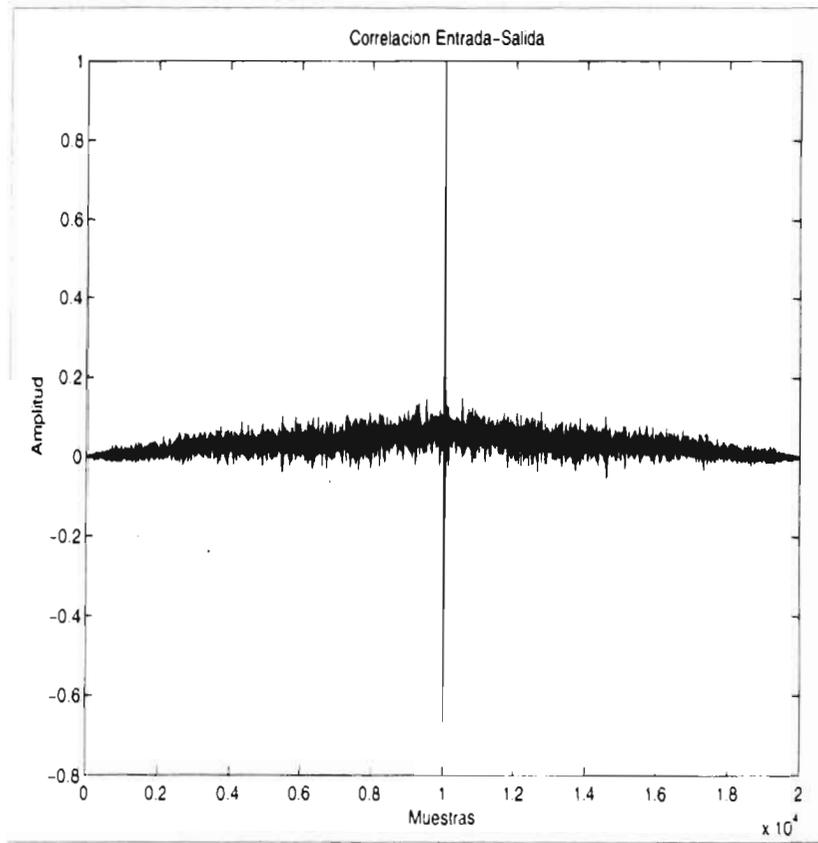


Figura 5.4: Correlación cruzada de la señal de entrada al canal con la señal de salida del mismo como resultado de aplicar el algoritmo SARF para igualar el canal  $H_5(z)$ .

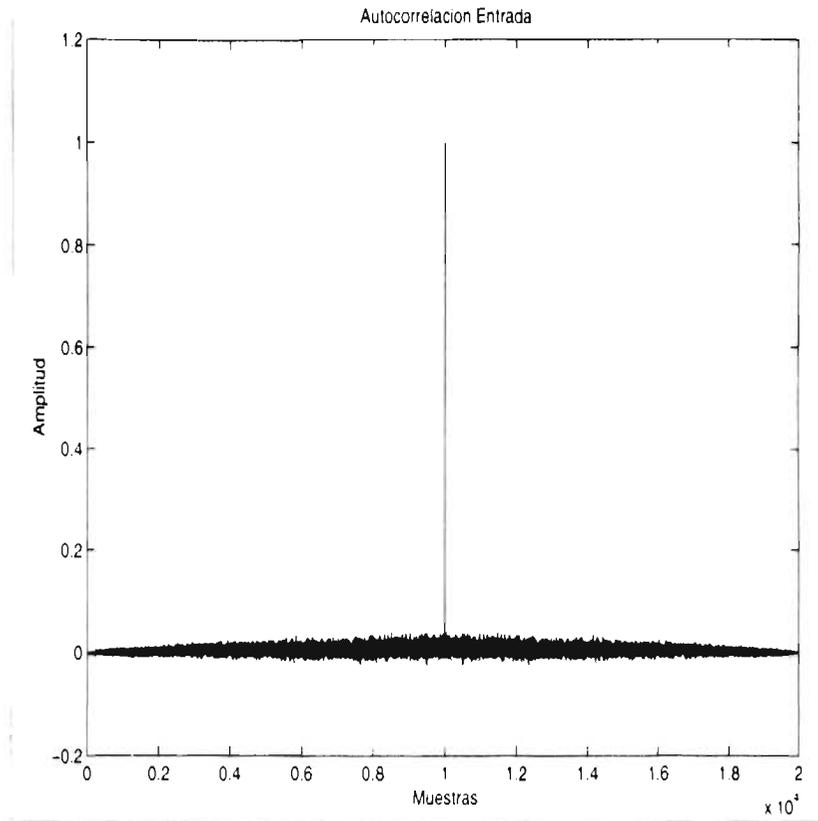


Figura 5.5: Autocorrelación de la señal de entrada al canal como resultado de aplicar el algoritmo SARF para igualar el canal  $H_5(z)$ .

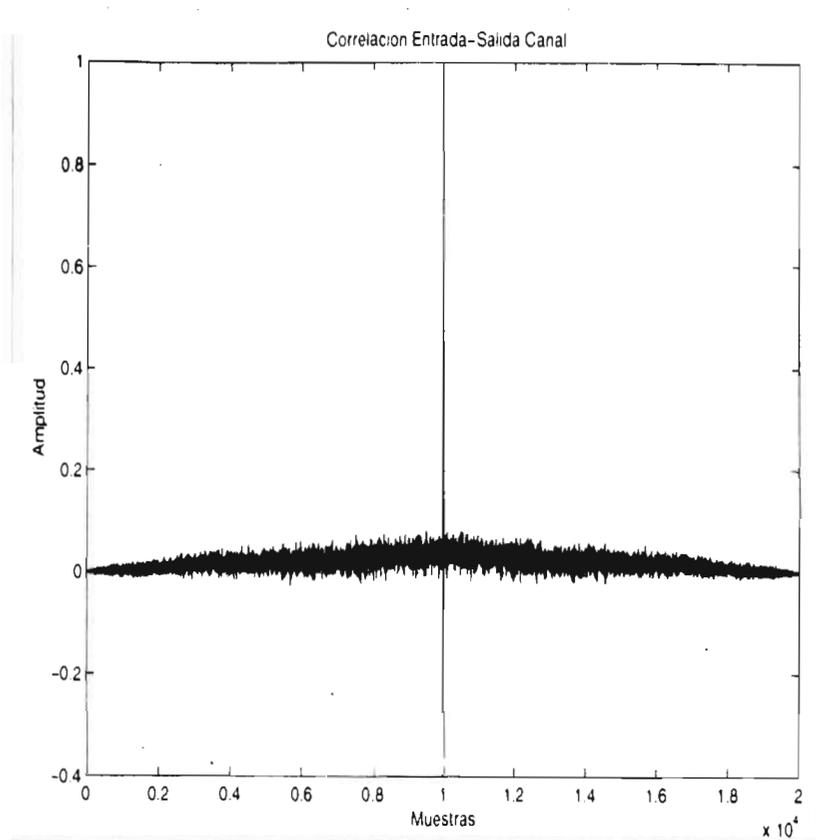


Figura 5.6: Correlación de la señal de entrada al canal con la señal de salida del canal.

### 5.2.2 Evaluaciones con el Algoritmo "Normalized Least Mean Squares (NLMS)"

En esta sección se presentarán los resultados obtenidos al aplicar el algoritmo NLMS a los siguientes canales:  $H_1$ ,  $H_2$ ,  $H_3$ ,  $H_4$ ,  $H_5$ ,  $H_6$ ,  $H_7$  y  $H_8$ . Nótese que  $H_1$ ,  $H_2$  y  $H_5$  son canales de fase mínima, mientras que el resto son de fase no mínima.

En general en todos estos casos el igualador necesitara de muchos coeficientes para producir resultados muy buenos, pues el igualador ideal tiene una respuesta impulsional infinita, la cual se esta aproximando mediante un filtro FIR. Además de esta dificultad el igualador se encontrará con la difícil tarea de igualar el canal cuando éste se caracterize por una fase no mínima pues en este caso además de

que el igualador ideal tiene una respuesta impulsional infinita, éste es inestable: por lo tanto el algoritmo de igualación solo dara una aproximación al igualador ideal. Esta aproximación será mejor mientras más grande sea el soporte del filtro

Orden del Filtro	Número de Operaciones Flotantes por Muestra de Salida
6	136.22
8	212.17
10	304.12
12	412.07
14	536.02
16	675.97
18	831.92
20	1003.87
22	1191.82
24	1395.77
26	1615.72
28	1851.67
30	2103.62
40	3603.37
50	5503.12
70	10502.62
100	21001.87

Tabla 5.1: Número de operaciones flotantes por muestra de salida para el algoritmo NLMS

adaptivo, sin embargo esto traera como consecuencia el incremento en el número de operaciones flotantes, lo que se traducirá en un mayor tiempo de procesamiento lo cual afecta directamente a la posibilidad de establecer un enlace de comunicaciones en tiempo real. Además recuerde la definición de un algoritmo rápido que dice que un algoritmo puede clasificarse como tal si el número de operaciones a efectuar crece de una manera lineal con respecto al número de parámetros a ajustar [Hay91]; dado que el algoritmo NLMS no es un algoritmo rápido se experimentará un crecimiento no lineal del número de operaciones con respecto al incremento en el orden del filtro adaptivo (vease la tabla 5.3).

### 5.2.2.1 Canales de Fase Mínima

**Canal  $H_1(z)$**  En esta sección se compara el desempeño del algoritmo nlms para los canales de fase mínima, para lo cual se comenzara comentando los resultados obtenidos para el canal  $H_1(z)$ . Cabe señalar que dado que el igualador ideal para este tipo de canal es de respuesta impulsional infinita se requirio de un filtro de un orden de al menos 22 para producir un resultado bueno; sin embargo aun para este orden la convolución de la respuesta al impulso del canal con la del igualador no da

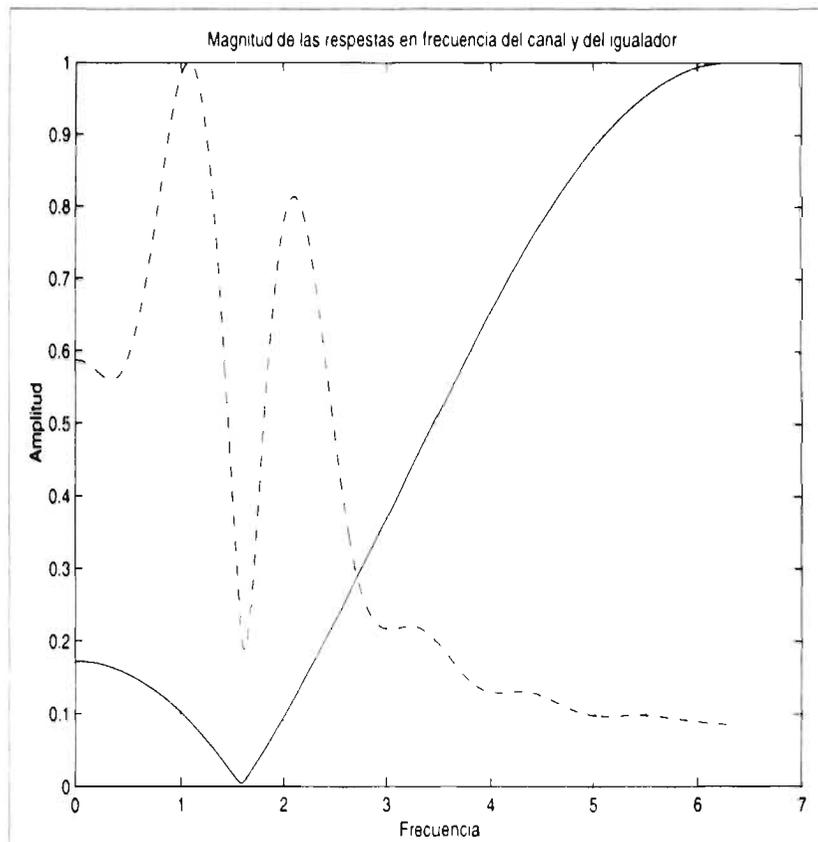


Figura 5.7: Respuesta en frecuencia del igualador y del canal. Línea continua (Canal).

una sola muestra sino que se puede apreciar que existe una pequeña interferencia intersímbolos por lo cual se elevó el orden del filtro adaptivo. A pesar de este incremento en el orden del filtro adaptivo, las gráficas siempre muestran que siempre existirá una pequeña interferencia intersímbolos dado que la mejoría es muy gradual

y, tal vez dependiendo de las especificaciones del enlace de comunicaciones requerido, con un filtro de orden 22 sea suficiente para igualar el canal. Lo más significativo es que la antiresonancia que en baja frecuencia presenta el canal no se logra compensar ni aun con un filtro de orden 100 (veáse la fig. 5.7). Esto quiere decir que cualquier señal que se propage a través del canal a dicha frecuencia será atenuada. Por otro lado esto no se presentó cuando se igualó el canal con el algoritmo SARF: en este último caso se logró una igualación casi perfecta y se logró compensar totalmente el efecto descrito anteriormente. Con el objeto de dar una muestra de los resultados obtenidos en la figura 5.7 se muestra tanto la respuesta del canal como la respuesta del igualador, en las cuales se puede observar el problema hasta aquí descrito.

En cuanto a la convergencia del algoritmo NLMS cabe señalar que esta casi se mantiene constante frente a las variaciones del orden del filtro, pues las variaciones observadas sólo son de 400 iteraciones y, la convergencia en promedio se produce poco después de las 1100 iteraciones. Aquí cabe señalar que para acelerar la convergencia del algoritmo en las primeras iteraciones para estas pruebas se selecciono un paso de adaptación de 1.5, lo que producirá un rápido acercamiento a la solución óptima: después de las 1050 iteraciones se cambió el paso de adaptación a 0.09 con lo cuál se logra disminuir el error en exceso [WS85].

**Canal  $H_2$**  De nueva cuenta, el hecho de igualar un canal cuyo igualador ideal es todo polo, mediante un filtro adaptivo FIR implica que se requerirá de un gran número de coeficientes para lograr buenos resultados. Lo primero que debemos notar es que los polos de  $H_2$  y los polos de  $H_1$  solo varían en su magnitud. Dado que la magnitud de los polos de  $H_2$  es menor a la magnitud de los polos de  $H_1$  cabría esperar que el algoritmo NLMS pueda compensar de una mejor manera la antiresonancia del canal ubicada aproximadamente en  $\pi/4$  dado que en el caso aquí tratado no es tan pronunciada como en el caso anterior.

Aun cuando la antiresonancia del canal  $H_2$  no es tan pronunciada como la de  $H_1$  se requiere de al menos de un filtro de orden 11 para lograr un resultado aceptable en el sentido de que al realizar la convolución normalizada de las respuestas al impulso del canal y la del igualador se observan muestras de magnitud menor a 0.1: las cuales indican una ligera interferencia intersímbolos. Si se requiere de un mejor resultado se debe utilizar un filtro de orden 22, sin embargo como se observa de la tabla 5.4 esto traerá como consecuencia inevitable el aumento en un poco más del doble de las operaciones realizadas por iteración o por muestra de salida. Cualquier aumento en este índice de desempeño no es bueno para el caso de que sea requerido un enlace de comunicaciones en tiempo real a alta velocidad. Por otro lado se observo que la convergencia al igual que en el caso anterior no sufre grandes variaciones con respecto al orden del filtro empleado. La convergencia promedio es de 4700 iteraciones mientras que la máxima es de 5500 y la mínima es de 4500. Sin embargo, como la convergencia del algoritmo se estima de la gráfica de la magnitud de coeficientes del filtro adaptivo contra las iteraciones, el valor que se proporciona es resultado de una apreciación subjetiva. Por lo tanto se puede decir

que la convergencia del algoritmo NLMS para el canal  $H_2$  se produce a las 1700 iteraciones. En estas evaluaciones se uso de una  $\mu = 1.54$  para las primeras 1125 iteraciones y luego se cambio a 0.09 con el objeto de reducir el error en exceso.

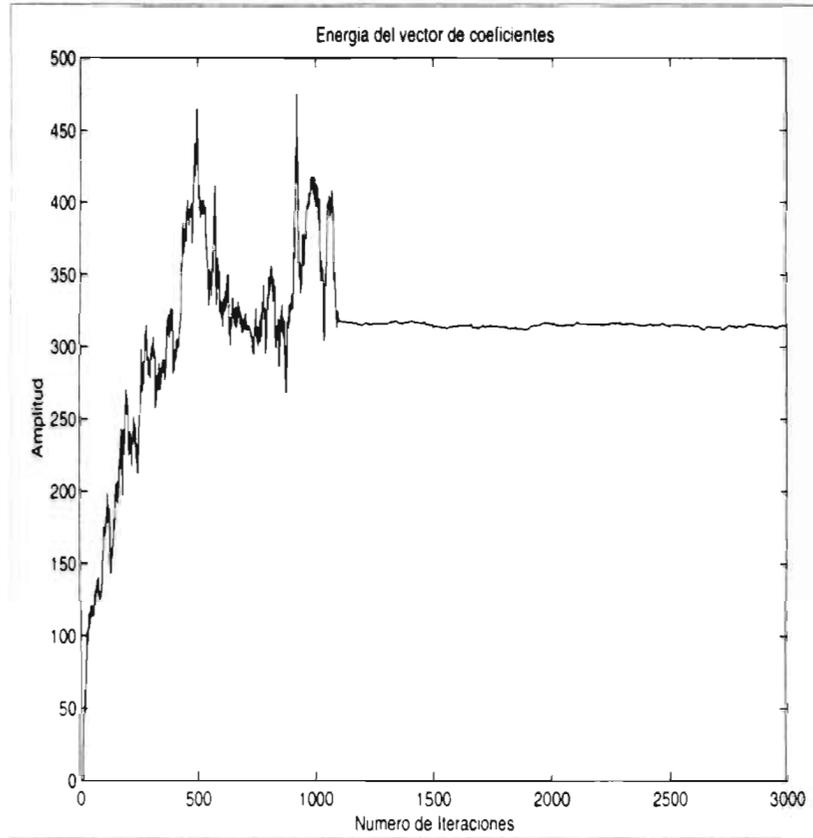


Figura 5.8: Energía del vector de coeficientes, Método nlms. Canal  $H_1(z)$ .

Nótese que la convergencia del algoritmo NLMS es más lenta que para el caso anterior, es decir que para  $H_1$ ; esto se debe en parte a la estrategia que se utilizó en la variación de la constante de adaptación; la cual consiste en realizar las primeras iteraciones con una  $\mu$  grande y, después se decrementa el parámetro de adaptación con el fin de disminuir el llamado error en exceso [WS85]. El número de iteraciones que se requieren para que el vector de coeficientes alcance su valor óptimo varía en función del valor de  $\mu$  inicial y del instante en el que se realice su cambio de valor. Para ilustrar este último punto vease las graficas de las figuras 5.8 y 5.9.

Finalmente se presenta la gráfica de la figura 5.10 donde se muestran tanto la respuesta en frecuencia del canal como la del igualador. Para obtener esta gráfica se aplicó el método nlms con un filtro adaptivo de orden 22, un parámetro de adaptación  $\mu = 1.5$ , el cual se cambió a 0.09 después de las 1050 iteraciones. En esta gráfica se puede apreciar que existe un ligero desfase entre la reso-

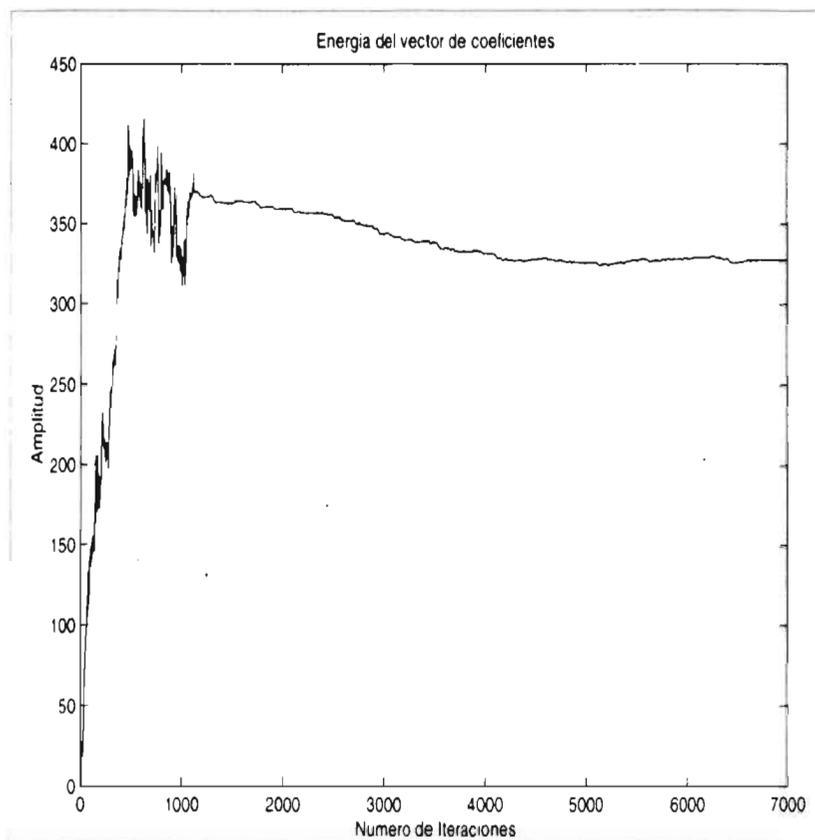


Figura 5.9: Energía del vector de coeficientes. Método nlms. Canal  $H_2(z)$ .

nancia del igualador y la antiresonancia del canal, lo cual se debe principalmente a que el filtro adaptivo tiene que aproximar la respuesta impulsional infinita del igualador ideal mediante un número finito de coeficientes. Por lo tanto, como se ha mencionado a lo largo del presente capítulo, esto traerá como consecuencia que sólo tengamos una aproximación al igualador ideal; esta aproximación se puede mejorar aumentando el número de coeficientes. Este aumento en el número de coeficientes no es muy conveniente desde el punto de vista del tiempo procesamiento requerido

para igualar el canal, ni mucho menos en el caso de un algoritmo no rápido, en el cual el aumento en el número de operaciones con respecto al número de coeficientes a ajustar no es lineal como se puede observar en la tabla 5.5. En esta tabla se nota que el número de operaciones por iteración o por muestra de salida es aproximadamente siete veces mayor cuando el orden del filtro es 22 que si se utiliza un filtro de orden diez.

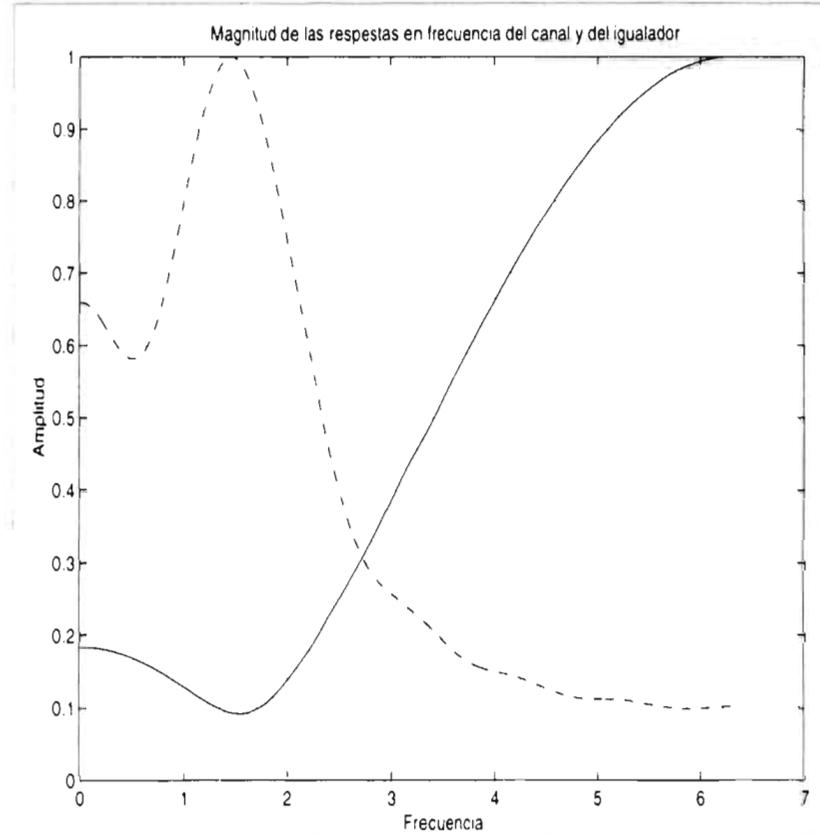


Figura 5.10: Respuesta en frecuencia del igualador y del canal. Algoritmo NLMS con  $\mu = 1.5$  y de 0.09 a partir de las 1050 iteraciones. Canal  $H_2(z)$ .

**Canal  $H_5(z)$**  A continuación se analiza el desempeño del algoritmo NLMS para igualar el canal  $H_5(z)$ . En este caso se requiere de un filtro de orden 30 para igualar el canal, aunque se puede considerar que con un orden igual a 10 la igualación es aceptable dado que la convolución de las respuestas al impulso del canal y del

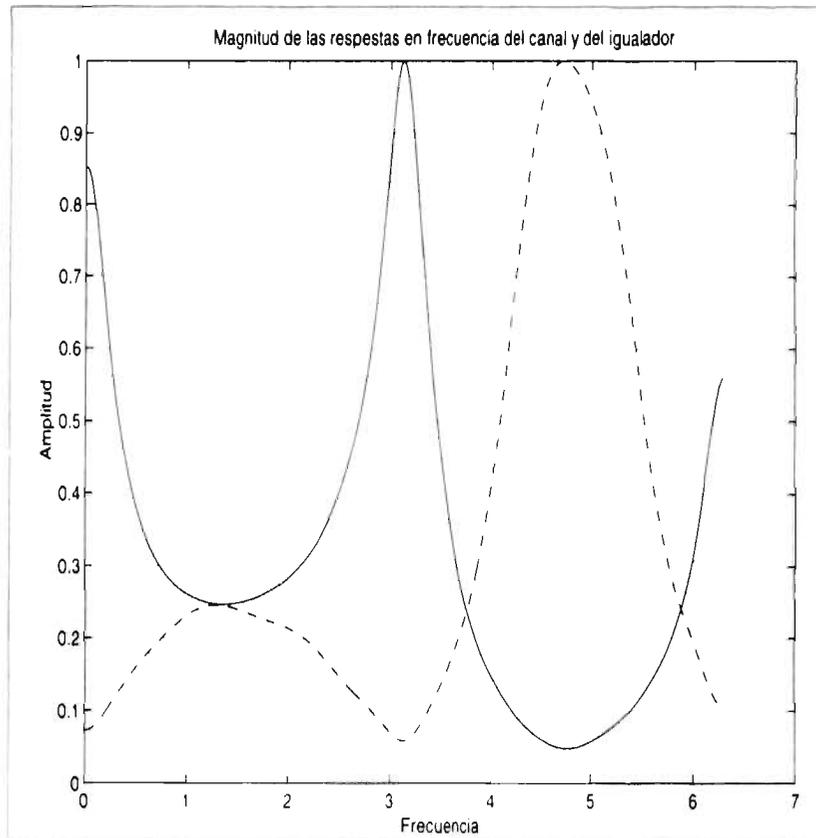


Figura 5.11: Respuestas en frecuencia tanto del canal como del igualador. Línea punteada (Canal). Algoritmo NLMS con  $\mu = 1.0$  y de 0.09 a un tercio del número total de iteraciones. Canal  $H_5(z)$ .

igualador presenta muestras secundarias menores a 0.25. Las evaluaciones efectuadas indican que para un orden del filtro entre 25 y 30 los resultados son óptimos en el sentido de que al elevar aun más el orden del filtro que se emplea para igualar el canal, se produce una mejoría muy gradual que no compensa el aumento del número de operaciones flotantes por muestra de salida. Por otro lado, la respuesta

en frecuencia del igualador salvo ligeros corrimientos en frecuencia de sus resonancias y antiresonancias constituye ya una muy buena aproximación a su respuesta ideal.

### 5.2.2.2 Canales de Fase No Mínima

**Canal  $H_3$**  A continuación se presentan los resultados obtenidos al aplicar el algoritmo NLMS para igualar el canal  $H_3$ , el cual es un canal de fase máxima. Por lo tanto el igualador ideal es inestable. Por lo tanto cabría esperar que se necesitase de

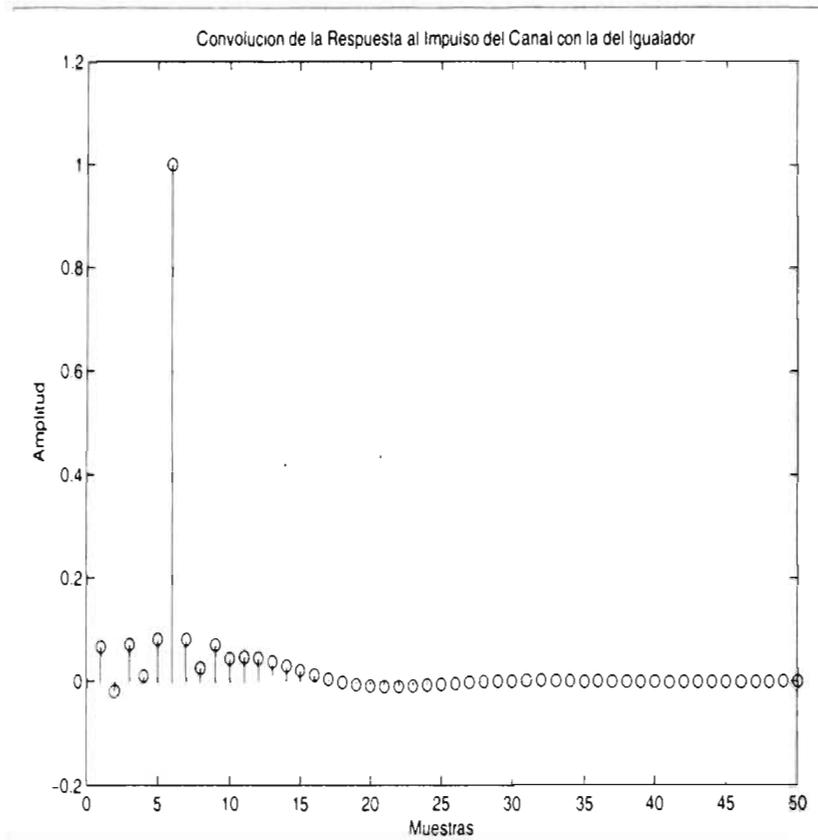


Figura 5.12: Convolución de la respuesta al impulso del canal con la del igualador. Algoritmo NLMS con  $\mu = 1.0$  y de 0.09. Canal  $H_3(z)$ . Línea continua (Canal).

un gran número de coeficientes para aproximar la respuesta al impulso del igualador ideal sin embargo a partir de seis coeficientes se logra una respuesta aceptable en el sentido de que las especificaciones del enlace de comunicaciones lo permitan. A

pesar de esto con un orden del filtro igual a diez se obtiene un resultado bastante aceptable pues la convolución de la respuesta al impulso de canal con la del igualador presenta una muestra unitaria y todas las demas son menores a 0.1. De aquí que tomando en cuenta las características del canal y dado que no se puede utilizar un filtro adaptable IIR, se puede decir que se obtiene un buen resultado ya que un filtro de orden diez es perfectamente realizable en la práctica,

Con el objeto de dar una idea de los resultados obtenidos en la figura 1.12 se muestra la convolución de la respuesta al impulso del canal con la respuesta al impulso del igualador. En este caso se emplearon 30 coeficientes para igualar el canal y un paso de adaptación de  $\mu = 1.0$  para las primeras 500 iteraciones y de 0.09 para el resto. De esta gráfica se puede apreciar que todavía existe una interferencia intersímbolos residual, la cual podría atenuarse aun más si se aumenta el orden del filtro. Sin embargo obsérvese de la tabla 5.4 que para un orden del filtro igual a 30 se tienen aproximadamente 2100 operaciones flotantes por muestra de salida, lo que ya es demasiado para una aplicación en tiempo real. Por lo tanto aumentar el orden del filtro no es una buena idea y más si se considera que el aumento en el número de operaciones con respecto al aumento en el número de coeficientes no es lineal por tanto sería mejor considerar un algoritmo rápido donde esta relación sí es lineal.

**Canal  $H_4(z)$**  A continuación se presentan los resultados obtenidos al igualar el canal  $H_4(z)$  con el algoritmo nlms. El resultado obtenido con un orden del filtro adaptivo igual a diez no es bueno, ya que la convolución de la respuesta al impulso del canal con la del igualador muestra un interferencia intersímbolos considerable. Sin embargo este resultado puede mejorarse conforme se incrementa el orden del filtro, lo que no es conveniente por las razones anteriormente citadas. Cuando se tiene un orden igual a 30 se logra una igualación más o menos buena, sin embargo siguen

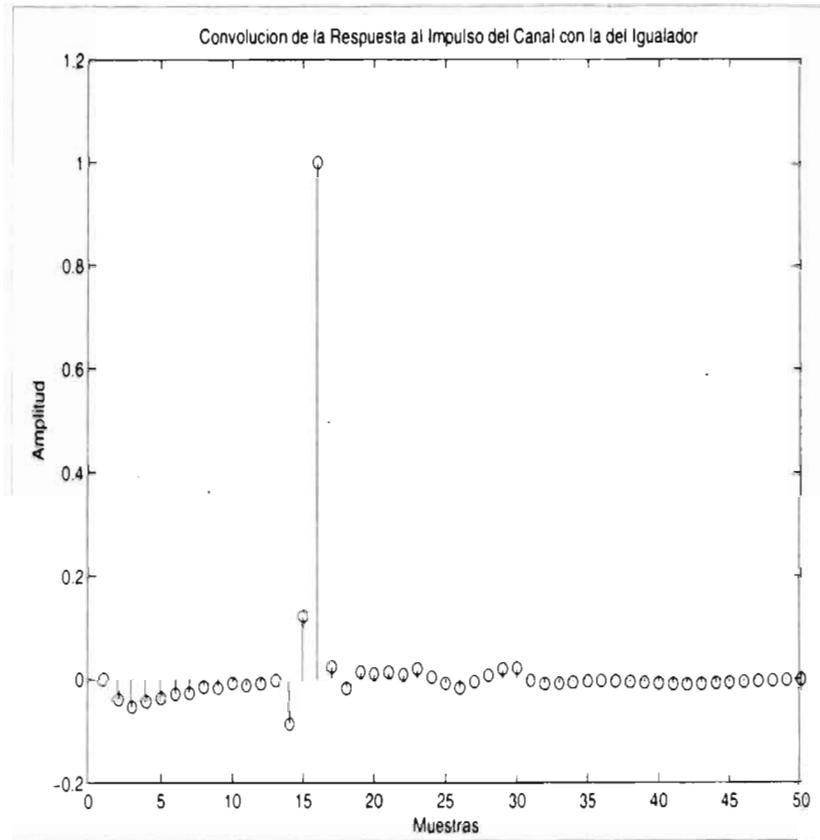


Figura 5.13: Convolución de las respuestas al impulso del canal y del igualador. Algoritmo NLMS con  $\mu = 1.0$  y de 0.09 a un tercio del número total de iteraciones. Canal  $H_4(z)$ .

existiendo algunos pulsos en la convolución de las respuestas al impulso aunque todos son muy pequeños en comparación con la unidad (vease la figura 5.13).

Al igual que en los casos anteriores la variación de la convergencia con respecto al orden del filtro no es muy grande y puede deberse a un simple error de apreciación: la convergencia promedio se produce alrededor de las 700 iteraciones, la máxima en 800 y la mínima en 500.

**Canal  $H_6$**  Los resultados obtenidos con un orden del filtro entre 10 y 15 no son buenos, ya que en la gráfica de la convolución de las respuestas al impulso del canal y del igualador se observo además de la muestra unitaria, muestras secundarias de una magnitud de hasta 0.35, lo que nos indica que todavía existe una interferencia intersímbolos de consideración. Sin embargo a apartir de un orden 20 los resultados son aceptables.

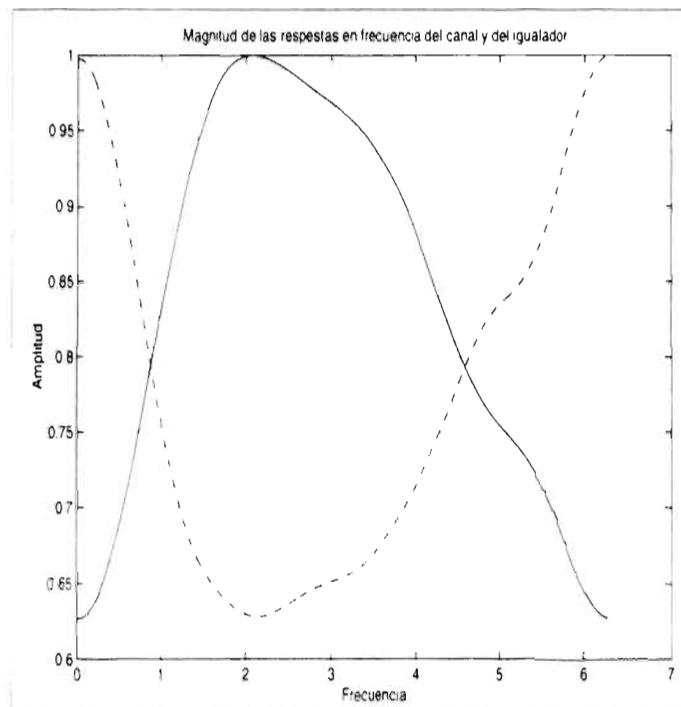


Figura 5.11: Respuesta en frecuencia del canal y del igualador. Algoritmo NLMS con  $\mu = 1.0$  y de 0.09 a las 500 iteraciones. Canal  $H_6(z)$ . Línea continua (Canal).

En [Pro95] se dice que  $H_6$  es representativo de un canal telefónico de buena calidad, lo que se corrobora con los resultados aquí presentados pues a pesar de ser un canal de fase no mínima, la igualación no presenta graves problemas. Para ilustrar

los resultados obtenidos, en la figura 5.15 se presenta la respuesta en frecuencia del canal y la del igualador para un filtro de orden 30.

**Canal  $H_7(z)$**  A continuación se comentarán los resultados obtenidos para  $H_7$ . Al igual que en los casos anteriores al tratarse de un canal de fase mínima se espera tener dificultades para la igualación del canal, además dado que el canal es FIR, el igualador ideal será IIR e inestable por lo que sólo se puede obtener una aproximación al mismo. Esta aproximación será mucho mejor mientras más coeficientes se consideren para igualar el canal sin embargo al realizar las evaluaciones se observó que con un orden igual a diez se obtienen resultados excelentes sin necesidad de incrementar el orden del filtro empleado para igualar el canal. Aparentemente un

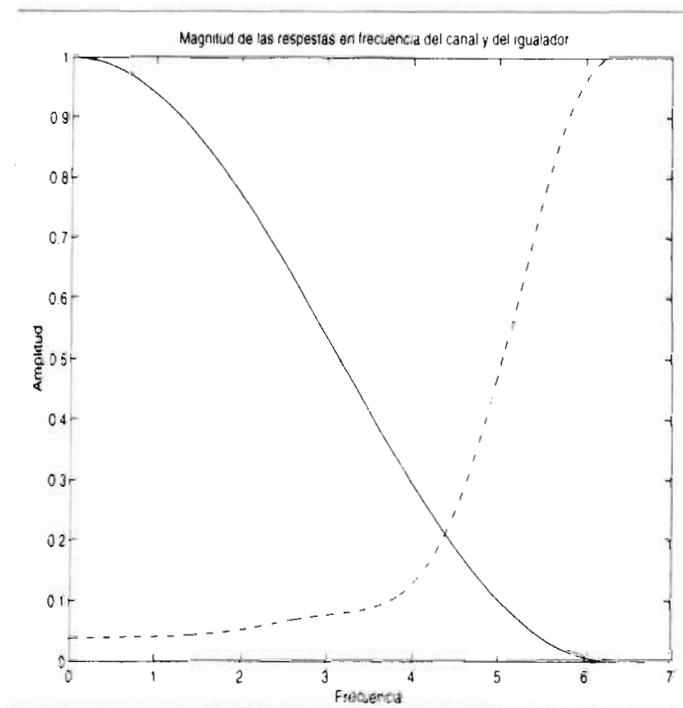


Figura 5.15: Respuesta en frecuencia del canal y del igualador. Algoritmo NLMS con  $\mu = 1.5$  y de 0.09 a las 1200 iteraciones. Canal  $H_7(z)$ . Línea continua (Canal).

filtro de orden diez no es un filtro de un orden exagerado para implementar en la práctica, sin embargo si tomamos en cuenta que la respuesta al impulso del canal sólo consta de tres coeficientes nos damos cuenta de que en realidad se necesita de

aproximadamente del triple de coeficientes para lograr una respuesta adecuada. De la gráfica de la convolución de las respuestas al impulso se observa además de la muestra unitaria, muestras de magnitud aproximada a 0.1 alternadas en signo, lo que indica que todavía existe una interferencia intersímbolo que no se ha podido cancelar. Sin embargo, la respuesta en frecuencia del igualador es ya una buena aproximación a la respuesta ideal. Para ilustrar este último punto vease la figura 5.16, en donde se presenta la gráfica de las respuestas en frecuencia tanto del canal como la del igualador.

**Canal  $H_8(z)$**  De acuerdo con [Pro95] este canal presenta las peores características espectrales con respecto a los canales  $H_6(z)$ ,  $H_7(z)$ , lo cual dificulta la tarea de la igualación de canal. De aquí que se requiera de un filtro de orden 30 para igualar el canal; pero a diferencia de  $H_6$  donde también se requirió de un filtro de orden 30, aquí la convergencia es más lenta requiriendo de 4 000 iteraciones para converger utilizando una  $\mu = 1.5$  para las primeras 2750 iteraciones y de 0.09 para el resto, en comparación con las dos mil iteraciones que se requirieron para lograr la convergencia para el canal  $H_6(z)$ .

De la gráfica de la convolución se aprecia una interferencia intersímbolos de cierta consideración ya que aun se observan muestras de -0.2 y de 0.1; las cuales como están alternadas en signo son representativas de un comportamiento oscilatorio. Sin embargo esta interferencia intersímbolos residual difícilmente se logrará cancelar al incrementar el orden del filtro adaptivo, pues se observa que al incrementar el orden del filtro las mejoras son mínimas por un lado, mientras que por el otro el aumento del número de las operaciones flotantes que esto acarrea hace irrealizable el filtro para un caso práctico. Por lo que para este caso se recomienda aceptar este desempeño a costa de obtener por otros medios la calidad deseada en la recepción.

La única mejora sensible al incrementar el orden del filtro radica en que la resonancia que presenta la respuesta en frecuencia del igualador compensa de una mejor manera el efecto de la antiresonancia del canal. De aquí podemos concluir, que los dos canales más difíciles de igualar son los que presentan antiresonancias en su respuesta en frecuencia tal y como se refiere en [Pro95].

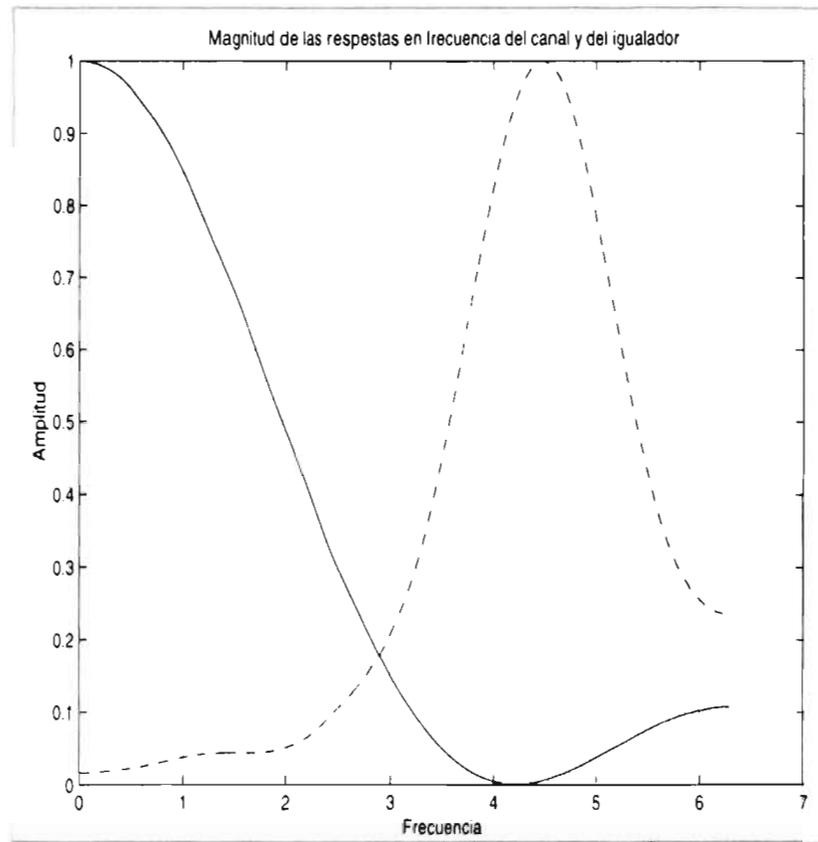


Figura 5.16: Respuesta en frecuencia del canal y del igualador. Algoritmo NLMS con  $\mu = 1.5$  y de 0.09 a las 2750 iteraciones. Canal  $H_8(z)$ . Línea continua (Canal).

### 5.2.3 Evaluaciones con el Algoritmo "Fast Transversal Filters FTF"

En esta sección se presentarán los resultados que se obtuvieron de aplicar el algoritmo FTF para igualar los canales tratados en este trabajo. En general los resultados obtenidos para el algoritmo FTF no difieren de los obtenidos para el caso del algoritmo NLMS por lo que solo se hará referencia a las diferencias entre ambos métodos.

Orden del Filtro	Número de Operaciones Flotantes por Muestra de Salida
6	173.90
8	223.95
10	272.96
12	322.43
14	371.86
16	421.25
18	470.60
20	519.91
22	569.18
24	618.41
26	667.60
28	716.75
30	765.86
40	1010.81
50	1254.76
70	1739.66
100	2459.51

Tabla 5.5: Número de operaciones flotantes por muestra de salida para el algoritmo FTF

Como lo indica el mismo nombre del algoritmo FTF, éste es un método rápido lo cual nos indica que el número de operaciones flotantes se incrementa de una forma lineal con respecto al incremento del número de coeficientes a ajustar del filtro adaptivo [Hay91]. Este hecho es de gran repercusión para los canales aquí tratados ya que como se menciona en la sección anterior los canales de fase no mínima en general requieren de un gran número de coeficientes para lograr una igualación aceptable; sin embargo no es muy recomendable aplicar el algoritmo nlms en estos casos debido

a que el aumento en el número de operaciones flotantes con respecto al número de coeficientes a ajustar no es lineal, lo cuál finalmente afecta nuestra habilidad para establecer un enlace de comunicaciones en tiempo real a altas tasas de transmisión. Sin embargo, para el caso del algoritmo FTF es posible incrementar un poco más el orden del filtro, lo cuál nos permite una mejor igualación sin pagar el costo del exagerado aumento en el número de operaciones flotantes que se tiene que pagar en el caso del algoritmo nlms (compare las tablas 5.4 y 5.5).

Otra de las ventajas del algoritmo FTF es que converge más rápido a la solución óptima que los algoritmos adaptivos basados en el gradiente estocástico, como lo es en este caso el algoritmo nlms. Por lo cuál desde el punto de vista de la rapidez de convergencia y del número de operaciones es más recomendable utilizar el algoritmo FTF para los canales tratados en este trabajo. Sin embargo, desde el punto de vista de una implementación en un procesador digital de señales de punto fijo es necesario analizar la estabilidad numérica de los algoritmos, lo cuál esta fuera del objetivo de este trabajo.

### 5.2.3.1 Canales de Fase Mínima

En esta sección se analiza el desempeño del algoritmo FTF aplicado en la igualación de canales de fase mínima. En general los resultados son semejantes a los obtenidos con el algoritmo nlms.

**Canal  $H_1(z)$**  El problema fundamental que presenta el algoritmo FTF para la igualación de este canal radica a igual que para el algoritmo nlms, en que no puede compensar debidamente la antiresonancia que presenta el canal en baja frecuencia. Se puede pensar que al incrementar el orden del canal, éste nulo tendrá que ser compensado a partir de cierto valor para el orden del filtro. Sin embargo los beneficios que se obtienen al elevar el orden del canal son muy graduales, de esta manera se alcanza un orden de 100 y no se logra compensar este nulo. Por lo que se sugiere el uso del algoritmo SARF, el cual no tuvo mayores problemas para igualar el canal aquí tratado. Para tener una idea de los resultados obtenidos vease la figura 5.7, la cuál se obtuvo utilizando el método nlms, sin embargo como el resultado es muy similar al obtenido con el método FTF se prefirió no ser repetitivo.

Además como en el caso anterior la convolución de las respuestas al impulso del igualador con la del canal muestra una interferencia intersímbolos residual que no ha podido ser compensada mediante la acción del igualador. Esta interfeerencia se debe a que la convolución no es sólo una muestra unitaria sino que presenta muestras secundarias de una magnitud menor a 0.1 para un orden del filtro adaptivo igual a 30. Sin embargo el algoritmo FTF para este caso converge alrededor de las 1000 iteraciones mientras que el algoritmo nlms converge a aproximadamente a las 1100 iteraciones pero haciendo uso del recurso de cambiar el paso de adaptación despues de las 1050 iteraciones; sin emargo recuerdese que en esta estrategia el número de iteraciones en que converge el algoritmo depende del instante de tiempo en el que

se cambie el valor del paso de adaptación. Por otro lado el número de operaciones flotantes por muestra de salida para este último caso es de 765.86 para el algoritmo FTF y de 2 103.62 para el caso del algoritmo nlms.

**Canal  $H_2$**  Los resultados obtenidos para este canal son bastante similares a los obtenidos con el algoritmo nlms. En este caso debido a que la antiresonancia del canal es menos pronunciada que en el caso anterior, los algoritmos NLMS y FTF logran una mejor igualación.

El orden del filtro adaptivo se vario desde 8 hasta 30, los mejores resultados se obtuvieron con un orden igual a 30, sin embargo a partir de un orden igual a 18 se obtienen buenos resultados. En todos estos casos la resonancia del igualador no coincide exactamente con la antiresonancia del canal sino que presenta un corrimiento hacia baja frecuencia, el cual va disminuyendo conforme se aumenta el orden del filtro adaptivo.

Dado que los resultados son muy semejantes a los obtenidos con el algoritmo nlms se prefirió no incluir más gráficas para no ser repetitivos. Sin embargo es conveniente comparar los resultados obtenidos utilizando ambos algoritmos para un orden del filtro igual a 30: el algoritmo FTF para este caso converge alrededor de las mil iteraciones, mientras que el algoritmo NLMS converge a las cinco mil iteraciones a pesar de usar una  $\mu$  de 1.5 para las primeras 1050 iteraciones y una  $\mu$  de 0.09 para las restantes. En cuanto al número de operaciones flotantes por muestra de salida el algoritmo ftf claramente es superior, ya que este índice es tres veces menor que para el caso del algoritmo NLMS.

**Canal  $H_5(z)$**  Este canal resulto difícil de igualar cuando se le aplico el algoritmo SARF debido a la posición relativa de sus polos y ceros. Dada la proximidad que existe entre el par de polos conjugado dado por  $0.9000e^{\pm j135^\circ}$  y el par de ceros conjugado dado por  $0.8252e^{\pm j134.46^\circ}$ , por las razones anteriormente citadas se puede producir una cancelación indeseable de polos y ceros dando por resultado una igualación no muy buena. Sin embargo para los métodos FTF y NLMS, la igualación que se logra es muy buena cuando se utiliza un filtro de orden entre 25 y 30. Sin embargo los resultados se pueden considerar buenos a partir de un orden 10. Los resultados son mejores en estos dos últimos casos debido a que se esta utilizando un filtro FIR de un gran número de coeficientes para aproximar la respuesta impulsional infinita del canal.

Aquí como en los casos anteriores se refiere al lector a la figura 5.11 para obtener una idea de los resultados obtenidos al aplicar el algoritmo FTF para igualar el canal aquí tratado, ya que los resultados obtenidos no varían en gran medida con respecto a los resultados obtenidos para este mismo canal pero con el algoritmo NLMS.

### 5.2.3.2 Canales de Fase no Mínima

**Canal  $H_3$**  En este caso a pesar de que  $H_3$  es un canal de fase no mínima a partir de un orden seis para el filtro adaptivo la igualación es buena en el sentido de que la convolución de las respuestas al impulso del canal y del igualador presenta además de la muestra unitaria, muestras secundarias de valor menor a 0.2. Sin embargo, esto se puede mejorar si se incrementa el orden del filtro a 10, de esta manera las muestras secundarias que aparecen en la convolución son menores a 0.1.

En cuanto a la convergencia de este algoritmo para el canal aquí tratado, de la gráfica de la magnitud del vector de coeficientes del filtro adaptivo contra el número de iteraciones se puede apreciar que el algoritmo converge aproximadamente a las 4000 iteraciones dado que ésta curva tiene una pendiente sumamente pequeña. Por otro lado el algoritmo NLMS converge a las 25 mil iteraciones cuando se usa una  $\mu$  de uno. Dado que los resultados obtenidos con el algoritmo FTF son semejantes a los obtenidos con el algoritmo NLMS, se refiere a lector a la figura 1.12, para obtener una idea de los resultados obtenidos.

**Canal  $H_4$**  En este caso se logra una muy buena igualación a partir de un orden 30 para el filtro adaptivo, en el sentido de que la convolución de las respuestas al impulso presenta muestras secundarias menores a 0.05 en magnitud. En este caso la convergencia se logra alrededor de las 100 iteraciones, mientras que el algoritmo NLMS converge a las 500 iteraciones utilizando una  $\mu$  de 1.0 y a la tercera parte del número de iteraciones totales se cambia a 0.09.

El orden del filtro se vario desde 10, donde la convolución de las respuestas al impulso presenta muestras secundarias menores 0.2: hasta llegar a un orden 30 donde la convolución es casi una sola muestra unitaria salvo una o dos muestras de valor menor a 0.1 y el resto de un valor prácticamente igual a cero. En cuanto a la respuesta en frecuencia del igualador, ésta es muy semejante a la obtenida con el algoritmo NLMS. Sin embargo, la diferencia más notable entre ambos algoritmos radica al igual que en los casos anteriores en la convergencia y en el número de operaciones flotantes. Mientras que el algoritmo FTF converge a las 100 iteraciones aproximadamente, el algoritmo NLMS converge aproximadamente a las 700 iteraciones utilizando una  $\mu$  de 1.0 para el primer tercio del número total de iteraciones y de 0.09 para el resto. Para obtener una idea de los resultados obtenidos vease la figura 5.13.

**Canal  $H_6$**  Este canal como se meciono anteriormente es el modelo de un cable telefónico de buena calidad [Pro95] por lo tanto no presenta nulos ó antiresonancias, lo cual facilita la tarea del igualador [Pro95].

Al variar el orden del filtro adaptivo se observo que a partir de un orden igual a 15 se obtiene una convolución cuyas muestras secundarias no exceden de un décimo de la muestra unitaria por lo que se puede considerar un resultado bueno si las especificaciones del enlace de comunicaciones nos lo permiten. Como es natural se obtienen mejores resultados al ir incrementando el orden del filtro hasta llegar a

un orden entre 25 y 30 donde los resultados son excelentes. Recuerdese que para el algoritmo FTF el incremento en el número de operaciones flotantes por muestra de salida con respecto al incremento en el número de coeficientes a ajustar es lineal, por lo que no es tan drástico aumentar el orden del filtro para obtener mejores resultados en la igualación de canal como en el caso del algoritmo NLMS.

El algoritmo FTF para este caso converge alrededor de las 50 iteraciones mientras que el algoritmo NLMS converge a las 1800 iteraciones con una  $\mu$  de 0.09. Finalmente dado que los resultados obtenidos para igualar el canal hasta aquí tratado, utilizando el algoritmo FTF son muy similares a los obtenidos con el algoritmo NLMS se refiere al lector a la figura 5.14.

**Canal  $H_7$ :** Este canal presenta peores características espectrales que el anterior según [Pro95]. De aquí que si se utiliza un filtro de orden 30 para igualar el canal se obtengan resultados muy buenos, sin embargo la convolución de las respuestas al impulso del canal y del igualador sigue presentando muestras secundarias menor a 0.1. Esto quiere decir que existe cierta interferencia intersímbolos que no ha podido ser compensada por el igualador. Sin embargo si se utiliza un filtro de orden igual a 70 se obtiene una convolución la cuál prácticamente es igual a una sola muestra, sin embargo el algoritmo FTF para este orden converge alrededor de las 3500 iteraciones. Nótese que el número de operaciones flotantes por muestra de salida para este último caso es de 1740. Este mismo número de operaciones corresponde a un filtro de aproximadamente orden 28 implementado con el algoritmo NLMS. Por lo tanto, tanto desde el punto de vista de la convergencia como del número de operaciones flotantes conviene utilizar el algoritmo FTF. Aquí cabe señalar que esta última recomendación es tomando en cuenta que los algoritmos se implementarán en punto flotante ó que la longitud de palabra es infinita; ya que para implementaciones en punto fijo con una longitud de palabra finita será necesario hacer un análisis de la estabilidad numérica de ambos algoritmos frente a errores de cuantización. Finalmente, ya que los resultados son bastante similares con respecto a los obtenidos con el algoritmo nlms, se refiere al lector a la fig. 5.15 para obtener una idea de los resultados obtenidos con el algoritmo aquí tratado.

**Canal  $H_8(z)$**  Este canal muestra características espectrales más pobres con respecto a los dos casos anteriores; por lo tanto cabe esperar que los resultados de la igualación no sean tan buenos como los que se obtuvieron para  $H_6$  y  $H_7$ .

Nuevamente los resultados no difieren demasiado de los obtenidos utilizando el algoritmo NLMS. Un filtro de orden igual a 30, produce buenos resultados en el sentido de que la convolución de las respuestas al impulso del canal y del igualador presenta muestras secundarias menores a 0.1; lo cuál es indicativo de una interferencia intersímbolos residual que no ha podido ser eliminada por la acción del igualador. Por otro lado, se observa que el incremento en el orden del filtro adaptivo produce una mejoría en los resultados obtenidos muy gradual, de tal manera que los resultados no compensan el incremento en el número de operaciones flotantes por

muestra de salida.

El algoritmo FTF converge a las dos mil iteraciones aproximadamente, mientras que el algoritmo NLMS converge aproximadamente entre las 4000 y 4500 iteraciones cuando se usa una  $\mu$  de 1.5 para las primeras 2750 y una  $\mu$  de 0.09 para el resto de las iteraciones. Finalmente para obtener una idea de los resultados obtenidos y dado que éstos son muy similares a los obtenidos al aplicar el algoritmo NLMS, se refiere al lector a la fig. 5.16.

#### 5.2.4 Evaluaciones con el Algoritmo Least "Lattice" Squares (LSL) con Errores A Posteriori de Estimación

En esta sección se comentarán los resultados de aplicar el algoritmo LSL a la igualación de los canales tratados en este capítulo. En general los resultados obtenidos son iguales a los que se obtuvieron en las evaluaciones que se realizaron con el algoritmo FTF. La diferencia más notable con respecto al algoritmo FTF radica en el número de operaciones flotantes que realiza el algoritmo LSL por muestra de salida. Si se compara la tabla 5.6 con las tablas 5.5 y 5.4 se observa que el algoritmo LSL realiza más operaciones por muestra de salida, lo cual no es muy conveniente para los algoritmos tratados en este capítulo ya que como se señaló en su oportunidad los igualadores ideales para los canales aquí presentados en general tienen una respuesta impulsional infinita y, si el canal es de fase no mínima entonces el igualador ideal será inestable. Esto nos conduce de una manera inevitable a aproximar una respuesta impulsional infinita con un número finito de coeficientes ó taps del igualador. Por lo tanto resulta lógico que se requiera de un orden elevado del filtro adaptable para lograr una igualación adecuada. Dado que el algoritmo LSL realiza una mayor cantidad de operaciones flotantes por muestra de salida que el algoritmo FTF, también será el que más tiempo de procesamiento requiera. Sin embargo, su convergencia es igual a la del algoritmo FTF, por lo cual converge más rápido que el algoritmo NLMS.

Por otro lado la gran ventaja de una estructura "lattice" es su alto grado de modularidad, para visualizar esto de una mejor forma considere el caso en el que la función de transferencia del canal cambie de orden frente al tiempo ó bien considere el caso en el cual no se conoce con exactitud el orden de la función de transferencia del canal que se desea igualar. En estos casos si se desea aumentar el orden o disminuir el orden del filtro adaptivo de una forma dinámica basta con sólo agregar o eliminar respectivamente las etapas necesarias del filtro para lograr el orden que se desee. Mientras que en el caso de un filtro de estructura transversal esto último no es posible, por lo que se tiene que volver a realizar la estimación con el orden del filtro deseado o bien se tiene que estar haciendo continuamente la conversión entre los coeficientes del filtro que el algoritmo LSL realiza más operaciones flotantes por muestra de salida que el algoritmo FTF. El filtro de una sola realización en el tiempo sino que también realiza recursiones en el orden, las cuales incrementan el número de operaciones flotantes que intervienen en la cuenta que realiza el comando flops

Orden del Filtro	Número de Operaciones Flotantes por Muestra de Salida
6	279.02
8	377.05
10	475.08
12	573.11
14	671.14
16	769.17
18	867.20
20	965.23
22	1063.26
24	1161.29
26	1259.32
28	1357.35
30	1455.38
40	1945.53
50	2435.68
70	3415.98
100	4886.43

Tabla 5.6: Número de operaciones flotantes por muestra de salida para el algoritmo LSI.

de MATLAB.

### 5.3 Conclusiones

En este capítulo se analizó el desempeño por separado de cada uno de los algoritmos aquí tratados sin dejar de mencionar las diferencias más notables donde se considero oportuno. Dado que el objetivo de este trabajo consiste en comparar además de analizar el desempeño de los distintos algoritmos aquí tratados, en el siguiente capítulo se compararan los resultados obtenidos para los distintos algoritmos y, se indicará cual de todos los algoritmos es el idoneo para igualar el canal según las especificaciones previamente establecidas para el enlace de comunicaciones.

En lo que respecta a los distintos tipos de canal se observa que los algoritmos que poseen antiresonancias o nulos espectrales son los más difíciles de igualar en el sentido de que se requiere de un filtro de mayor orden para lograr resultados

satisfactorios. En este último caso si el canal es de fase mínima se puede aplicar un algoritmo adaptable IIR, el cual produce los resultados adecuados. Sin embargo, cabe **hacer** notar que estos algoritmos ven limitada su utilidad seriamente ya que sólo se **aplican** a canales de fase mínima pero en la práctica la gran mayoría de los canales **son** de fase no mínima [bSH94]. Por tanto la investigación continua abierta y más aun si tomamos en cuenta la creciente necesidad de las sociedades modernas de transmitir grandes cantidades de información en tiempo real.

## Capítulo 6

# Comparación de los Algoritmos

En este capítulo se compara el desempeño de los algoritmos tratados individualmente en el capítulo cinco, aunque ya se hayan mencionado algunas de las diferencias entre cada uno de los métodos tratados en este trabajo. Se ha preferido separar las evaluaciones individuales de cada algoritmo de las comparaciones entre estos con el objeto de hacer un énfasis mayor en las diferencias resultantes de aplicar cada algoritmo tratado a cada uno de los canales aquí tratados, dado que el objetivo final de este trabajo es dar una serie de recomendaciones referentes a la aplicación de los distintos algoritmos de igualación de canal.

En este capítulo se incluirán más gráficas donde sea necesario con el objeto de efectuar las comparaciones entre los distintos algoritmos aquí tratados.

### 6.1 Comparación de Resultados

Como primer punto para efectuar esta comparación de resultados es interesante comparar el número de operaciones flotantes por muestra de salida que realiza cada algoritmo, dado que este índice de desempeño nos dará una idea de cual es el algoritmo más rápido bajo el tipo de arquitectura empleada.

Como es de esperarse los algoritmos rápidos, en este caso el algoritmo FTF y el algoritmo LSL, guardan una relación lineal entre el número de coeficientes a ajustar y el número de operaciones flotantes por muestra de salida; por otro lado en el algoritmo NLMS el número de operaciones flotantes por muestra de salida crece a una tasa mayor que el número de coeficientes a ajustar. De aquí que si se requiere de un gran número de coeficientes para igualar un determinado canal, sea más conveniente utilizar un algoritmo rápido ya sea el FTF o el LSL.

Otra característica importante de cualquier algoritmo de filtrado adaptable es el número de iteraciones que realiza para converger a la solución óptima. Como era de esperarse los algoritmos rápidos convergen de una manera más rápida a la solución óptima que los algoritmos de gradiente; por lo cual se recomienda la utilización de algoritmos rápidos como el FTF y el LSL en caso que se requiera un enlace de comunicaciones en tiempo real a altas tasas de transmisión.

Orden del Filtro	NLMS	FTF	LSL
6	136.22	173.90	279.02
8	212.17	223.45	377.05
10	304.12	272.96	475.08
12	412.07	322.43	573.11
14	536.02	371.86	671.14
16	675.97	421.25	769.17
18	831.42	470.60	867.20
20	1003.87	519.91	965.23
22	1191.82	569.18	1063.26
24	1395.77	618.41	1161.29
26	1615.72	667.60	1259.32
28	1851.67	716.75	1357.35
30	2103.62	765.86	1455.38
40	3603.37	1010.81	1945.53
50	5503.12	1254.76	2435.68
70	10502.62	1739.66	3415.98
100	21001.87	2459.51	4886.43

Tabla 6.1: Número de operaciones flotantes por muestra de salida para los algoritmos NLMS, FTF y LSL.

Otro punto importante a tomar en cuenta en la implantación de un algoritmo de filtrado adaptable en un procesador de punto fijo lo constituye su robustez numérica. Dado que la estructura "lattice" presenta mejores propiedades numéricas que la estructura transversal, se prefiere pagar el aumento en el número de operaciones flotantes por muestra de salida que representa el algoritmo LSL en comparación del algoritmo FTF a cambio de una mejor robustez numérica. Sólo se preferirá el algoritmo FTF en caso en los que el número de operaciones flotantes por muestra de salida sea una consideración crítica para el enlace de comunicaciones. En tal caso se deberá compensar la robustez numérica más pobre del algoritmo FTF utilizando un procesador de mayor longitud de palabra. Por otro lado el algoritmo NLMS presenta buenas propiedades numéricas por lo que si las especificaciones del enlace en cuanto a velocidad de transmisión lo permite se implantará este algoritmo.

## 6.1.1 Canales de Fase Mínima

### 6.1.1.1 Canal $H_1(z)$

En este caso el único algoritmo que puede compensar perfectamente la antiresonancia que presenta la respuesta en frecuencia del canal es el algoritmo SARF; sin embargo se debe pagar cierto precio por este alto desempeño. En primer lugar fue necesario elevar el orden del filtro ya que de lo contrario el algoritmo converge a un mínimo local, lo que no es muy conveniente desde el punto de vista computacional ya que de no existir un mínimo local en este caso, el algoritmo realizaría del orden de 74 operaciones flotantes por muestra de salida sin embargo al elevarse el orden a cuatro ceros y seis polos el algoritmo realiza cerca de 7100 iteraciones por muestra de salida, lo que si se compara con los algoritmos de estructura FIR correspondería a un filtro de un orden entre 50 y 70 para el algoritmo NLMS, a un filtro de orden mayor que 100 para los algoritmos FTF y LSL. De aquí que no sea tan grande el ahorro que propone el algoritmo SARF, sin embargo si lo que se desea es calidad necesitaremos recurrir a él. Para obtener una idea de los resultados obtenidos se presenta en la figura 6.1 la gráfica de la respuesta en frecuencia del canal y del igualador para la igualación del canal  $H_1(z)$  utilizando el algoritmo SARF; la cual se debe comparar con la figura 5.7 del capítulo anterior que representa el mismo resultado pero usando el algoritmo NLMS.

Antes de abandonar este tema cabe señalar, como se mencionó a su debido tiempo, que los algoritmos que utilizan polos y ceros sufren de problemas de inestabilidad. Por lo tanto en muchos casos se hace necesario un monitoreo de la estabilidad lo que provoca el gran aumento en el número de operaciones flotantes que realiza un algoritmo de este tipo por muestra de salida. Se han propuesto varios métodos para monitorear la estabilidad de esta clase de filtros pero en general ninguno ha resuelto el problema de una forma satisfactoria [CRJ84].

Por otro lado si no se requiere de una igualación de alta calidad se puede optar por el algoritmo FTF para igualar este canal. Se puede usar un filtro de orden 30 para el cual el algoritmo FTF realiza aproximadamente 766 operaciones flotantes por muestra de salida y; así aunque la igualación no es tan buena se alivia la carga computacional del receptor en cuanto al proceso de igualación se refiere y; lo más importante se puede cumplir con las especificaciones del enlace por otros medios como sería el utilizar un código para detectar y corregir errores más eficiente, elevar la potencia de transmisión, etc.

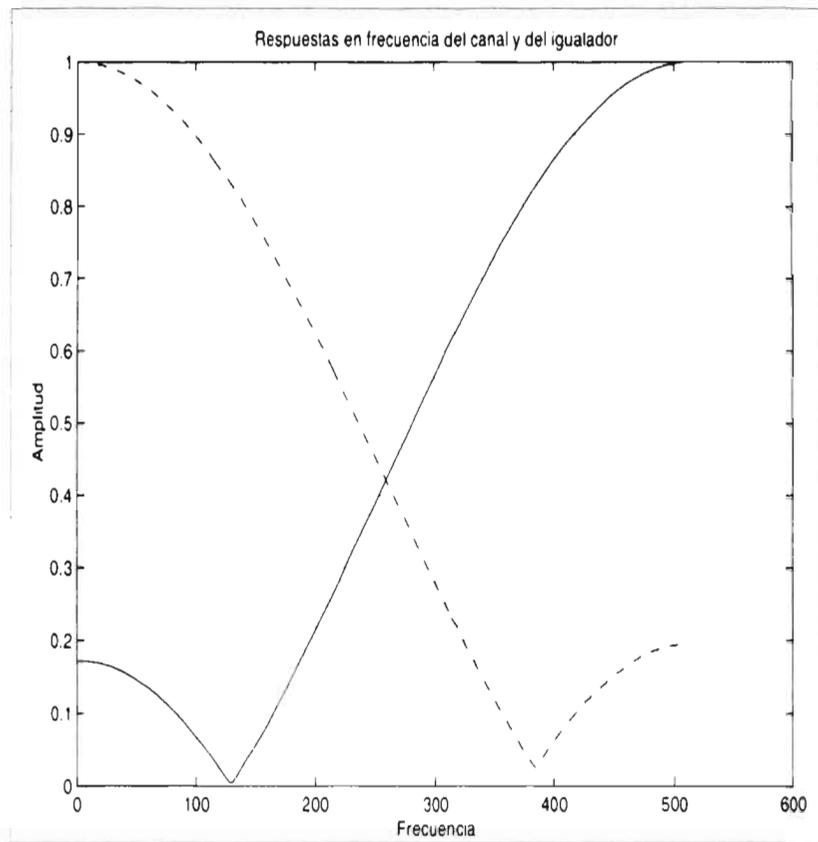


Figura 6.1: Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo SARF para igualar el canal  $H_1(z)$ . Línea punteada (igualador).

### 6.1.1.2 Canal $H_2$

Aquí nuevamente como en el caso anterior si lo que se desea es una alta calidad en la igualación se debe utilizar el algoritmo SARF. El precio a pagar es como en el caso anterior una elevada carga computacional debida a los dos siguientes factores: 1) la necesidad de elevar el orden del filtro por que de lo contrario el algoritmo converge a un mínimo local y, 2) principalmente a la necesidad del monitoreo de la estabilidad del filtro. Dado que el orden del filtro se elevo de un modelo de dos polos a un modelo de seis polos y cuatro ceros, el algoritmo realiza aproximadamente 7 100 operaciones flotantas por muestra de salida, lo cual ya es de consideración. De esta manera si no

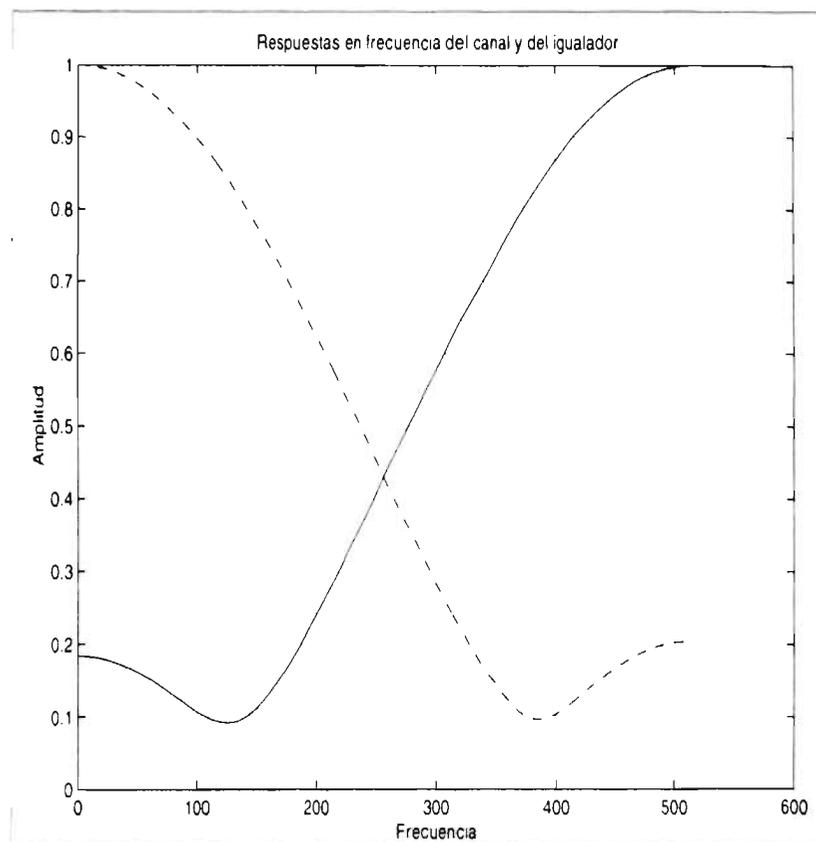


Figura 6.2: Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo SARF para igualar el canal  $H_2(z)$ . Línea punteada (igualador).

es necesaria una igualación de tan alta calidad y aprovechando el hecho de que la

antiresonancia espectral que presenta este canal es más sencilla de igualar se puede usar un algoritmo FTF con un orden de 30, el cuál sólo realizará 766 operaciones flotantes por muestra de salida, lo que se traducirá en una reducción muy notoria de la carga computacional en el receptor. Al mismo tiempo que se realiza este cambio se puede buscar por otros medios el cumplimiento de las especificaciones del enlace de comunicaciones.

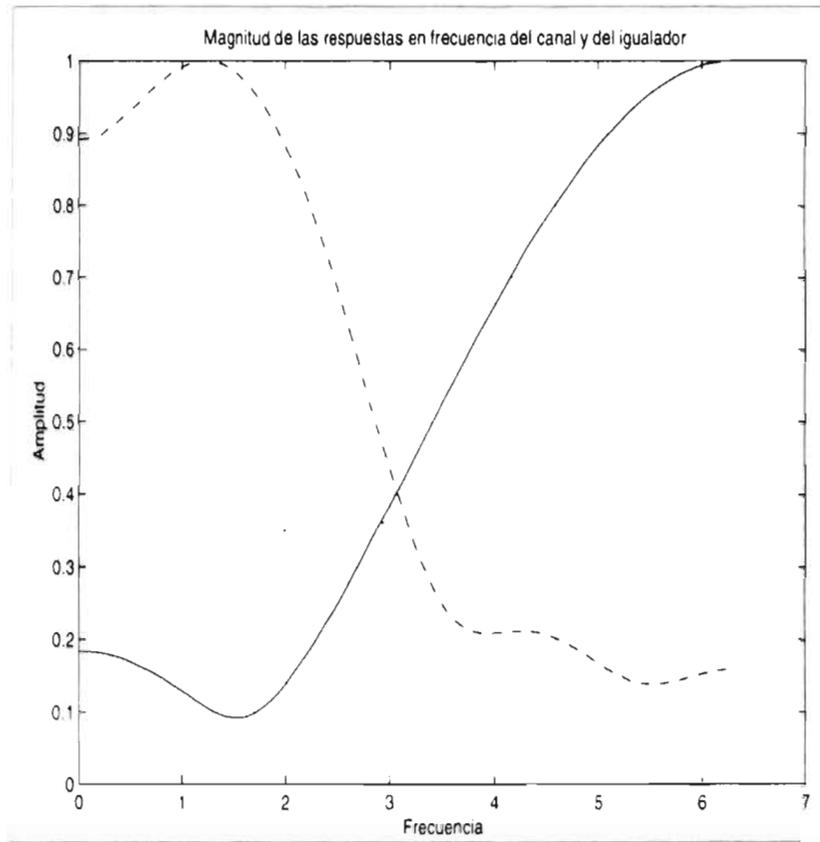


Figura 6.3: Respuesta en frecuencia del canal y del igualador cuando se utiliza el algoritmo FTF para igualar el canal  $H_2(z)$ . Línea punteada (igualador).

Con el objeto de dar una idea de los resultados obtenidos al igualar este canal con el algoritmo SARF y el algoritmo FTF se presentan en las figuras 6.2 y 6.3 las respuestas en frecuencia de los igualadores obtenidos al aplicar aplicar estos algoritmos para igualar el canal aquí tratado. En la figura 6.2, se tiene los resultados de aplicar el algoritmo SARF para igualar el canal aquí tratado con un filtro de

seis polos y cuatro ceros dado que para un filtro de orden inferior el algoritmo converge a un mínimo local. Mientras que por otro lado en la figura 6.3 se muestra la misma gráfica para el caso del algoritmo FTF utilizando un filtro de orden 12. Para este último caso el algoritmo convergió aproximadamente poco después de las 500 iteraciones. Como se puede apreciar de ambas gráficas el resultado obtenido con el algoritmo SARF es el idóneo, mientras que el resultado obtenido con el algoritmo FTF no es tan bueno pero como se explicó anteriormente realiza un menor número de operaciones flotantes por muestra de salida. Por lo tanto, para los algoritmos adaptables IIR es vital el desarrollo de un mecanismo computacionalmente eficiente para estabilizar el igualador.

### 6.1.1.3 Canal $H(z)$

En este caso el algoritmo SARF no es capaz de igualar el canal correctamente, debido a la cercanía de los polos y ceros de este sistema, lo cual implica que debido a errores numéricos puede ocurrir una cancelación indeseable de polos y ceros dando lugar a una mala igualación de este canal. Es por ello que los algoritmos de estructura FIR funcionan mejor, ya que ellos sólo necesitarán de un gran número de coeficientes para aproximar la respuesta impulsional infinita del igualador ideal, de esta manera lograrán una igualación muy buena pero al costo de utilizar un filtro de un orden relativamente grande. Este costo como se hizo mencio para los dos casos anteriores esta en entre dicho, ya que la necesidad de monitorear la estabilidad del filtro incrementa demasiado el costo computacional de aplicar un algoritmo de este tipo.

Por otra parte, si la prioridad es igualar el canal en tiempo real, el algoritmo que se debe seleccionar sin duda alguna es el FTF. Sin embargo si se requiere implementar el algoritmo en punto fijo será necesario un análisis de la estabilidad numérica de los algoritmos aquí tratados. Por lo que esta recomendación tiene lugar si la implementación se lleva a cabo en punto flotante.

Finalmente no se debe menospreciar la flexibilidad de la estructura "lattice", la cual puede resultar de extrema utilidad ya que para aumentar ó disminuir el orden del filtro adaptivo solo se requiere de aumentar o disminuir el número adecuado de etapas para obtener un filtro del orden deseado, mientras que para el caso de una estructura transversal esto no es posible. Por lo tanto, la estructura "lattice" resulta de utilidad si se requiere de variar el orden del filtro de una manera adaptiva.

Con el objeto de obtener una idea de los resultados obtenidos al igualar este canal observese las gráficas de las figuras 5.3 y la figura 5.11 donde se comparan las respuestas en frecuencia del igualador con la del canal obtenidas con el algoritmo SARF y el algoritmo NLMS. Aquí se prefirió no incluir más figuras para no ser repetitivos, ya que los resultados obtenidos con el algoritmo FTF y LSL son similares a los obtenidos con el algoritmo NLMS, la única diferencia es que el algoritmo NLMS converge más despacio a la solución óptima.

### 6.1.2 Canales de Fase no Mínima

Como se señaló anteriormente esta clase de canales no pueden ser igualados utilizando un filtro adaptivo IIR [CG85], por lo que es necesario recurrir a un filtro de estructura FIR. Sin embargo, estos esquemas de igualación presentaran el inconveniente de requerir un gran número de coeficientes para poder realizar la igualación de una manera satisfactoria. Esto debido a que el igualador ideal para un canal de fase no mínima será inestable y de respuesta impulsional infinita.

Como se demostro anteriormente, en general para poder igualar los canales de fase no mínima aquí tratados se requirió de un gran número de coeficientes para lograr una igualación satisfactoria. Sin embargo el hecho de elevar el número de coeficientes de esta manera nos conduce inevitablemente a la selección de un algoritmo rápido para poder igualar el canal con el mínimo número de operaciones flotantes por muestra de salida. Sin embargo, si se quiere efectuar una implantación de estos algoritmos en punto fijo será necesario hacer un análisis de la estabilidad numérica de los mismos por lo que nuestra selección final del algoritmo también se verá influida por el tipo de arquitectura empleada así como la precisión numérica disponible para dicha arquitectura.

Aunque analizar las características de robustez y estabilidad numérica no es el propósito del presente trabajo cabe señalar que el algoritmo LSL es de gran interés en el caso de llevar a cabo una implementación del mismo en punto fijo, ya que la estructura "lattice" del filtro igualador presenta mejores propiedades de robustez y estabilidad numérica que su contraparte transversal [PM96]. Además cabe señalar que también existen algoritmos adaptables IIR de estructura "lattice", los cuales tienen el gran atractivo de que no es necesario obtener los polos del filtro IIR para verificar su estabilidad sino basta con que todos los coeficientes de reflexión sean menores a la unidad. Así de esta manera no se requiere de calcular los polos del filtro adaptable para cada iteración del algoritmo sino solo para aquellas iteraciones en que el algoritmo sea inestable.

#### 6.1.2.1 Presentación de Gráficas

Como se menciono anteriormente, los algoritmos NLMS, FTF y LSL no difieren en sus resultados para los canales tratados en este trabajo pues la solución a la que convergen es prácticamente la misma (véase el capítulo cinco), la diferencia fundamental es la rapidez de convergencia; mientras los algoritmos FTF y LSL convergen prácticamente en el mismo número de iteraciones para un orden del filtro dado, el algoritmo NLMS necesita de un número mucho mayor de iteraciones para converger a la misma solución para el mismo canal y el mismo orden del filtro empleado, a pesar de en el algoritmo NLMS se utilizó una  $\mu$  relativamente grande para las primeras iteraciones y luego se cambió a una  $\mu$  pequeña para el resto de las iteraciones. Se hubiera podido presentar un cuadro sinóptico donde se compararán el número de iteraciones a las cuales converge cada algoritmo sin embargo dado que la determinación del número exacto de iteraciones a las cuales converge el algoritmo

es una medida un tanto subjetiva, se dejará juzgar al lector este índice de desempeño a partir de las correspondientes gráficas. Como ejemplo de esta situación se muestra en las figuras 6.4, 6.5 y 6.6 se presentan las gráficas de la magnitud al cuadrado del vector de coeficientes del filtro adaptivo como resultado de igualar el canal  $H_4(z)$  con los algoritmos NLMS, FTF y LSL respectivamente respectivamente.

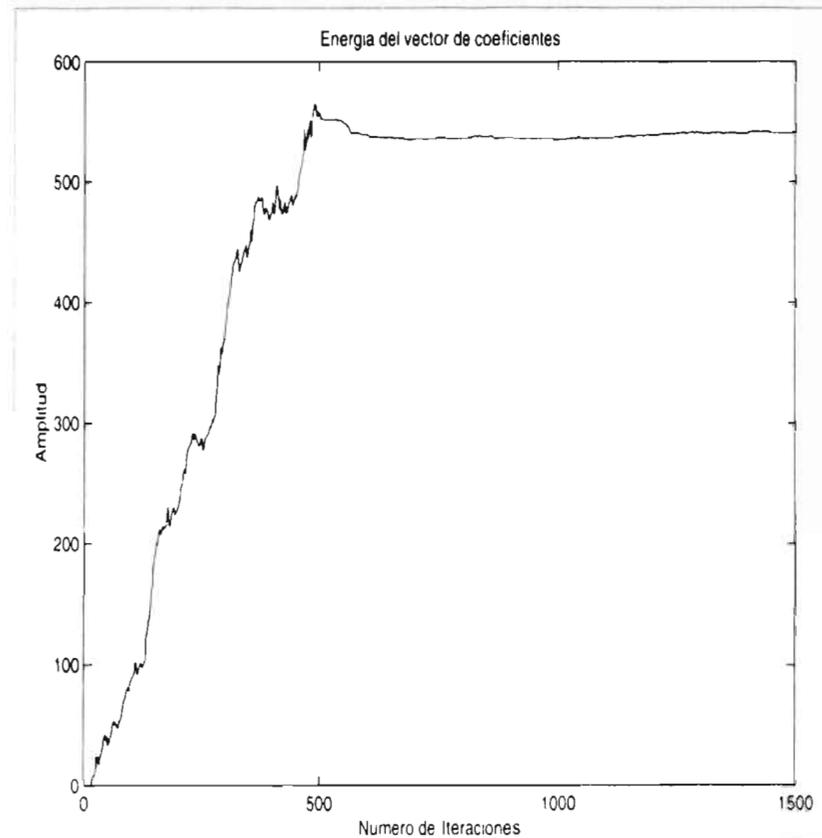


Figura 6.4: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo NLMS para igualar el canal  $H_4(z)$ .

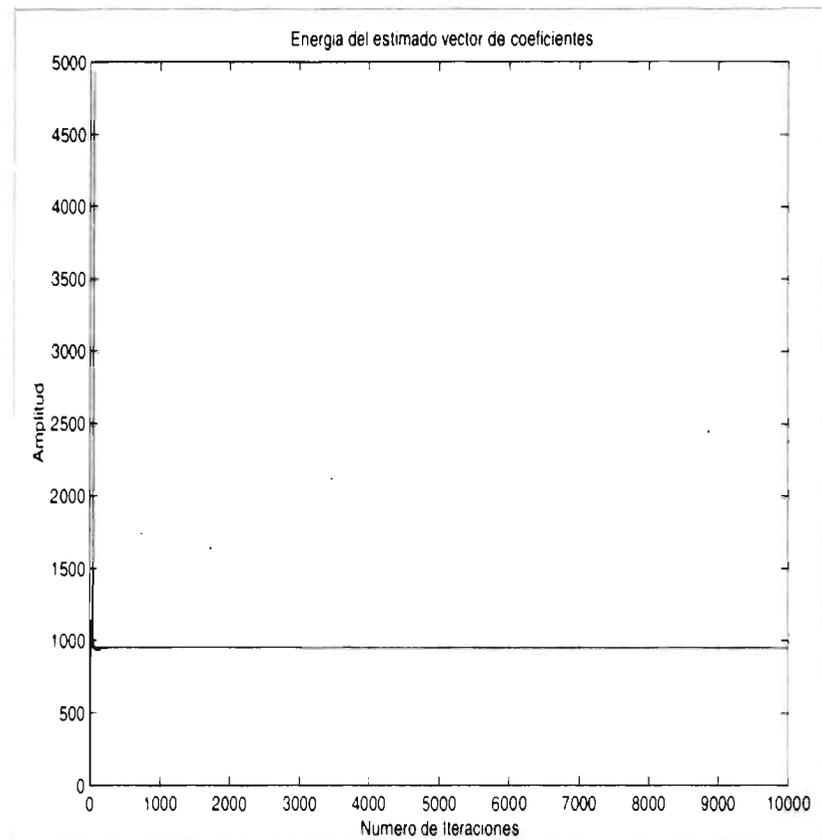
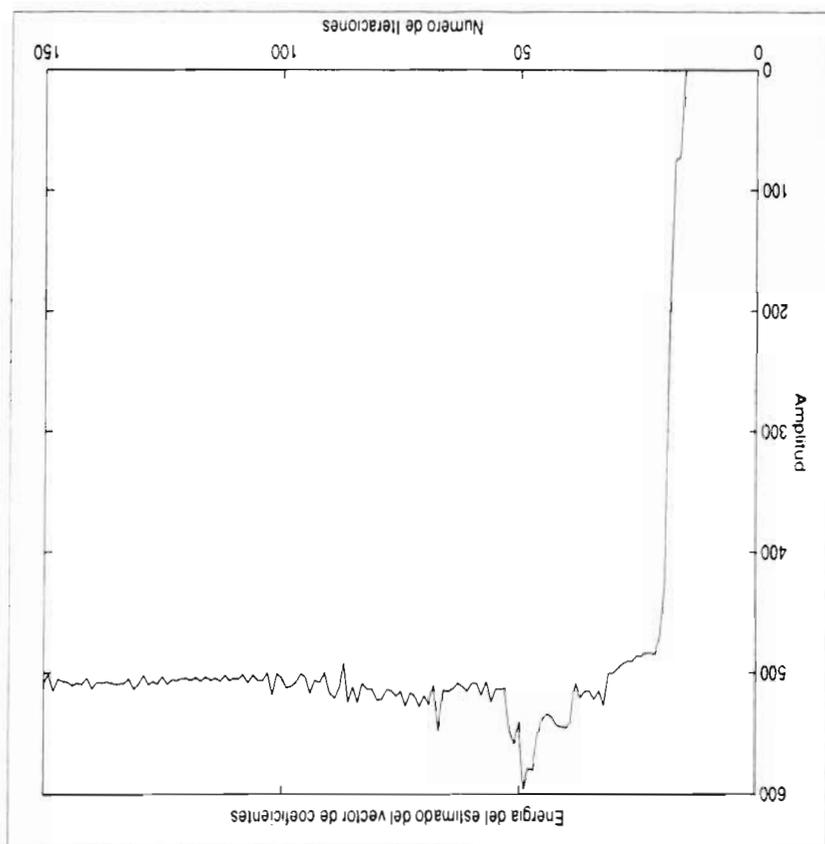


Figura 6.5: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo FTF para igualar el canal  $H_4(z)$ .

Figura 6.6: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo LSL para igualar el canal  $H_d(z)$



En todos estos casos se utilizó un filtro de orden 30 para igualar el canal. Obsérvese que para el algoritmo NLMS se utilizó la estrategia de utilizar un paso de adaptación de 1.0 para las primeras 500 iteraciones después de las cuales éste parámetro se cambió a 0.09, a pesar de lo cual el algoritmo converge de una manera mucho más lenta en comparación a los algoritmos FTF y LSL. Finalmente dada la impotancia práctica del canal  $H_6(z)$  se presentan las respectivas gráficas de la magnitud al cuadrado del vector de parámetros del filtro adaptivo para los tres algoritmos tratados en este capítulo.

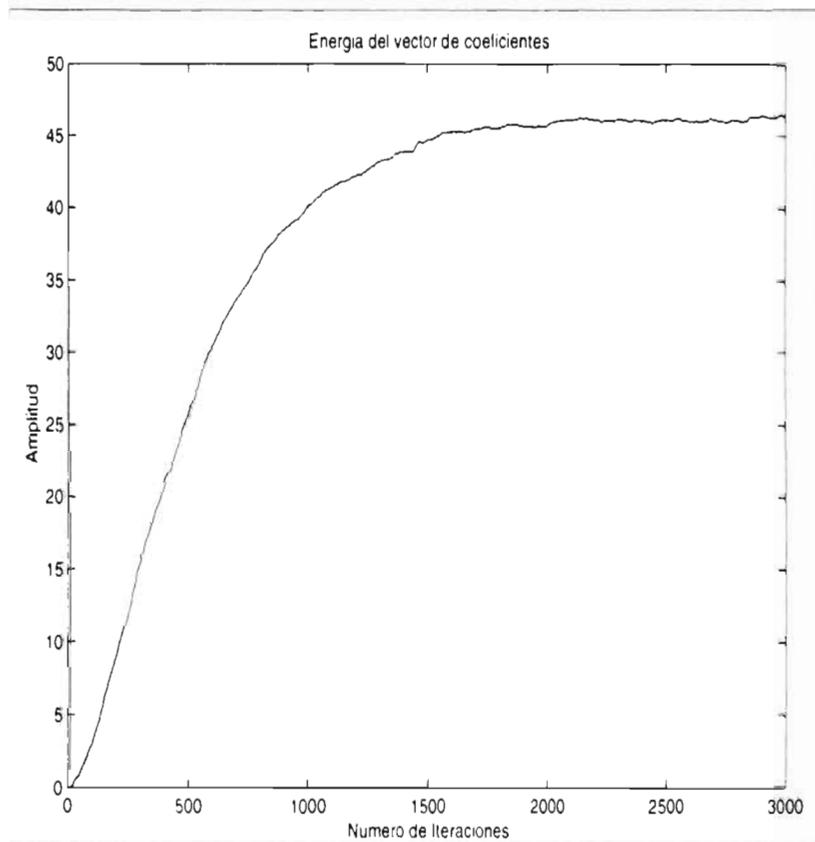


Figura 6.7: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo NLMS para igualar el canal  $H_6(z)$ .

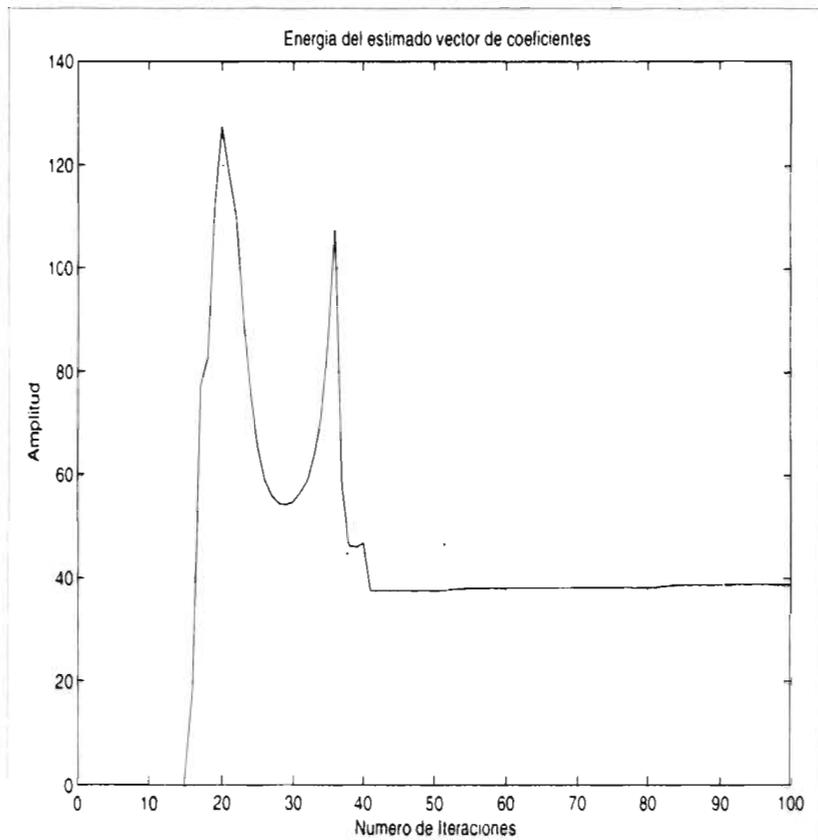


Figura 6.8: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo FTF para igualar el canal  $H_6(z)$ .

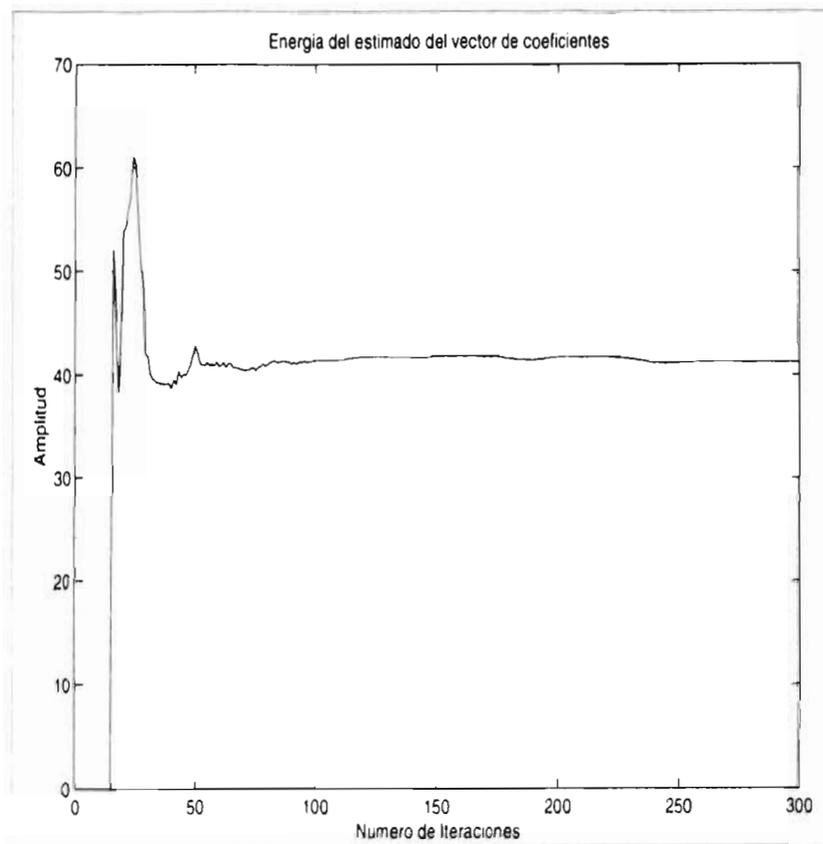


Figura 6.9: Magnitud al cuadrado del vector de coeficientes del filtro adaptivo cuando se utiliza el algoritmo LSL para igualar el canal  $H_6(z)$

En lo que respecta a las figs. 6.7, 6.8 y 6.9 para igualar el canal se utilizó un filtro adaptivo de orden 30 en los tres casos aquí tratados. El tamaño de paso de adaptación utilizado en el algoritmo NLMS fue de 0.09 y se mantuvo constante para todo el número de iteraciones realizadas. Como se puede observar de las gráficas los algoritmos FTF y LSL convergen alrededor de la iteración cien sino es que antes, mientras que el algoritmo NLMS converge alrededor de las 2000 iteraciones, de aquí que se recomienda utilizar el algoritmo FTF ó LSL para igualar canales de fase no mínima, sin embargo para efectuar una implantación práctica en punto fijo se debe de hacer un análisis de las propiedades numéricas de los algoritmos que se deseen implantar, lo cual sale fuera del objetivo de este trabajo.

## 6.2 Conclusiones

La diferencia fundamental en cuanto a los algoritmos tratados en este capítulo radica en el número de iteraciones con que cada uno converge. Por un lado, el algoritmo NLMS es el que más iteraciones requiere, mientras que los algoritmos FTF y LSL convergen en un número relativamente pequeño de iteraciones, lo que los vuelve de especial interés para el caso de aplicaciones en las que se requiere de enlaces de comunicaciones en tiempo real y que manejen altas tasas de transmisión.

Cabe señalar que la estructura "lattice" del filtro adaptivo que utiliza el algoritmo LSL posee mejores propiedades numéricas que las de un filtro transversal como el del algoritmo FTF; por lo cual el algoritmo LSL es más popular en las implantaciones prácticas que el algoritmo FTF a pesar de realizar más operaciones flotantes por muestra de salida. Sólo en los casos en que la velocidad de transmisión sea de esencial importancia se preferirá el algoritmo FTF, pero se deberá compensar sus propiedades numéricas más pobres mediante la utilización de un procesador de mayor longitud de palabra,[PM96].

**Parte IV**  
**Conclusiones y Perspectivas**

## Conclusiones

De las evaluaciones realizadas a los algoritmos es fácil darse cuenta que los canales que presenta mayor problema para lograr su igualación son aquéllos que presentan antiresonancias espectrales de consideración, las cuales están representadas por polos de magnitud cercana a uno ó mayor que uno. En caso de que los polos caigan dentro del círculo unitario, el problema de la igualación se resuelve de una manera satisfactoria aplicando un algoritmo que adapte los coeficientes de un filtro IIR, el precio a pagar lo constituye el gran número de operaciones a realizar el cual se debe fundamentalmente a la necesidad de verificar la estabilidad del filtro IIR. En caso de que algún polo caiga fuera del círculo unitario no es posible aplicar un algoritmo de este tipo por lo que habrá que recurrir a un algoritmo rápido.

En cuanto a los algoritmos que adaptan los coeficientes de un filtro IIR se comprobó a pesar de todas las especulaciones que se puedan encontrar en la literatura qué es necesario verificar la estabilidad del mismo. Por otro lado, el mecanismo más seguro para estabilizar el filtro consiste en proyectar las raíces dentro del círculo unitario dado que si se opta por reducir los parámetros de adaptación cada vez que se detecte un polo fuera del círculo unitario, se corre el riesgo de reducir a cero los parámetros de adaptación antes de que el algoritmo converga a la solución óptima.

Los algoritmos más adecuados para igualar canales de fase mínima son los rápidos dado que se puede aumentar el número de coeficientes sin aumentar de una manera exagerada el número de operaciones flotantes a realizar por muestra de salida como sucede en el caso del algoritmo NLMS.

En cuanto a los algoritmos en "lattice" estos son más adecuados para su implantación en procesadores digitales de punto fijo ya que la estructura "lattice" presenta mejor robustez a errores numéricos que la estructura transversal [?, PM96]. Sólo se preferirá la estructura transversal cuando la velocidad de transmisión sea una consideración esencial para el enlace de comunicaciones pero se deberá compensar su robustez numérica más pobre seleccionando un procesador digital de señales de mayor longitud de palabra, lo cual incrementará el costo del sistema.

# Perspectivas

El problema de la igualación de canal es un problema dinámico el cual a través del tiempo se ha vuelto más complejo debido a las nuevas tecnologías emergentes y a las crecientes necesidades de transmitir una mayor cantidad de información a través de un enlace de comunicaciones, el cual puede consistir desde una red telefónica hasta un enlace satelital. En la actualidad en los países industrializados se realizan grandes esfuerzos por desarrollar el teléfono de manos libres, las comunicaciones móviles personales via satelite, etc. Estas nuevas vertientes del desarrollo de las telecomunicaciones han impuesto nuevos retos a los ya existentes para poder concretar esta aplicaciones a un nivel comercial tan desarrollado como el de la telefonía tradicional.

La necesidad de la sociedad moderna por mayores cantidades de información lista para procesarse requiere forzosamente de la transmisión a altas tasas para poder disponer de dicha información en tiempo real. Debido a las restricciones tanto físicas como legales siempre se dispondrá de un ancho de banda limitado, lo que restringe demasiado la máxima tasa de transmisión y nos conduce a la necesidad de transmitir a tasas por arriba de las que permite el ancho de banda del canal. Esto ocasiona una seria interferencia intersímbolos en la recepción la cual impide al receptor la toma de una correcta decisión sobre el símbolo transmitido; para ello se hace necesario la igualación adaptiva del canal.

Los principales problemas de la igualación clásica de canal lo constituye el siguiente hecho [b5H94]:

- La mayoría de los canales utilizados para la transmisión de datos son de fase no mínima.

De acuerdo con esto el igualador ideal para estos canales es inestable. Este último hecho tiene profundas consecuencias para la igualación adaptiva pues en la mayoría de los casos no se podrá aplicar un esquema de igualación con retroalimentación debido a que forzosamente en algún instante del proceso de adaptación los polos abandonarán el círculo unitario. Lo que nos conduce a los algoritmos de estructura FIR, sin embargo estos requeriran de un gran número de coeficientes para poder igualar el canal de una manera satisfactoria lo cual tampoco es recomendable para el tipo de aplicaciones que se pretende en la actualidad.

Por otro lado considere la transmisión de señales en el marco de las comunicaciones móviles inalámbricas. En esta situación existe la propagación multitrayectoria, la cuál ha demostrado la necesidad de esquemas de igualación de canal en los que no

se utilice una secuencia de entrenamiento [bSH91]. Esto debido a que la propagación de una misma señal a través de trayectorias múltiples puede ocasionar una interferencia substractiva en el receptor ocasionando que la señal recibida sea muy débil, si esto se llega a presentar en el proceso de entrenamiento del canal la igualación que se logra es deficiente [bSH94]. Este fenómeno nos conduce a la búsqueda de métodos para igualar el canal en los que se precinda de la señal de entrenamiento, estos métodos reciben el nombre genérico de algoritmos de igualación ciega.

Los algoritmos de igualación ciega a pesar de su nombre necesitan cierto conocimiento de los estadísticos de las señales de entrada al canal. Se han hecho grandes esfuerzos en este sentido, sin embargo el problema más grave de estos algoritmos radica en el hecho de que para estimar los diversos momentos estadísticos hacen la suposición de que las señales transmitidas a través del canal son realizaciones de procesos estocásticos ergódicos. Esto implica que los promedios estadísticos se pueden aproximar utilizando promedios temporales, sin embargo ese es el problema: para que estos promedios temporales sean buenos en el sentido de que se aproximen a los promedios estadísticos se requiere promediar una gran cantidad de datos, lo cual incrementa de una manera excesiva la carga computacional en el receptor. De tal manera se tiene que recurrir a otro tipos de algoritmos ó a arquitecturas en paralelo.

Por otro lado, el profesor William A. Gardner y su equipo de investigadores en la Universidad de California en Davis, ha realizado grandes esfuerzos en el modelado de las señales que se utilizan en las comunicaciones como realizaciones de procesos cicloestacionarios [Gar86]. Los resultados obtenidos al modelar las señales moduladas mediante las distintas técnicas utilizadas en las comunicaciones digitales son bastante interesantes, ya que se logra una reducción en el orden de los estadísticos utilizados para identificar ó igualar el canal. Sin embargo, no es posible identificar la función de transferencia del canal utilizando solamente estadísticos de segundo orden sino que se hace necesario una interpolación para suplir la información faltante [LD91]; por lo tanto se requiere de estadísticos de orden superior.

Por último la teoría de la cicloestacionareidad es bastante interesante, sin embargo debido al desarrollo actual de la electrónica es difícil su aplicación y más si se trata de problemas en tiempo real en donde no se conocen las ciclo frecuencias de las señales transmitidas por el canal. Sin embargo en la gran mayoría de las aplicaciones en comunicaciones se conoce el tipo de modulación empleada para transmitir las señales, lo que ayuda a determinar de una manera a priori las ciclo frecuencias de interés. Este conocimiento se hace imperativo si se desea aplicar esta teoría a la igualación de canal, ya que en caso contrario sería necesario el cálculo del ciclo espectro sobre todo el rango de ciclo frecuencias, lo que computacionalmente es un proceso demasiado extensivo e indudablemente requerira del uso de arquitecturas en paralelo, lo cual encarecerá de una manera exagerada el costo del receptor de tal manera que resulta impráctica la aplicación de esta teoría.

Finalmente, la técnica más promisoría para la igualación ciega de canal es la "maximum likelihood sequence estimation (MLSE)". Esta técnica se puede combi-

nar con la "blind channel estimation" por medio del algoritmo de Viterbi[Se90]. Las técnicas de igualación basadas en la MLSE usualmente se desempeñan mejor que los igualadores lineales de canal y, pueden igualar canales con antiresonancias espectrales pronunciadas sin mayores problemas [bSH94]. El costo de estas técnicas es su complejidad lo que se traduce en la utilización de un mayor número de recursos para su implementación.

La aplicación de las técnicas anteriormente mencionadas depende de un mayor desarrollo de la electrónica de estado sólido y del procesamiento en paralelo. Por lo que corresponde al futuro inmediato, las técnicas basadas en la MLSE son las que tienen las mayores probabilidades de imponerse.

Por todo lo anterior, se puede concluir que la igualación de canal es un área de vital importancia en el desarrollo de los sistemas de comunicaciones actuales. Gran parte de los desarrollos futuros en el área de las comunicaciones dependen en gran medida de los logros que se puedan obtener en la resolución de este problema.

**Parte V**  
**Apéndices**

# Apéndice A

## Fundamentos del Algebra Lineal

La importancia del algebra lineal en el procesamiento digital de señales tiene como fundamento el hecho de que los algoritmos para procesar señales son en realidad procedimientos para manipular eficazmente vectores y matrices. Por tanto, desde el punto de vista del algebra lineal, los algoritmos para el procesamiento de señales pueden ser escritos en términos de operaciones matriciales. Estas operaciones matriciales están diseñadas para revelar la estructura subyacente de las observaciones y, extraer los estadísticos de mayor relevancia para derivar inferencias.

En este apéndice se hará una breve revisión de la teoría de matrices y del algebra lineal. Se comenzará con los espacios vectoriales para luego proseguir con matrices y transformaciones lineales. Se estudiará la estructura de los espacios Euclidianos y sus subespacios. El estudio de los eigen valores y los eigen vectores nos conducirá a la teoría espectral de las matrices Hermiteanas y a los operadores de proyección.

### A.1 Espacios Vectoriales

Sea  $\mathcal{R}^n$  el conjunto de todos los vectores reales de la forma  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ . El conjunto  $\mathcal{R}^n$  es un espacio vectorial lineal porque, para cualquier escalar real  $a \in \mathcal{R}$  y, para dos vectores cualesquiera  $\mathbf{x} \in \mathcal{R}^n$  y  $\mathbf{y} \in \mathcal{R}^n$ , los siguientes vectores también pertenecen a  $\mathcal{R}^n$ :

$$\begin{aligned} \mathbf{x} \in \mathcal{R}^n &\Rightarrow a\mathbf{x} = [ax_1, ax_2, \dots, ax_n]^T \in \mathcal{R}^n \\ \mathbf{x} \in \mathcal{R}^n, \mathbf{y} \in \mathcal{R}^n &\Rightarrow \mathbf{x} + \mathbf{y} = [x_1 + y_1, \dots, x_n + y_n]^T \in \mathcal{R}^n \end{aligned}$$

Se llama  $a\mathbf{x}$  multiplicación escalar y a  $\mathbf{x} + \mathbf{y}$  adición vectorial.

### A.1.1 Espacios Euclidianos.

Se define la norma Euclidea de cualquier vector  $\mathbf{x} \in \mathcal{R}^n$  como:

$$\|\mathbf{x}\| = (\mathbf{x}^T \mathbf{x})^{1/2} = \left( \sum_{i=1}^n x_i^2 \right)^{1/2} \quad (\text{A.1})$$

El espacio real lineal  $\mathcal{R}^n$ , con la norma Euclidea definida en A.1, es un espacio vectorial lineal normado que aquí llamaremos espacio Euclideo de dimensión  $n$ . Esta norma puede utilizarse para medir la distancia entre dos vectores  $\mathbf{x}$  y  $\mathbf{y}$  contenidos en  $\mathcal{R}^n$ :

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = [(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})]^{1/2} = \left[ \sum_{i=1}^n (x_i - y_i)^2 \right]^{1/2} \quad (\text{A.2})$$

Esta métrica satisface las condiciones [Sch91]:

1.  $\|\mathbf{x} - \mathbf{y}\| \geq 0, y \|\mathbf{x} - \mathbf{y}\| = 0 \rightarrow \mathbf{x} = \mathbf{y}$
2.  $\|\mathbf{x} - \mathbf{y}\| = \|\mathbf{y} - \mathbf{x}\|$  (*simetría*);
3.  $\|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{x} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{y}\|$  (*desigualdad del triángulo*);

Ahora  $\mathcal{R}^n$  es un espacio métrico [Sch91].

### A.1.2 Espacios de Hilbert

El producto interno entre dos vectores cualesquiera, no nulos, en  $\mathcal{R}^n$  se define como:

$$\mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i \quad (\text{A.3})$$

Si  $\mathbf{x}^T \mathbf{y} = 0$ , se dice que  $\mathbf{x}$  y  $\mathbf{y}$  son ortogonales. Todo espacio vectorial lineal normado en el que toda secuencia de Cauchy converge y en donde además está definido el producto interno, se llama espacio de Hilbert [Sch91].

### A.1.3 Matrices.

Los vectores  $\mathbf{x}_i$  los agrupamos en matrices  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$  y decimos que  $\mathbf{X}$  pertenece al espacio vectorial lineal  $\mathcal{R}^{n \times p}$ . Este espacio vectorial lineal contiene todas las matrices reales que consisten de  $p$  vectores de dimensión  $n$  o, de una manera equivalente, de  $np$  escalares de la forma  $\mathbf{X} = \{x_{ij}\}$ . Este espacio vectorial lineal es cerrado bajo la multiplicación por escalar y la adición de matrices [Sch91].

## A.2 Independencia Lineal

Si la suma  $a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \dots + a_n\mathbf{x}_n$  es igual a cero sólo cuando  $a_1 = a_2 = \dots = a_n = 0$ , entonces decimos que los vectores  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  son linealmente independientes. En caso de que uno ó más de los coeficientes  $a_k$  sea diferente de cero se dice que los vectores son linealmente dependientes [Str88].

### A.2.1 Determinante Gramiano

La matriz Gramiana para la matriz de datos  $\mathbf{X}$  se define como [Sch91]:

$$\mathbf{G} = \mathbf{X}^T \mathbf{X} = \{g_{ij}\} \quad (\text{A.4})$$

donde:  $g_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ . El determinante de la matriz Gramiana se llama Determinante Gramiano. Su importancia radica en el siguiente teorema:

**Teorema A.1** Los vectores  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$  son linealmente independientes si el determinante de Gram es diferente de cero.

## A.3 Factores QR

Existe otra forma de saber si un conjunto de  $p$  vectores es o no linealmente independiente: se transforma al conjunto de  $p$  vectores en otro conjunto equivalente de vectores ortogonales para saber cuantos vectores ortogonales distintos de cero pueden ser generados. Esta idea nos conduce al estudio de esquemas de ortogonalización, o factorización **QR** de matrices de datos. Al referirnos los factores **QR** de una matriz  $\mathbf{X}$ , tenemos en mente la siguiente representación:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p] = \mathbf{Q}\mathbf{R} \\ &= \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_p \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1p} \\ & r_{22} & & \\ & & \ddots & \vdots \\ & & & r_{pp} \end{bmatrix} \end{aligned} \quad (\text{A.5})$$

En esta representación los vectores  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  son linealmente independientes, y los vectores  $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$  son ortogonales. De aquí que  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$ , por lo que  $\mathbf{Q}$  es una matriz ortogonal [Wat91].

Antes de proceder al estudio de los algoritmos para la factorización **QR** de una matriz de datos, recordemos las siguientes interpretaciones del factor  $\mathbf{X} = \mathbf{Q}\mathbf{R}$ :

- $\mathbf{X} = \mathbf{Q}\mathbf{R}$ : La forma de síntesis de la factorización **QR** que muestra como las columnas ortogonales de  $\mathbf{Q}$  son utilizadas para formar las columnas linealmente independientes de  $\mathbf{X}$ ;

- $\mathbf{Q} = \mathbf{X}\mathbf{R}^{-1}$ : La forma de análisis de la factorización **QR** que muestra como las columnas linealmente independientes de  $\mathbf{X}$  son analizadas para producir las columnas ortogonales de  $\mathbf{Q}$ ;
- $\mathbf{Q}^T\mathbf{X} = \mathbf{R}$ : La forma de correlación cruzada de la factorización **QR** que muestra como la matriz de datos  $\mathbf{X}$  al ser premultiplicada por la matriz ortogonal  $\mathbf{Q}^T$  es transformada a una matriz triangular superior; también muestra que los vectores ortogonales  $\mathbf{u}_i$  están causalmente decorrelacionados con los vectores linealmente independientes  $\mathbf{x}_j$ :  $\mathbf{u}_i^T\mathbf{x}_j = 0$  para  $i > j$  [Sch91];
- $\mathbf{X}^T\mathbf{X} = \mathbf{R}^T\mathbf{R}$ : La matriz Gramiana de  $\mathbf{X}$  es igual a los factores  $\mathbf{R}^T\mathbf{R}$  de una descomposición **LU** [Sch91].

### A.3.1 Procedimiento de Gram-Schmidt

El procedimiento de Gram-Schmidt comienza con la inicialización [Wat91]:

$$\mathbf{y}_1 = \mathbf{x}_1; \quad \mathbf{u}_1 = \frac{\mathbf{y}_1}{\|\mathbf{y}_1\|} \quad (\text{A.6})$$

Construyase  $\mathbf{u}_2$  como sigue:

$$\mathbf{y}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2^T\mathbf{y}_1}{\|\mathbf{y}_1\|^2}\mathbf{y}_1 = \mathbf{x}_2 - \mathbf{x}_2^T\mathbf{u}_1\mathbf{u}_1; \quad \mathbf{u}_2 = \frac{\mathbf{y}_2}{\|\mathbf{y}_2\|} \quad (\text{A.7})$$

El vector  $\mathbf{u}_2 = \mathbf{0}$  si  $\mathbf{x}_2 = a\mathbf{x}_1$ , lo cual significa que  $\mathbf{x}_2$  es linealmente dependiente con respecto a  $\mathbf{x}_1$ . Asíumase que  $\mathbf{u}_2 \neq \mathbf{0}$ , entonces  $\mathbf{u}_1$  y  $\mathbf{u}_2$  son linealmente y ortogonales [Wat91]:

$$\|\mathbf{y}_2\| \|\mathbf{y}_1\| \|\mathbf{u}_2^T\mathbf{u}_1\| = \mathbf{y}_2^T\mathbf{y}_1 = \mathbf{x}_2^T\mathbf{y}_1 - \frac{\mathbf{x}_2^T\mathbf{y}_1}{\|\mathbf{y}_1\|^2}\mathbf{y}_1^T\mathbf{y}_1 = 0 \quad (\text{A.8})$$

Continúe este procedimiento, generando cada nuevo vector  $\mathbf{u}_k$  como sigue [Wat91]:

$$\mathbf{y}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} (\mathbf{x}_k^T\mathbf{u}_i)\mathbf{u}_i; \quad \mathbf{u}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|} \quad (\text{A.9})$$

El vector  $\mathbf{u}_k = \mathbf{0}$  si  $\mathbf{x}_k$  es una combinación lineal de  $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k-1})$  [Wat91].

Este procedimiento mapea los vectores  $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \dots, \mathbf{x}_p)$  en los vectores  $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p)$ . En el último conjunto de vectores habrá  $r$  vectores diferentes de cero que son linealmente independientes y ortogonales y  $(p - r)$  vectores iguales a cero. Para cada  $\mathbf{u}_i$  que es cero, concluimos que el correspondiente  $\mathbf{x}_i$  es linealmente dependiente.

Cuando el proceso de Gram-Schmidt termina, tenemos la siguiente repre-

sentación para los vectores  $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p)$ :

$$\begin{aligned} \mathbf{Q} &= \mathbf{X}\mathbf{R}^{-1} \\ \mathbf{Q} &= [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p] \quad \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p] \\ \mathbf{R}^{-1} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ & a_{22} & & \\ & & \ddots & \vdots \\ & & & \end{bmatrix} \end{aligned} \quad (\text{A.10})$$

Se dice que la matriz de datos es analizada para producir los vectores ortogonales  $\mathbf{u}_i$ . Cada  $a_{ii}$  es diferente de cero siempre y cuando  $\mathbf{x}_i$  sea linealmente independiente de  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{i-1})$ .

Existen otras formas de obtener  $\mathbf{Q}$ ,  $\mathbf{R}$  o  $\mathbf{R}^{-1}$ , entre las cuales se encuentran las transformaciones de Householder y Givens. Estas son las más populares ya que generan la matriz  $\mathbf{Q}$  mediante una secuencia de sencillas transformaciones ortogonales [Wat91].

### A.3.2 Transformación Householder.

Asumamos que la matriz de datos  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  tiene  $n$  columnas linealmente independientes  $\mathbf{x}_i$ , y representemos la factorización  $\mathbf{QR}$  de la matriz de datos como:

$$\mathbf{Q}^T \mathbf{X} = \mathbf{R} \quad (\text{A.11})$$

La transformación Householder encuentra una representación para la matriz  $\mathbf{Q}^T$  que transforma columna por columna a la matriz de datos  $\mathbf{X}$  en una matriz triangular superior [Sch91], de tal manera que  $\mathbf{Q}^T$  se pueda expresar como:

$$\mathbf{Q}^T = \mathbf{Q}_n^T \mathbf{Q}_{n-1}^T \dots \mathbf{Q}_1^T \quad (\text{A.12})$$

donde cada una de las transformaciones  $\mathbf{Q}_i^T$  es ortogonal:  $\mathbf{Q}_i^T \mathbf{Q}_i = \mathbf{I}$ . Al realizar esta transformación, la matriz  $\mathbf{Q}$  estará almacenada implícitamente en las matrices  $\mathbf{Q}_i$ .

La matriz  $\mathbf{Q}_k^T$  tiene la siguiente forma [Sch91]:

$$\mathbf{Q}_k^T = \left[ \begin{array}{c|c} \mathbf{I}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & \hat{\mathbf{Q}}_k^T \end{array} \right] \quad (\text{A.13})$$

donde  $\hat{\mathbf{Q}}_k^T$  es tal que:

$$\hat{\mathbf{Q}}_k^T \hat{\mathbf{x}}_k = \begin{bmatrix} \mathbf{r}_{kk} \\ \mathbf{0} \end{bmatrix} \quad (\text{A.14})$$

donde  $\mathbf{r}_{kk}$  es el vector columna de  $k \times 1$  y  $\mathbf{0}$  es el vector nulo de  $(n-k) \times 1$ . Esto es  $\hat{\mathbf{Q}}_k^T$  produce ceros en la  $k$ -ésima columna de  $\mathbf{X}$  sin alterar las columnas anteriores, las cuales tienen la forma [Sch91]:

$$\begin{bmatrix} \mathbf{r}_l \\ \mathbf{0} \end{bmatrix}$$

donde  $l < r$  y  $\mathbf{0}$  es el vector nulo de  $n-l \times 1$ . Se puede demostrar que  $\mathbf{Q}_k^T$  está dada por [Sch91]:

$$\hat{\mathbf{Q}}_k^T = \mathbf{I} - \frac{2\mathbf{v}_k\mathbf{v}_k^T}{\mathbf{v}_k^T\mathbf{v}_k} \quad (\text{A.15})$$

$$\mathbf{v}_k = \hat{\mathbf{x}}_k - \|\hat{\mathbf{x}}_k\| \mathbf{e}_1$$

donde  $\mathbf{e}_1 = [1, 0, \dots, 0]$ .

### A.3.3 Transformación de Givens

La secuencia de las transformaciones de Householder convierten columna por columna a la matriz de datos  $\mathbf{X}$  en la matriz triangular superior  $\mathbf{R}$ ; mientras que las rotaciones de Givens efectúan dicha transformación elemento por elemento [Sch91]. La idea básica es la misma: transformar la matriz  $\mathbf{X}$  en la matriz  $\mathbf{R}$  mediante una secuencia de transformaciones ortogonales. El esquema consiste en introducir ceros elemento por elemento para completar un renglón antes de proceder con el siguiente:

$$\begin{aligned} \mathbf{Q}_{21}^T \mathbf{X} &= \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \\ \mathbf{Q}_{31}^T \mathbf{Q}_{21}^T \mathbf{X} &= \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ * & * & * & * \end{bmatrix} \\ \mathbf{Q}_{32}^T \mathbf{Q}_{31}^T \mathbf{Q}_{21}^T \mathbf{X} &= \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ * & * & * & * \end{bmatrix} \\ \mathbf{Q}_{41}^T \mathbf{Q}_{32}^T \mathbf{Q}_{31}^T \mathbf{Q}_{21}^T \mathbf{X} &= \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & * & * & * \end{bmatrix} \end{aligned} \quad (\text{A.16})$$

De este patrón, se observa que  $\mathbf{Q}^T \mathbf{X} = \mathbf{R}$  se puede expresar como:

$$\prod_{i=2}^n \prod_{j=1}^{i-1} \mathbf{Q}_{ij}^T \mathbf{X} = \mathbf{R} \quad (\text{A.17})$$

La matriz  $\mathbf{Q}_{ij}^T$  transforma en ceros el  $j$ -ésimo elemento del renglón  $i$ , o bien el  $i$ -ésimo elemento de la columna  $j$ ,  $i > j$ :

$$\mathbf{Q}_{ij}^T \mathbf{x}_j = \begin{bmatrix} x_{1j} \\ \vdots \\ x_{jj} \\ \vdots \\ 0 \\ \vdots \\ x_{nj} \end{bmatrix} \quad (\text{A.18})$$

donde el rotador de Givens,  $\mathbf{Q}_{ij}^T$ , está dado por:

$$\mathbf{Q}_{ij}^T = \begin{bmatrix} \mathbf{I} & & & & \\ & \cos(\theta_{ij}) & & \sin(\theta_{ij}) & \\ & & \mathbf{I} & & \\ & -\sin(\theta_{ij}) & & \cos(\theta_{ij}) & \\ & & & & \mathbf{I} \end{bmatrix} \quad (\text{A.19})$$

$$\cos(\theta_{ij}) = \frac{x_{jj}}{(x_{jj}^2 + x_{ij}^2)^{1/2}} \quad \sin(\theta_{ij}) = \frac{x_{ij}}{(x_{jj}^2 + x_{ij}^2)^{1/2}}$$

Obsérvese que  $\mathbf{Q}_{ij}^T$  es ortogonal y que preserva los ceros inducidos anteriormente:

$$\mathbf{Q}_{ij}^T \mathbf{Q}_{ij} = \mathbf{I} \quad (\text{A.20})$$

## A.4 Espacios Lineales

Sea  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$  un conjunto de vectores donde  $\mathbf{x}_i \in \mathbb{R}^n$ . El **espacio generado** por estos vectores es el conjunto de todos los vectores,  $\mathbf{x} \in \mathbb{R}^n$ , que pueden ser generados mediante una combinación lineal del conjunto [Str88]:

$$S(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p) = \left\{ \mathbf{x} \mid \mathbf{x} = \sum_{i=1}^p a_i \mathbf{x}_i \right\} \quad (\text{A.21})$$

El espacio generado  $S$  es un subespacio vectorial lineal de  $\mathbb{R}^n$ . En este apéndice se tomarán los términos "espacio generado" y subespacio como equivalentes.

### A.4.1 Bases

Si todos los vectores dependientes en el conjunto  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$  son desechados (utilizando el algoritmo de ortogonalización Gram-Schmidt), y dejamos todos los vectores linealmente independientes, entonces estos vectores forman una **base** del

subespacio  $S$ . Sea esta base  $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(r)})$ ,  $r \leq p$ , entonces cada vector del subespacio  $S$  podrá ser escrito como la suma [Str88]:

$$\mathbf{x} = \sum_{i=1}^r a_i \mathbf{x}_{(i)} = \mathbf{X}\mathbf{a}; \quad \mathbf{X} = [\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(r)}] \quad (\text{A.22})$$

#### A.4.2 Subespacios Directos

Sea  $\{\mathbf{x}_{(i)}\}_1^r$  un conjunto de vectores linealmente independientes, el cuál puede ser dividido en  $k$  conjuntos disjuntos de vectores. Cada conjunto disjunto genera su respectivo subespacio  $S_j$  (ó forma una base para el espacio que genera). Por lo tanto, el subespacio  $S$  puede ser escrito como una suma directa de subespacios disjuntos [Sch91]:

$$S = S_1 \oplus S_2 \oplus \dots \oplus S_k$$

$$S_j = \{\mathbf{x} \mid \mathbf{x} = \sum_{i \in I_j} a_i \mathbf{x}_{(i)}\}; \quad I_j: j\text{-ésimo subconjunto disjunto de } \{1, 2, \dots, r\} \quad (\text{A.23})$$

De manera similar, si los vectores ortogonales  $\{\mathbf{u}_{(i)}\}_1^r$  son divididos en subconjuntos, entonces  $S$  puede ser expresada como la suma directa de los subespacios ortogonales [Sch91]:

$$S = U_1 \oplus U_2 \oplus \dots \oplus U_k \quad (\text{A.24})$$

$$U_j = \{\mathbf{x} \mid \mathbf{x} = \sum_{i \in I_j} b_i \mathbf{u}_{(i)}\}$$

#### A.4.3 Dimensión, Rango y Nulidad

Sea  $S$  el subespacio generado por los vectores  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$ :

$$S(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p) = \left\{ \mathbf{x} \mid \mathbf{x} = \sum_{i=1}^p a_i \mathbf{x}_i \right\} \quad (\text{A.25})$$

Los vectores  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$  pueden ser organizados en la matriz de datos  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$ , y el subespacio  $S$  puede ser representado por:

$$\mathbf{R}(\mathbf{X}) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{X}\mathbf{a}\} \quad (\text{A.26})$$

Este conjunto se conoce con el nombre de **rango** de la transformación lineal  $\mathbf{x} = \mathbf{X}\mathbf{a}$ , y, lo denotamos por  $\mathbf{R}(\mathbf{X})$  o  $\langle \mathbf{X} \rangle$  [Sch91]. Por tanto,  $S = \mathbf{R}(\mathbf{X}) = \langle \mathbf{X} \rangle$ . La dimensión del subespacio  $S$ ,  $\dim S$ , es el número máximo de vectores linealmente independientes en el conjunto  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$ . La dimensión del subespacio  $S$  es a su vez igual al número de columnas linealmente independientes de la matriz  $\mathbf{X}$  [Str88]. Por tanto, llamamos a éste número el **rango** o **clase** de  $\mathbf{X}$  y lo denotamos con  $\rho(\mathbf{X})$ . Por lo tanto, tenemos la identidad:

$$\dim S = \dim \mathbf{R}(\mathbf{X}) = \rho(\mathbf{X}) \quad (\text{A.27})$$

Definimos como el **nulo** de la transformación lineal  $\mathbf{x} = \mathbf{X}\mathbf{a}$ , al conjunto de vectores  $\mathbf{a}$  tales que  $\mathbf{X}\mathbf{a} = \mathbf{0}$ :

$$N(\mathbf{X}) = \{\mathbf{a} \mid \mathbf{X}\mathbf{a} = \mathbf{0}\} \quad (\text{A.28})$$

Asúmase que existe una base para  $N(\mathbf{X})$  de la forma  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{p-r})$ :

$$N(\mathbf{X}) = \left\{ \mathbf{a} \mid \mathbf{a} = \sum_{i=1}^{p-r} \alpha_i \mathbf{a}_i \right\} \quad (\text{A.29})$$

$$\dim N(\mathbf{X}) = p - r$$

Las dimensiones de  $R(\mathbf{X})$  y de  $N(\mathbf{X})$  están relacionadas mediante la ley de la Nulidad de Sylvester [Sch91]:

$$\dim R(\mathbf{X}) = r$$

$$\dim N(\mathbf{X}) = p - r \quad (\text{A.30})$$

$$\dim R(\mathbf{X}) + \dim N(\mathbf{X}) = p$$

## A.5 Matrices Hermiteanas

Una matriz Hermiteana,  $\mathbf{R}$ , es igual al complejo conjugado de su transpuesta:  $(\mathbf{R}^T)^* = \mathbf{R}$ . Para simplificar la notación hacemos  $(\mathbf{R}^T)^* = \mathbf{R}^H$ . Existe un número importante de resultados relativos a las matrices Hermiteanas, aquí solo mencionaremos los más relevantes para los fines del presente trabajo. Para el desarrollo de estos resultados véase [Wat91]:

- Los **Eigenvalores** de una matriz Hermiteana son reales.
- Los **Eigenvectores** de una matriz Hermiteana son ortogonales.
- Las matrices Hermiteanas son diagonalizables.

## A.6 Descomposición en Valores Singulares

La descomposición en valores singulares (DVS) es una poderosa representación para las matrices rectangulares. La DVS diagonaliza simultáneamente la matriz Gramiana  $\mathbf{H}^H \mathbf{H}$  y el producto exterior  $\mathbf{H} \mathbf{H}^H$  [Sch91]. Esta última propiedad garantiza que la matriz Gramiana  $\mathbf{H}^H \mathbf{H}$  y la proyección  $\mathbf{H} (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$  sean simultáneamente diagonalizables [Sch91].

Considere una matriz  $\mathbf{H}$  de  $N \times p$  y, su matriz Gramiana  $\mathbf{G} = \mathbf{H}^H \mathbf{H}$ . La matriz Gramiana es Hermiteana y definida no negativa [Sch91]. Sean  $\lambda_1^2, \dots, \lambda_p^2$  los eigenvalores de  $\mathbf{G}$  (también tenemos que:  $\lambda_1, \dots, \lambda_p$  son los valores singulares de  $\mathbf{H}$ ). Del teorema de Gram se tiene que el rango de la matriz  $\mathbf{G}$  es igual al rango de  $\mathbf{H}$  [Sch91]. Asumamos que  $r \leq p$ . Entonces:

$$\lambda_1^2 \geq \lambda_2^2 \geq \dots \geq \lambda_r^2 > 0; \quad \lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_p = 0 \quad (\text{A.31})$$

Los resultados de la sección A.5 pueden ser utilizados para representar la matriz Gramiana como:

$$\mathbf{G} = \left[ \mathbf{V}_1 \mid \mathbf{V}_2 \right] \left[ \begin{array}{c|c} \Lambda_1^2 & \mathbf{0} \\ \hline \mathbf{0} & \Lambda_2^2 \end{array} \right] \left[ \begin{array}{c} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{array} \right] \quad (\text{A.32})$$

$$\mathbf{V}_1^H \mathbf{G} \mathbf{V}_1 = \Lambda_1^2; \quad \mathbf{V}_1^H \mathbf{V}_1 = \mathbf{I}; \quad \Lambda_1^2 = \text{diag} [\lambda_1^2 \cdots \lambda_r^2]$$

$$\mathbf{V}_2^H \mathbf{G} \mathbf{V}_2 = \Lambda_2^2; \quad \mathbf{V}_2^H \mathbf{V}_2 = \mathbf{I}; \quad \Lambda_2^2 = \text{diag} [0 \cdots 0]$$

$$\mathbf{V}_2^H \mathbf{V}_1 = \mathbf{0}$$

Estas ecuaciones pueden organizarse en la ecuación [Sch91]:

$$\left[ \begin{array}{c} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{array} \right] \mathbf{H}^H \mathbf{H} \left[ \mathbf{V}_1 \mid \mathbf{V}_2 \right] = \left[ \begin{array}{c|c} \Lambda_1^2 & \mathbf{0} \\ \hline \mathbf{0} & \Lambda_2^2 \end{array} \right] \quad (\text{A.33})$$

Se sigue que  $\mathbf{H}$  es una matriz de la forma [Sch91]:

$$\mathbf{H} \left[ \mathbf{V}_1 \mid \mathbf{V}_2 \right] = \left[ \mathbf{U}_1 \mid \mathbf{U}_2 \right] \left[ \begin{array}{c|c} \Lambda_1 & \mathbf{0} \\ \hline \mathbf{0} & \Lambda_2 \end{array} \right] \quad (\text{A.34})$$

donde  $\mathbf{U}_1$  y  $\mathbf{U}_2$  son matrices unitarias:

$$\left[ \begin{array}{c} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{array} \right] \left[ \mathbf{U}_1 \mid \mathbf{U}_2 \right] = \left[ \begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{array} \right] \quad (\text{A.35})$$

Se resuelve para  $\mathbf{H}$  como sigue:

$$\begin{aligned} \mathbf{H} &= \left[ \mathbf{U}_1 \mid \mathbf{U}_2 \right] \left[ \begin{array}{c|c} \Lambda_1 & \mathbf{0} \\ \hline \mathbf{0} & \Lambda_2 \end{array} \right] \left[ \begin{array}{c} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{array} \right] \\ &= \mathbf{U} \Lambda \mathbf{V}^H \end{aligned} \quad (\text{A.36})$$

$$\mathbf{U} = \left[ \mathbf{U}_1 \mid \mathbf{U}_2 \right]; \quad \mathbf{V} = \left[ \mathbf{V}_1 \mid \mathbf{V}_2 \right]$$

Esta es la descomposición en valores singulares de  $\mathbf{H}$ . La matriz  $\mathbf{U}$  es de  $N \times p$  y  $\mathbf{U}_1$  es de  $N \times r$ . La matriz  $\mathbf{V}^H$  es de  $p \times p$  y  $\mathbf{V}_1$  es de  $r \times p$ . La matriz  $\Lambda$  es de  $p \times p$  y  $\Lambda_1$  es una matriz diagonal, de  $r \times r$ , cuyos elementos son los valores singulares distintos de cero de la matriz  $\mathbf{H}$ . Obsérvese que las matrices  $\mathbf{H}^H \mathbf{H}$  y  $\mathbf{H} \mathbf{H}^H$  se pueden expresar

como [Sch91]:

$$\begin{aligned}
\mathbf{H}^H \mathbf{H} &= \mathbf{V}_1 \Lambda_1^2 \mathbf{V}_1^H + \mathbf{V}_2 \Lambda_2^2 \mathbf{V}_2^H \\
&= \mathbf{V}_1 \Lambda_1^2 \mathbf{V}_1^H \\
\mathbf{H} \mathbf{H}^H &= \mathbf{U}_1 \Lambda_1 \mathbf{V}_1^H \mathbf{V}_1 \Lambda_1 \mathbf{U}_1^H + \mathbf{U}_2 \Lambda_2 \mathbf{V}_2^H \mathbf{V}_2 \Lambda_2 \mathbf{U}_2^H \\
&= \mathbf{U}_1 \Lambda_1^2 \mathbf{U}_1^H
\end{aligned} \tag{A.37}$$

De aquí se diga que la descomposición en valores singulares diagonaliza simultáneamente el producto interno  $\mathbf{H}^H \mathbf{H}$  y el producto externo  $\mathbf{H} \mathbf{H}^H$  [Sch91].

### A.6.1 Rango y Espacio Nulo

Recuérdese la definición de rango y espacio nulo:

$$\begin{aligned}
1. R(\mathbf{H}) &= \{\mathbf{x} \mid \mathbf{x} = \mathbf{H}\theta\} \\
2. N(\mathbf{H}) &= \{\theta \mid \mathbf{x} = \mathbf{H}\theta = \mathbf{0}\}
\end{aligned} \tag{A.38}$$

De la descomposición en valores singulares de  $\mathbf{H}$ , podemos expresar estos espacios como [Sch91]:

$$\begin{aligned}
1. R(\mathbf{H}) &= \{\mathbf{x} \mid \mathbf{x} = \mathbf{U} \Lambda \mathbf{V}^H \theta\} \\
&= \{\mathbf{x} \mid \mathbf{x} = \mathbf{U}_1 \Phi\} \\
&= R(\mathbf{U}_1) \\
2. N(\mathbf{H}) &= \{\theta \mid \mathbf{x} = \mathbf{U} \Lambda \mathbf{V}^H \theta = \mathbf{0}\} \\
&= \{\theta \mid \mathbf{V}_1^H \theta = \mathbf{0}\} \\
&= N(\mathbf{V}_1^H)
\end{aligned} \tag{A.39}$$

Se dice que el rango de  $\mathbf{H}$  es igual al rango de  $\mathbf{U}_1$  y, que el espacio nulo de  $\mathbf{H}$  es igual al espacio nulo de  $\mathbf{V}_1^H$  [Sch91]. La dimensión del rango es  $r$ , y la dimensión del espacio nulo es  $p - r$ .

### A.6.2 Aproximación de Bajo Rango

Sea  $\mathbf{H}$  una matriz de rango  $p$ , la cual se aproxima mediante una matriz de rango  $r$ ; para ello se define el siguiente error entre  $\mathbf{H}$  y su aproximación [Sch91]:

$$\epsilon^2 = \text{tr}(\mathbf{H} - \mathbf{H}_r)^H (\mathbf{H} - \mathbf{H}_r) \quad (\text{A.40})$$

Este error se conoce como la norma de Frobenius de  $\mathbf{H} - \mathbf{H}_r$ , la cual es la suma del cuadrado de todos los elementos [Sch91]. Sea  $\mathbf{H}_r = \mathbf{U}\Sigma_r\mathbf{V}^H$ , donde  $\Sigma_r = \text{diag}[\lambda_1 \cdots \lambda_r 0 \cdots 0]$ . Entonces:

$$\begin{aligned} \epsilon^2 &= \text{tr}\mathbf{V}(\Sigma - \Sigma_r)\mathbf{U}^H\mathbf{U}(\Sigma - \Sigma_r)\mathbf{V}^H \\ &= \sum_{r+1}^p \lambda_i^2 \end{aligned} \quad (\text{A.41})$$

El error  $\mathbf{H} - \mathbf{H}_r$  es ortogonal a  $\mathbf{H}_r$ :

$$\begin{aligned} (\mathbf{H} - \mathbf{H}_r)\mathbf{H}_r^H &= \mathbf{U}(\Sigma - \Sigma_r)\mathbf{V}^H\mathbf{V}\Sigma_r\mathbf{U}^H \\ &= \mathbf{U}(\Sigma - \Sigma_r)\Sigma_r\mathbf{U}^H = \mathbf{0} \end{aligned} \quad (\text{A.42})$$

Esto hace de  $\mathbf{H}_r = \mathbf{U}\Sigma_r\mathbf{V}^H$  es la aproximación por mínimos cuadrados, de rango  $r$ , a la matriz  $\mathbf{H}$  [Sch91].

### A.6.3 Descomposición de la Matriz Identidad

Considere la descomposición de la matriz  $\mathbf{S}$ :

$$\mathbf{S} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \quad (\text{A.43})$$

El Gramiano de  $\mathbf{S}$  es [Sch91]:

$$\begin{aligned} \mathbf{S}^H\mathbf{S} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^2 & \mathbf{0} \\ \mathbf{0} & \Sigma_2^2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \\ \mathbf{S}\mathbf{S}^H &= \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^2 & \mathbf{0} \\ \mathbf{0} & \Sigma_2^2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{bmatrix} \end{aligned} \quad (\text{A.44})$$

Por tanto, la matriz identidad se puede escribir como [Sch91]:

$$\begin{aligned} \mathbf{I} &= \mathbf{S}(\mathbf{S}^H\mathbf{S})^{-1}\mathbf{S}^H = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{bmatrix} \\ &= \mathbf{U}_1\mathbf{U}_1^H + \mathbf{U}_2\mathbf{U}_2^H \end{aligned} \quad (\text{A.45})$$

Este resultado es una forma más de expresar que los subespacios  $\langle \mathbf{U}_1 \rangle$  y  $\langle \mathbf{U}_2 \rangle$  generan el espacio  $\mathbb{R}^N$  [Sch91].

## A.7 Proyecciones, Rotaciones y Pseudo-inversas

Sean las matrices:  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_p]$  y  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2 \cdots \mathbf{a}_{N-p}]$  de dimensión  $N \times p$  y  $N \times (N - p)$ , respectivamente. Denotamos los subespacios generados por las columnas de  $\mathbf{H}$  y de  $\mathbf{A}$  como  $\langle \mathbf{H} \rangle$  y  $\langle \mathbf{A} \rangle$  respectivamente. Asumiremos que tanto  $\mathbf{H}$  como  $\mathbf{A}$  son de rango completo y que,  $\mathbf{A}^T \mathbf{H} = \mathbf{0}$ . Por lo tanto:  $\begin{bmatrix} \mathbf{H} & \mathbf{A} \end{bmatrix} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_p, \mathbf{a}_1, \mathbf{a}_2 \cdots \mathbf{a}_{N-p}]$  es una matriz de rango completo, de  $N \times N$ , cuyas columnas generan el espacio:  $\mathbb{R}^N$ . Se dice que los subespacios  $\langle \mathbf{H} \rangle$  y  $\langle \mathbf{A} \rangle$  forman una descomposición ortogonal de  $\mathbb{R}^N$  [Sch91]. El subespacio  $\langle \mathbf{H} \rangle$  es llamado subespacio señal, y  $\langle \mathbf{A} \rangle$  subespacio ortogonal.

Las matrices reales  $\mathbf{H}$  y  $\mathbf{A}$  tienen la siguiente descomposición en valores singulares:

$$\mathbf{H} = \mathbf{U}_H \Sigma_H \mathbf{V}_H^T \quad \mathbf{A} = \mathbf{U}_A \Sigma_A \mathbf{V}_A^T \quad (\text{A.46})$$

donde  $\mathbf{U}_H$ ,  $\mathbf{U}_A$ ,  $\mathbf{V}_H$ , y  $\mathbf{V}_A$  son matrices ortogonales. Los subespacios  $\langle \mathbf{U}_H \rangle$  y  $\langle \mathbf{U}_A \rangle$  son idénticos a los subespacios  $\langle \mathbf{H} \rangle$  y  $\langle \mathbf{A} \rangle$  respectivamente [Sch91].

### A.7.1 Proyecciones

La descomposición del vector  $\mathbf{x} \in \mathbb{R}^n$  en dos componentes ortogonales, una que yace en  $\langle \mathbf{H} \rangle$  y la otra en  $\langle \mathbf{A} \rangle$ , está dada por [Sch91]:

$$\mathbf{x} = \mathbf{P}_H \mathbf{x} + \mathbf{P}_A \mathbf{x} \quad (\text{A.47})$$

donde  $\mathbf{P}_H$  y  $\mathbf{P}_A$  están dadas por:

$$\mathbf{P}_H = \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \quad \mathbf{P}_A = \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \mathbf{I} - \mathbf{P}_H \quad (\text{A.48})$$

De una manera equivalente se tiene:

$$\mathbf{P}_H = \mathbf{U}_H \mathbf{U}_H^T \quad \mathbf{P}_A = \mathbf{U}_A \mathbf{U}_A^T = \mathbf{I} - \mathbf{U}_H \mathbf{U}_H^T \quad (\text{A.49})$$

$$\mathbf{P}_H + \mathbf{P}_A = \mathbf{I}$$

Usando las ecuaciones anteriores  $\mathbf{x}$  se puede expresar como [Sch91]:

$$\mathbf{x} = \mathbf{U}_H \mathbf{U}_H^T \mathbf{x} + \mathbf{U}_A \mathbf{U}_A^T \mathbf{x} \quad (\text{A.50})$$

### A.7.2 Rotaciones

Un vector  $\mathbf{x} \in \mathbb{R}^N$  puede ser rotado en el subespacio  $\langle \mathbf{H} \rangle$  ó "rotado alrededor del subespacio  $\mathbf{A}$ ", mediante la transformación [Sch91]:

$$\mathbf{Q}_H \mathbf{x} \quad (\text{A.51})$$

donde  $\mathbf{Q}_H$  es una matriz de rotación que se construye a partir de  $\mathbf{P}_H$ ,  $\mathbf{P}_A$  y de una matriz ortogonal  $\mathbf{Q}$  [Sch91]:

$$\mathbf{Q}_H = \mathbf{U}_H \mathbf{Q} \mathbf{U}_H^T + \mathbf{P}_A \quad (\text{A.52})$$

### A.7.3 La Pseudo-inversa

La pseudo-inversa de una matriz real  $\mathbf{H}$  esta definida por:

$$\begin{aligned} \mathbf{H}^\# &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \\ &= \mathbf{V}_H \Sigma_H^{-1} \mathbf{U}_H^T \end{aligned} \quad (\text{A.53})$$

La pseudo-inversa de una matriz tiene las siguientes cuatro propiedades fundamentales:

$$\begin{aligned} \mathbf{H}^\# \mathbf{H} &= \mathbf{I} \\ \mathbf{H} \mathbf{H}^\# &= \mathbf{P}_H \\ \mathbf{H} \mathbf{H}^\# \mathbf{H} &= \mathbf{H} \\ \mathbf{H}^\# \mathbf{H} \mathbf{H}^\# &= \mathbf{H}^\# \end{aligned} \quad (\text{A.54})$$

La matriz pseudo inversa soluciona el problema de los mínimos cuadrados [Sch91]:

$$\begin{aligned} \min_{\theta} (\mathbf{y} - \mathbf{H}\theta)^T (\mathbf{y} - \mathbf{H}\theta) \\ \hat{\theta} = \mathbf{H}^\# \mathbf{y} \end{aligned} \quad (\text{A.55})$$

La matriz de proyección  $\mathbf{P}_H$  soluciona el siguiente problema [Sch91]:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{x} = \mathbf{H}\theta} (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) \\ \hat{\mathbf{x}} = \mathbf{P}_H \mathbf{y} \end{aligned} \quad (\text{A.56})$$

El error entre  $\mathbf{y}$  y  $\hat{\mathbf{x}}$  es ortogonal a  $\hat{\mathbf{x}}$ :

$$\begin{aligned} (\mathbf{y} - \hat{\mathbf{x}})^T \hat{\mathbf{x}} &= \mathbf{y}^T (\mathbf{I} - \mathbf{P}_H) \mathbf{P}_H \mathbf{y} \\ &= \mathbf{0} \end{aligned} \quad (\text{A.57})$$

### A.7.4 Representaciones Ortogonales

La DVS diagonaliza simultáneamente la proyección  $\mathbf{P}_H$ , la matriz Gramiana  $\mathbf{G}_H = \mathbf{H}^T \mathbf{H}$ , y la pseudo-inversa  $\mathbf{H}^\#$ :

$$\mathbf{P}_H = \mathbf{U} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^T$$

$$\mathbf{G}_H = \mathbf{V} \Sigma^2 \mathbf{V}^T \quad (\text{A.58})$$

$$\mathbf{H}^\# = \mathbf{V} \Sigma^{-1} \mathbf{U}^T; \quad \Sigma^{-1} = \begin{bmatrix} \Sigma_H^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} .$$

Por lo tanto, los estimadores  $\hat{\theta}$  y  $\hat{\mathbf{x}}$  pueden expresarse como [Sch91]:

$$\begin{aligned} \hat{\theta} &= \mathbf{V} \Sigma^{-1} \mathbf{U}^T \mathbf{y} \\ &= \sum_{i=1}^p \frac{1}{\lambda_i} \mathbf{v}_i \mathbf{u}_i^T \mathbf{y} \\ \hat{\mathbf{x}} &= \sum_{i=1}^p \mathbf{u}_i \mathbf{u}_i^T \mathbf{y} \end{aligned} \quad (\text{A.59})$$

## A.8 Fórmulas de Inversión de Matrices

Existen dos lemas fundamentales de inversión matricial que nos dan una representación para la inversa de una matriz particionada [Sch91].

**Lema 1 (Inversa de una matriz particionada).** Sea  $\mathbf{R}$  una matriz particionada:

$$\mathbf{R} = \left[ \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right] \quad (\text{A.60})$$

La inversa de  $\mathbf{R}$  esta dada por:

$$\mathbf{R}^{-1} = \left[ \begin{array}{c|c} \mathbf{E}^{-1} & \mathbf{F}\mathbf{H}^{-1} \\ \hline \mathbf{H}^{-1}\mathbf{G} & \mathbf{H}^{-1} \end{array} \right]$$
$$\mathbf{E} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$$
$$\mathbf{A}\mathbf{F} = -\mathbf{B} \quad (\text{A.61})$$
$$\mathbf{G}\mathbf{A} = -\mathbf{C}$$
$$\mathbf{H} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$$

Se asume que todas las inversas existen. Las matrices  $\mathbf{E}$  y  $\mathbf{H}$  se les conoce como el complemento de Schur de la matriz  $\mathbf{A}$ , y de la matriz  $\mathbf{D}$ , respectivamente [Sch91]. $\square$

**Lema 2 (Lema de Inversión Matricial)** Sea  $\mathbf{E}$  el complemento de Schur de la matriz  $\mathbf{A}$ :

$$\mathbf{E} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} \quad (\text{A.62})$$

La inversa de  $\mathbf{E}$  esta dada por [Sch91]:

$$\mathbf{E}^{-1} = \mathbf{A}^{-1} + \mathbf{F}\mathbf{H}^{-1}\mathbf{G}$$
$$\mathbf{A}\mathbf{F} = -\mathbf{B}$$
$$\mathbf{G}\mathbf{A} = -\mathbf{C} \quad (\text{A.63})$$
$$\mathbf{H} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$$

Los lemas 1 y 2 se combinan para dar lugar a la siguiente representación para la inversa de una matriz particionada. $\square$

**Teorema (Inversa de una Matriz Particionada)** La inversa de una matriz particionada:

$$\mathbf{R} = \left[ \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right] \quad (\text{A.64})$$

esta dada por [Sch91]:

$$\mathbf{R}^{-1} = \left[ \begin{array}{c|c} \mathbf{A}^{-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \left[ \begin{array}{c} \mathbf{F} \\ \mathbf{I} \end{array} \right] (\mathbf{H}^{-1}) \left[ \mathbf{G} \mid \mathbf{I} \right]$$

$$\mathbf{A}\mathbf{F} = -\mathbf{B} \tag{A.65}$$

$$\mathbf{G}\mathbf{A} = -\mathbf{C}$$

$$\mathbf{H} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \quad \square.$$

## Glosario

- **AFM**: "adaptive filter mode", la traducción propuesta al español de este anglicismo es: "modo de filtrado adaptable".
- **AR**: "autoregressive", la traducción propuesta al español de este anglicismo es: proceso autoregresivo.
- **ARMA**: "autoregressive moving average", la traducción propuesta de este anglicismo al español es: "proceso autoregresivo y de promedio móvil".
- **FTF**: "fast transversal filter", la traducción propuesta al español de este anglicismo es: "filtro rápido transversal".
- **HARF**: "hyperstable adaptive recursive filter", la traducción propuesta al español de este anglicismo es: filtro adaptivo hyperstable".
- **IF**: "independent filtering algorithm", la traducción propuesta al español de este anglicismo es: "algoritmo de filtrado independiente".
- **ISI**: "intersymbol interference", la traducción propuesta al español de este anglicismo es: "interferencia intersímbolos".
- **LMS**: "least mean square", la traducción propuesta al español de este anglicismo es: "mínimos cuadrados medios".
- **LSL**: "least squares lattice", la traducción propuesta al español de este anglicismo es: "mínimos cuadrados en lattice".
- **LS**: "least squares", la traducción propuesta al español de este anglicismo es: "mínimos cuadrados".
- **MA**: "moving average", la traducción propuesta al español de este anglicismo es: promedio móvil.
- **MSE**: "error cuadrático medio", la traducción propuesta al español de este anglicismo es: "error cuadrático medio".

- **NLMS**: "normalized least mean square", la traducción propuesta al español de este anglicismo es: "mínimos cuadrados medios normalizados".
- **PAM**: "Pulse Amplitude Modulation", la traducción propuesta al español de este anglicismo es: "pulse amplitude modulation"
- **QAM**: "quadrature amplitude modulation", la traducción propuesta al español de este anglicismo es: "modulación en amplitud de cuadratura".
- **RLS**: "recursive least squares", la traducción propuesta al español de este anglicismo es: mínimos cuadrados recursivos.
- **SARF**: "simple hyperstable adaptive recursive filter", la traducción propuesta al español de este anglicismo es: "filtro hyperstable adaptivo simplificado".
- **SIM**: "system identification mode", la traducción propuesta al español de este anglicismo es: "modo de identificación de sistemas".
- **SNR**: "signal to noise ratio", la traducción propuesta al español de este anglicismo es: "razón señal a ruido".
- **SPR**: "strictly positive real", la traducción propuesta al español de este anglicismo es: "estrictamente positivo y real".
- **SVD**: "singular value decomposition", la traducción propuesta al español de este anglicismo es: "descomposición en valores singulares".
- **ZF**: "zero forcing", la traducción propuesta al español de este anglicismo es: "algoritmo de forzado a cero".

## Bibliografía

- [AE71] K. J. Astrom and P. Eykhoff. System identification - a survey. *Automatica*, 7(2):123-162, 1971.
- [Alc86] Rogelio Alcántara. *Implantation D'Algorithmes Rapides sur des Processeurs de Traitement du Signal*. PhD thesis, Ecole Nationale Des Telecommunications, 1986.
- [And68] B. D. O. Anderson. A simplified viewpoint of hyperstability. *IEEE Trans. Automat. Contr.*, AC-13:292-294, Junio 1968.
- [Bel63] P. A. Bello. Characterization of randomly time invariant linear channels. *IEEE Trans. on Communications Systems*, 23:1043-1047, DECEMBER 1963.
- [Bin88] J. A. C. Bingham. *The Theory and Practice of Modern Design*. Wiley Interscience, 1988.
- [bSH94] Edited by Simon Haykin. *Blind Deconvolution*. Prentice Hall, Englewood, New Jersey, 1994.
- [CG85] C. F. N. Cowan and P. M. Grant. *Adaptive Filters*. Prentice Hall, Englewood Cliffs, NJ, 1985.
- [CK84] J.M. Cioffi and T. Kailath. Fast, recursive-least-squares transversal filters for adaptive filtering. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-32:304-337, 1984.
- [CRJ78] Jr. C. Richard Johnson. A convergence proof for a hypersatble adaptive recursive filter. *IEEE Trans. On Information Theory*, IT-25(6), Noviembre 1978.
- [CRJ84] Jr. C. Richard Johnson. Adaptive iir filtering: Current results and open issues. *IEEE Trans. On Information Theory*, IT-30(2):237-250, Marzo 1984.
- [CRJL77] Jr. C. R. Johnson and M. G. Larimore. Comments on and additions to 'an adaptive recursive lms filter'. In *Proc. IEEE*, volume 65, pages 1399-1402, Septiembre 1977.

- [CRJLTD81] Jr. C. R. Johnson, I. D. Landau, T. Taylor, and L. Dugard. On adaptive iir filters and parallel adaptive identifiers with adaptive error filtering. In *Proc. ICASSP.*, pages 538-541. Atlanta, GA., 1981.
- [CRJT80] Jr. C. Richard Johnson and T. Taylor. Failure of a parallel adaptive identifier with adaptive error filtering. *IEEE Trans. Automat. Contr.*, Ac-25(6), Diciembre 1980.
- [Fan85] H. Fan. *New Adaptive IIR Filtering Algorithms*. PhD thesis, Univ. Illinois, Urbana, IL, 1985.
- [FJ86] Hong Fan and W. Kenneth Jenkins. A new adaptive iir filter. *IEEE Trans. On Circuits and Systems*, CAS-33(10):939-946, Octubre 1986.
- [Gar86] Wiliam A. Gardner. The spectral correlation of cyclostationary time series. *Signal Processing (North-Holland)*, 11(1):13-36, Julio 1986.
- [GHW92] R.D. Gitlin, J.F. Hayes, and S.B. Weinstein. *Data Communication Principles*. Plenum Press, 1992.
- [GMK83] Carayannis G., D. G. Manolakis, and N. Kalouptsidis. A fast sequential algorithm for least squares filtering and prediction. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-31:1394-1402, 1983.
- [GS84] G. C. Goodwin and K. S. Sin. *Adaptive Filtering, Prediction and Control*. Prentice Hall, Englewood Cliffs, NJ, 1984.
- [HA69] L. Hitz and B. D. O. Anderson. Discrete positive real functions and their application to system stability. In *Proc. Inst. Elec. Eng.*, volume 116, pages 153-155, Enero 1969.
- [Hay91] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, N.J., 1991.
- [JN86] W. K. Jenkis and M. Nayeri. Adaptive filters realized with second order sections. In *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Proc.*, pages 2103-2106, Tokyo, Japon, Abril 1986.
- [JT79] C. R. Johnson Jr. and T. Taylor. Charf convergence studies. In *Proc. 13th Asilomar Conf. Circuits, Syst., and Computers*. Pacific Grove, CA., Noviembre 1979.
- [Lan76] I. D. Landau. Unbiased recursive identification using model reference adaptive techniques. *IEEE Trans. Automat. Contr.*, AC-21:194-202, Abril 1976.
- [Lan79] I. D. Landau. *Adaptive Control: The Model Reference Approach*. Marcel Dekker, New York, 1979.

- [Lar79] M. G. Larimore. Hyperstability and adaptive filtering. In *Proc. 13th Asilomar Conf. Circuits, Syst., and Computers*, pages 412–415. Pacific Grove, CA., Noviembre 1979.
- [LD94] Ye Li and Zhi Ding. Arma system identification based on second order cyclostationary. *IEEE Trans. on Signal Processing*, 42(12):3483–3494. Diciembre 1994.
- [Lju87] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- [LMP85] F. Ling, D. Manolakis, and J.G. Proakis. New forms of ls lattice algorithms and an analysis of their round off error characteristics. In *Proc. ICASSP*, pages 1739–1742. Tampa, Fla., 1985.
- [LS83] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, Mass., 1983.
- [LTCRJ80] Michael G. Larimore, Jhon R. Treichler, and Jr. C. Richard Johnson. Sarf: An algorithm for adapting iir digital filters. *IEEE Trans. On Acoustics, Speech, And Signal Processing*, ASSP-28(4):428–440. Agosto 1980.
- [MAG76] D. Markel and Jr A.H. Gray. *Linear Prediction of Speech*. Springer-Verlag, New York, 1976.
- [Men87] J. M. Mendel. *Lessons in Digital Estimation Theory*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- [ML78] M. Morf and D. T. Lee. Recursive least squares ladder forms for fast parameter tracking. In *Proc. Conf. Decision Control*, pages 1362–1367. San Diego, Calif., 1978.
- [MT77] F. Mosteller and J. W. Tukey. *Data Analysis and Regression*. Addison-Wesley, Reading, Mass., 1977.
- [Nay88] M. Nayeri. A weaker sufficient condition for the unimodality of error surfaces associated with exactly matching adaptive iir filters. In *Proc. 22nd Asilomar Conf. Signals, Systems, Computers.*, Pacific Grove, CA., Noviembre 1988.
- [PA78] D. Parikh and N. Ahmed. On an adaptive algorithm for iir filters. In *Proc. IEEE*, volume 66, pages 585–588, Mayo 1978.
- [PF89] B. Porat and B. Friedlander. Performance analysis of parameter estimation algorithms based on high order moments. *International Journal on Adaptive Control and Signal Processing*, 3:191–229, 1989.

- [PM96] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1996.
- [Pop73] V. M. Popov. *Hyperstability of Control Systems*. Springer-Verlag, Berlín, Alemania, 1973.
- [PRLN92] J.G. Proakis, C.M. Rader, F. Ling, and C.L. Nikias. *Advanced Digital Signal Processing*. Macmillan Publishing Company, New York, 1992.
- [Pro95] John G. Proakis. *Digital Communications*. McGraw Hill, New York, 1995.
- [PSA79] D. Parikh, S.C. Sinha, and N. Ahmed. On a modification of the sharp algorithm. In *Proc. 22nd Midwest Symp. Circuits, Syst.*, Philadelphia, PA., Junio 1979.
- [Quy82] Shahid Quayreshi. Adaptive equalizators. *IEEE Communications Magazine*, pages 9-16, Marzo 1982.
- [S75] T. Söderström. On the uniqueness of maximum likelihood identification. *Automatica*, 11:193-197, 1975.
- [Sch91] Louis L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley Publishing Company, 1991.
- [SEA76] S. D. Stearns, G. R. Elliot, and N. Ahmed. On adaptive recursive filtering. In *Proc. 10th Asilomar Conf. Circuits, Syst., and Computers*, pages 5-10, 1976.
- [Ses90] N. Seshadri. Joint data and channel estimation using fast blind trellis search techniques. In *Global Telecommunications Conference (GLOBECOM'86)*, pages 1659-1663, Fort Lauderdale, FL., 1990.
- [Shy89] John J. Shynk. Adaptive iir filtering. *IEEE ASSP Magazine*, Abril 1989.
- [SS82] T. Söderström and P. Stoica. Some properties of the output error method. *Automatica*, 18(1):93-99, Enero 1982.
- [Ste81] S. D. Stearns. Error surfaces of recursive adaptive filters. *IEEE Trans. On Acoustics, Speech and Signal Processing*, ASSP-29(3):763-766, Junio 1981.
- [Str88] Gilbert Strang. *Algebra Lineal y sus Aplicaciones*. Fondo Educativo Interamericano, 1988.

- [TFJ96] J.R. Treichler, I. Fijalkow, and C.R. Johnson, Jr. Fractionally spaced-equalizers. *IEEE Signal Processing Magazine*, pages 65-81, MAY 1996.
- [TLJ78] J. R. Treichler, M. G. Larimore, and C. R. Johnson Jr. Simple adaptive iir filtering. In *Proc. 1978 Int. Conf. Acoust. Speech and Signal Processing*, pages 118-122. Tulsa, OK., Abril 1978.
- [Wat91] David S. Watkins. *Fundamentals of Matrix Computations*. Jhon Wiley & Sons, 1991.
- [Whi75] S. A. White. An adaptive recursive filter. In *Proc. 9th Asilomar Conf. Circuits, Syst., and Computers*, pages 21-25, Pacific Grove, CA., Noviembre 1975.
- [WM77] B. Widrow and J. M. McCool. Comments on 'an adaptive recursive lms filter'. In *Proc. IEEE*, volume 65, pages 1402-1404. Septiembre 1977.
- [WS85] Bernard Widrow and Samuel Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, N.J., 1985.

# Índice Analítico

- algoritmo
  - AFM "adaptive filter mode" 101.
  - fast transversal filter (FTF) 28, 52
  - Gram-Schmidt 69.
  - HARF 90, 93
  - IF "independent filtering algorithm" 102.
  - least mean squares (LMS) 15, 25.
  - recursive least squares lattice algorithm (LSL) 28, 57
  - mínimos cuadrados (MC ó LS) 15, 28.
  - recursivo de los mínimos cuadrados 28, 33.
  - SARF 96.
  - algoritmo SIM "system identification mode" 101.
  - steepest descent 21.
  - zero forcing 15, 16.
- aliasing temporal 10.
- ancho de banda mínimo 11.
- banda de Nyquist 11.
- bases 155.
- complemento de Schur 161.
- criterio de la distorsión de pico 16.
- crosstalking 10.
- Delay spread 12.
- determinante Gramiano 150.
- Doppler spread 12.
- ecuaciones,
  - ecuación de error 83.
  - "least mean square equation error (LM-SEE)" 100.
  - normales 31.
  - de Wiener-Hopf 23.
- error,
  - error MSEOE 84.
  - error MSEE 88.
  - a posteriori de predicción hacia atrás 43.
  - a priori de predicción hacia adelante 41.
  - de estimación 22, 29.
  - de estimación a priori 36,
  - de estimación a posteriori 36.
  - de predicción a posteriori 38.
- espacios,
  - de Hilbert 150.
  - dimensión de un... 156.
  - Euclidianos 150.
  - generado 155.
  - lineales 155.
  - vectoriales 149.
  - vectorial lineal normado 150.
- factor
  - de Conversión 46.
  - de olvido 33.
  - QR 151.
- filtro,
  - estructuras de,
    - lattice 14.
    - tapped delay line 12, 14
  - filtro FIR 12.
  - filtro IIR 83, 105
  - "hyperstable adaptive recursive filter" (HARF) 81.
  - "simple hyperstable adaptive recursive filter (SARF)" 97.
- ganancia,
  - de Kalman 35.
- hyperestabilidad 81, 91
  - asintótica 91.
  - "hyperstable adaptive recursive filter" (HARF) 81.
- independencia lineal 151.
- igualación ciega 8.
- igualadores de canal,
  - decision feedback equalizer 14, 17
  - adaptivo 13.
  - de canal fraccionalmente espaciado 8, 13, 17.
  - lineal 14, 21

- retroalimentado (vea decision feedback equalizer),
- T-spaced equalizers 13,
- innovación 36,
- interferencia intersímbolos 10,
- ley de la nulidad de Sylvester 157,
- matrices,
  - de autocorrelación 24,
  - Gramiana 151,
  - Hermiteanas 157,
  - pseudo-inversa 162.
- método,
  - de la ecuación de error 82,
  - del error de salida 82, 84.
- Norma de Frobenius 160,
- parámetros,
  - forward 93,
  - backward 93,
- predictor,
  - forward 39,
- prewindowing 57,
- principio de ortogonalidad 30,
- procedimiento de Gram-Schmidt 152,
- producto,
  - interno 150,
  - exterior 157,
- propagación multitrayectoria 11,
- proceso autoregresivo y de promedio móvil
  - ARMA 81,
- proyecciones 161,
- tap 12,
  - espaciamientos del tap 12,
- respuesta deseada 21,
- rotaciones 161,
- "strict positive reality" (SPR) 81,
- subespacio 155,
  - directos 156,
  - ortogonal 161,
  - señal 161,
- superficies de error
  - error performance surface 23,
  - Superficie Error-Desempeño del Filtro Adaptivo 23,
- transformación,
  - de Householder 153,
  - de Givens 154,
  - nulo de una... 157,
  - rancho de una... 156,
  - valores singulares 157,
  - vector,
    - de Ganancia 35,
    - de coeficientes del filtro de predicción del error 39, 43,
  - wide sense stationary processes (WSS) 21,