

**SISTEMAS ADAPTABLES PARA  
FILTRADO Y DETECCIÓN**

**TESIS QUE PRESENTA LA**

**M. en C. Mariko Nakano Miyatake**

**PARA OBTENER DEL GRADO DE**

**DOCTOR EN CIENCIAS**

**DICIEMBRE 1998**

**UNIVERSIDAD AUTONOMA METROPOLITANA  
UNIDAD IZTAPALAPA**

**DIVISION DE CIENCIAS BASICAS E INGENIERIA**

# AGRADECIMIENTOS

Mi mas sincero agradecimiento:

Por su gran motivación, dirección y asesoría al

Dr. Héctor Pérez Meana

Por sus útiles discusiones y sugerencias durante la realización de esta investigación a

Dr. Kunihiro Fukushima de The Osaka University (Japón),

Dr. Haruhisa Takahashi de The University of Electro-Communications (Tokio, Japón)

Dr. Yoshirou Kumamoto de The University of Electro-Communications (Tokio, Japón)

Por su apoyo y facilidades proporcionadas a las instalaciones (Biblioteca, Laboratorios, Base de Datos, etc) de The University of Electro-Communications.

Dr. Masataka Ariyama , Rector de The University of Electro-Communications

Por su apoyo y estímulo durante la realización de esta investigación a

Dr. José Luis del Río

Prof. Luis Niño de Rivera y O.

Prof. Juan Carlos Sánchez García

Profa. Laura Ortiz Balbuena

Prof. Alejandro Martínez González

Prof. Gonzalo Duchén Sánchez

Prof. Rubén Vázquez Medina

# Contenido

<b>CAPITULO 1. Introducción</b>	1
1.1 Desarrollo del Proyecto de Investigación	4
1.2 Organización de la Tesis	5
<b>PARTE I ESTADO DEL ARTE</b>	
<b>CAPITULO 2. Sistemas Adaptables</b>	7
2.1 El Combinador Lineal Adaptable	8
2.2 Filtros Adaptables	8
2.2.1 Principales Estructuras de Filtrado Adaptable	9
2.2.1.1 Filtro Adaptable Transversal	9
2.2.1.2 Estructuras Ortogonalizadas para Filtrado Adaptable	10
2.2.1.3 Filtrado Adaptable en el Dominio de Frecuencia	11
2.2.1.4 Filtrado Adaptable en Subbandas	12
2.2.1.5 Filtros Adaptable Respuesta a Impulso Infinita	14
2.2.2 Aplicaciones de Filtrado Adaptable	15
2.2.2.1 Cancelación de Eco	15
2.2.2.2 Predicción Lineal	16
2.2.2.3 Igualador de Canal	16
2.3 Redes Neuronales	18
2.3.1 Neuronals	19
2.3.2 Principales Estructuras de Redes Neuronales	20
2.3.2.1 Redes Neuronales con Supervisor	21
2.3.2.2 Redes Neuronales sin Supervisor	24
2.3.3 Aplicación de Redes Neuronales	26
2.3.3.1 Reconocimiento de Patrones	26
2.3.3.2 Problema de Optimización	27
<b>CAPITULO 3. Algoritmo de Adaptación</b>	28
3.1 Introducción	29
3.2 Algoritmos de LMS	29
3.2.1 Complejidad Computacional y Característica de Convergencia	31
3.2.2 Propiedades del Gradiente Instantáneo	34
3.3 Algoritmo LMS Normalizado	35

3.3.1 Propiedad de Convergencia con Señales Estacionarias y No Estacionarias	38
3.4 Algoritmo de RLS (Recursive Least Square)	40
3.4.1 Propiedad de Convergencia con Señales Estacionarias y No Estacionarias	44
3.5 Algoritmo de Retropropagación	44

## PARTE II ALGORITMOS PROPUESTOS

<b>CAPITULO 4. Algoritmo LMS con Factores de Convergencia Variables en el Tiempo</b>	<b>48</b>
4.1 Introducción	49
4.2 Efecto del Factores de Convergencia en el Funcionamiento de los Algoritmos LMS	49
4.3 Constante de Tiempo	51
4.4 Algoritmo Propuesto	53
4.5 Algoritmos del Tipo LMS con Factores de Convergencia Variables con el Tiempo Propuestos por otros Autores.	59
4.5.1 Algoritmo VSLMS	59
4.5.2 Algoritmo VECLMS	60
4.5.3 Algoritmo CC_LMS	61
4.5.4 Algoritmo ACFLMS	63
4.5.5 Algoritmo NACFLMS	63
4.6 Simulaciones Computacionales	64
4.7 Conclusiones	66
<b>CAPITULO 5. Algoritmo para Filtrado Adaptable en el Dominio de Frecuencia con Corto Tiempo de Retardo</b>	<b>67</b>
5.1 Introducción	68
5.2 Algoritmo LMS en Bloque (FLMS)	70

5.2.1 Condición de Convergencia	72
5.2.2 Número de Multiplicaciones por Período de Muestreo	77
5.3 Filtro Adaptable con Corto Retardo de Bloque (Propuesto)	77
5.3.1 Estructura Propuesta	78
5.3.2 Algoritmo de Adaptación	79
5.3.3 Factor de Convergencia	80
5.3.4 Numero de Multiplicaciones por Período de Muestreo	83
5.4 Resultados Obtenidos por Simulación	84
5.5 Conclusiones	85
<b>CAPITULO 6. Algoritmos Adaptables para Filtrado y Predicción Usando una Red de Hopfield Modificada</b>	<b>87</b>
6.1 Introducción	88
6.2 Estructura Propuesta	89
6.3 Complejidad Computacional	93
6.4 Simulaciones Computacionales	94
6.5 Conclusiones	97
<b>CAPITULO 7. Red de Hopfield Modificada para Igualador de Canal de Comunicación</b>	<b>99</b>
7.1 Introducción	100
7.2 Modelo del Canal de Comunicación Móvil	102
7.3 La Estructura Propuesta	106
7.4 Simulaciones Computacional	108
7.5 Conclusiones	111

## PARTE III CONCLUSIONES

<b>CAPITULO 8. Conclusiones</b>	113
8.1 Introducción	114
8.2 Conclusiones y Línea de Investigación Futura	114
<b>CAPITULO 9. Bibliografía</b>	118
9.1 Referencias Capitulo 1	118
9.2 Referencias Capitulo 2	120
9.3 Referencias Capitulo 3	123
9.4 Referencias Capitulo 4	123
9.5 Referencias Capitulo 5	124
9.6 Referencias Capitulo 6	126
9.7 Referencias Capitulo 7	128
9.8 Publicaciones Realizadas	
9.8.1 Revistas Internacionales con Arbitraje Estricto	130
9.8.2 Capítulos en Libro	131
9.8.3 Artículos in Extenso en Memorias de Conferencias Internacionales	131
9.8.4 Artículos in Extenso en Memorias de Conferencias Nacionales	132

# CAPITULO 1

## Introducción

Los sistemas adaptables han sido tema de intensa investigación durante los últimos veinticinco años, tiempo en el cual han probado su enorme potencial en la solución de una amplia variedad de problemas prácticos en diversos campos de las ciencias e ingeniería [1.1], [1.2]. A lo largo de este tiempo, los sistemas adaptables se fueron incorporando paulatinamente a diversos sistemas prácticos, llegando a ser actualmente parte fundamental de los sistemas de cancelación de eco y ruido, eliminación de interferencias y compresión de datos empleados en los sistemas modernos de comunicación ya sea móviles o fijos, etc. [1.1], [1.2]. Los sistemas adaptables también han sido incorporados con éxito en equipos de exploración geológica, equipos médicos, sistemas de seguridad, equipos de control, etc. Su exitoso funcionamiento en estas y otras aplicaciones, así como el desarrollo experimentado por la tecnología digital durante los últimos años, han estimulado enormemente la investigación en este campo, dando como resultado la aparición de nuevos sistemas adaptables cada vez mas eficientes y sofisticados.

Los sistemas adaptables se pueden dividir básicamente en filtros adaptables los cuales son sistemas lineales, y las redes neuronales las cuales son esencialmente sistemas no lineales. Aunque cabe mencionar que existe una gran similitud entre ambos pudiendo ser, en muchos casos, los algoritmos desarrollados para un tipo de sistemas usados en el otro con modificaciones menores. Tanto los filtros adaptables como las redes neuronales hicieron su aparición aproximadamente en la misma época, sin embargo los filtros adaptables han alcanzado a la fecha un estado de desarrollo mayor que las redes neuronales, Sin embargo, debido a la intensa investigación que se lleva acabo actualmente, se espera que en breve estas alcancen un estado de madurez similar a la de los filtros adaptables.

Pese a la intensa investigación que se ha llevado a cabo en el campo de los filtros adaptables, aun existe un número importante de problemas por resolver mismos que serán lo que durante el desarrollo de la presente investigación. En ésta abordaremos algunos de los problemas que aun presentan los filtros adaptables, para lo cual se analizarán algunas de las limitaciones que aun se tienen en este campo y se propondrán soluciones a algunas de ellas que permitan mejorar el funcionamiento de los filtros adaptables. Así con el fin de presentar un panorama de los problemas aun existentes el filtrado adaptable, presentaremos una breve descripción de las partes que lo conforman, pudiéndose encontrar una descripción mas amplia en los capítulos posteriores.

Un filtro adaptable consiste de una etapa conocida como estructura, que es la parte del sistema que realiza el filtrado y de un algoritmo de adaptación, que es la encargada de llevar a cabo el entrenamiento del sistema. Ambas partes son igualmente importantes, estando además

estrechamente relacionada una con la otra, ya que difícilmente se puede desarrollar una estructura para filtrado adaptable sin considerar un algoritmo de adaptación asociado a esta y viceversa. Esto desde luego no implica que necesariamente se deban desarrollar ambos, ya que es posible desarrollar una nueva estructura y entrenarla con un algoritmo conocido, o desarrollar un algoritmo nuevo para una estructura previamente propuesta, siendo esto último bastante común.

Los algoritmos adaptables mas comunes se pueden dividir básicamente en algoritmos basados en la búsqueda del gradiente, de los cuales el algoritmo LMS es el mas representativo [1.1], [1.3]-[1.5] y algoritmos del tipo Newton-Rapson de los cuales el mas común es el algoritmo RLS [1.2]. Los algoritmos de basados en la búsqueda del gradiente son muy atractivos debido a su baja complejidad y robustez numérica, sin embargo su sensibilidad al ruido aditivo es inversamente proporcional a su velocidad de convergencia, ya que una menor sensibilidad al ruido aditivo requiere factores de convergencia pequeños y esto a su vez implica una velocidad de convergencia menor. La búsqueda de una solución que satisfaga en forma simultanea estos dos factores esencialmente opuestos, baja sensibilidad al ruido aditivo y altas velocidad de convergencia, ha estimulado grandemente la investigación en este campo, propiciado el desarrollo de nuevos algoritmos basados en la búsqueda del gradiente que, manteniendo una baja complejidad, reduzcan la sensibilidad al ruido aditivo sin degradar su velocidad de convergencia. Por lo que a las estructuras se refiere, la mas usada es la estructura transversal de respuesta a impulso finita la cual es estable con complejidad proporcional al orden del filtro. Sin embargo esta complejidad podría ser alta en ciertas aplicaciones, especialmente cuando la capacidad de computo es limitada. Por esta razón han sido propuestas a la fecha diversas estructuras en el dominio de la frecuencia [1.5]-[1.10], estructuras en subbandas [1.11]-[1.18], estructuras de respuesta a impulso infinita, IIR [1.19]-[1.25], etc. Estas estructuras reducen en la mayoría de los casos la complejidad de los filtros FIR. Sin embargo todas ellas presentan algunos otros problemas que deben ser resueltos. Por ejemplo las estructuras en el dominio de la frecuencia, las cuales presentan la complejidad computacional mas baja, introducen retardos adicionales en la salida del sistema las cuales son en la mayoría de los casos iguales al orden del filtro. Las estructuras en subbandas también introducen retardos en la salida del, aunque estos son menores que los introducidos por la estructuras en el dominio de frecuencia, así como distorsión en las señales debido al proceso de análisis – síntesis que se emplea, aunque en general son mas sus virtudes que sus defectos. Los filtros IIR presentan problemas de estabilidad, convergencia a mínimos locales [1.28]-[1.30]. Así aunque se han propuesto numerosos algoritmos para filtrado adaptable, muchos de ellos muy eficientes, aun restan problemas por resolver en este campo, los cuales deberán ser abordados.

Durante la realización de esta investigación se abordaron dos problemas de gran importancia en el campo del filtrado adaptable: Reducción de la sensibilidad al ruido aditivo en sistema adaptables sin incrementar la complejidad computacional de los mismos, y el incremento en la velocidad de convergencia del sistema. Con este fin, y dada la diversidad de aplicaciones de los sistemas adaptables, se desarrollaron tres algoritmos que permiten resolver los problemas mencionados anteriormente, cuando el sistema opera ya sea con altos niveles de ruido aditivo o en ambientes rápidamente cambiantes con el tiempo. El primero de ellos se desarrolló para aplicaciones que requieren sistemas adaptables con velocidad de convergencia moderadamente

alta, operando en ambientes cambiantes con el tiempo y con niveles de ruido aditivo también variables con el tiempo. Un ejemplo típico de estas aplicaciones es la cancelación de eco en sistemas de telecomunicación. El segundo algoritmo se desarrolló para aplicaciones que requieren sistemas de alto orden, con moderadamente altas velocidades de convergencia y capaces de operar en ambientes altamente ruidosos. Finalmente el tercer algoritmo se propone para aplicaciones requieren sistemas adaptables de bajo orden, capaces de operar con altos niveles de ruido aditivo y altas velocidades de convergencia.

## 1.1 Desarrollo del Proyecto de Investigación

Debido a su importancia y potenciales aplicaciones, se realizó la investigación doctoral en el campo de los sistemas adaptables con el fin de proponer soluciones a algunos de los problemas mencionados anteriormente. Inicialmente, con el fin de acotar el problema se llevó a cabo una extensa revisión bibliográfica sobre el tema para determinar el estado del arte en este campo e identificar las posibles líneas de investigación. De esta revisión bibliográfica se identificaron los problemas aun existentes, llevándose a cabo posteriormente un análisis detallado de los mismos con el fin de determinar la factibilidad de proponer soluciones a algunos de ellos. Este análisis mostró que era factible proponer nuevos algoritmos adaptables, basados en la búsqueda del gradiente con mucha menor sensibilidad al ruido aditivo, que los existentes a la fecha, tanto en el dominio del tiempo como en el de la frecuencia sin incrementar de manera considerable su complejidad ni reducir su velocidad de convergencia. Así se propuso un algoritmo adaptable con factor de convergencia variable con el tiempo, el cual permite reducir la degradación que experimentan los algoritmos basados en la búsqueda del gradiente, cuando operan en ambientes ruidosos. Este algoritmo se presenta con detalle en el capítulo 4. La propuesta fue evaluada extensivamente por medio de simulaciones y se comprobaron sus deseables características. Se analizó su comportamiento en ambientes estacionarios y no estacionarios.

De la revisión bibliográfica se determinó también la necesidad de un sistema adaptable de alto orden con velocidades de convergencia relativamente altas, que no introdujera retardos considerables en la salida del sistema. Se encontró que había propuestas, empleando la transformada de Fourier, con baja complejidad y reducido el retardo de bloque que permitían la realización en tiempo real de sistemas de alto orden, aunque estos presentaban en general baja velocidad de convergencia. Se encontró además que existen algoritmos de bloque empleando la transformada de Fourier con relativamente altas velocidades de convergencia y baja complejidad, los cuales también eran susceptibles de realizarse en tiempo real. Sin embargo el tiempo de retardo en estos es proporcional al orden del filtro. Al no encontrar evidencia de que ambos problemas hubieran sido resueltos de manera simultánea, y dada la necesidad de un algoritmo con esas características, se propuso un algoritmo para filtrado en bloque con corto tiempo de retardo el cual, además de reducir el retardo de bloque, incrementa considerablemente la velocidad de convergencia del algoritmo. Este algoritmo se presenta en el capítulo 5.

Finalmente, se observó la necesidad de tener sistemas con altas velocidades de convergencia que fueran capaces de operar en ambientes ruidosos, tales como los que se presentan en las comunicaciones móviles. Es conocido que los sistemas adaptables tipo RLS poseen estas deseables características. Sin embargo su uso se ve limitado a causa de su complejidad y problemas de estabilidad numérica. Así, con el fin de desarrollar un sistema que pudiera proporcionar altas velocidades de convergencia, siendo además capaz de operar en ambientes ruidosos, se desarrolló un sistema adaptable tipo RLS, (Recursive Least Square), basado en redes neuronales, el cual pudiera ser realizado tanto en forma analógica como en forma digital. Esto permitiría resolverse problemas relativos a la igualación de canales y codificación en sistemas de comunicación móvil, donde el sistema debe presentar una alta velocidad de convergencia que permita igualar canales que varían rápidamente con el tiempo, operando con

altos niveles de ruido aditivo. Las evaluaciones realizadas muestran que estos sistemas podrían ser sumamente atractivos ya que, además de que su funcionamiento es altamente satisfactorio, su estructura y el hecho de no requerir inversión de matrices podría facilitar su realización tanto en software como en hardware sin problemas de estabilidad numérica. Este sistema se propone en el capítulo 6 como identificador y predictor y se evalúa como igualador de canal en el capítulo 7.

## 1.2 Organización de la Tesis

La tesis se encuentra, básicamente, dividida en tres partes. La primera parte, que consiste de los capítulos 2 y 3, presenta una revisión del estado del arte de los sistemas adaptables, sus principales estructuras, algoritmos de aprendizaje etc. La segunda parte, compuesta por los capítulos 4 al 7, constituye la parte principal de la tesis y es donde se presentan las aportaciones al campo del conocimiento, producto de la presente investigación doctoral. Finalmente la tercera parte, compuestas por los capítulos 8 y 9, presenta las conclusiones de la presente investigación y posibles líneas futuras de investigación que se pueden desprender del siguiente trabajo, así como la bibliografía empleada durante la presente investigación. Seguidamente se presenta una breve descripción del contenido de cada uno de los nueve capítulos de que consta la presente tesis doctoral.

El capítulo 2 presenta una revisión de las estructuras más usadas en filtrado adaptable resaltando sus principales características, ventajas y desventajas de cada una de ellas. Se mencionan además algunas de las aplicaciones más exitosas de estos sistemas. Se presenta también en el capítulo 2 una revisión de las redes neuronales más usadas junto con algunas de sus principales aplicaciones. Esta revisión se presenta en virtud de que redes neuronales fueron empleadas en los capítulos 6 y 7, así como debido a la gran semejanza existente entre los filtros adaptables y las redes neuronales.

El capítulo 3 presenta una revisión detallada de los dos algoritmos más usados en filtrado adaptable. El algoritmo LMS "Least Mean Square Algorithm" y el algoritmo RLS, "Recursive Least Square Algorithm". Junto con el algoritmo más empleado para el entrenamiento de redes neuronales con supervisor, como lo es el algoritmo de retropropagación o "Backpropagation algorithm".

El capítulo 4 propone un algoritmo adaptable de paso variable basado en la búsqueda del gradiente, para aplicaciones en las cuales se presentan fluctuaciones en la potencia del ruido aditivo, y se requiere una velocidad de convergencia moderadamente alta, o para aplicaciones en las cuales la potencia del ruido aditivo es relativamente grande. Un ejemplo típico de estas aplicaciones es la cancelación de eco en sistemas de comunicación. Se presenta la evaluación del algoritmo propuesto usando simulación y se compara su funcionamiento con el de los otros algoritmos similares de paso variable propuestos recientemente. Los resultados muestran que el algoritmo propuesto supera a los demás cuando la potencia del ruido aditivo cambia con el tiempo.

El capítulo 5 presenta una estructura para filtrado adaptables en bloque en el dominio de la

frecuencia, para aplicaciones que requieren filtros adaptable de alto orden operando en ambientes ruidosos, siendo la capacidad de computo relativamente baja. Una solución a este problema es el empleo de filtrado en bloque en el dominio de la frecuencia, aunque estos introducen retardos iguales al orden del filtro, lo cual en ocasiones degrada el funcionamiento del sistema. El algoritmo propuesto en este capítulo, reduce el retardo de bloque introducido por el filtrado en bloque convencional, proporcionando además una velocidad de convergencia mayor que la de otros sistemas similares propuestos hasta la fecha. El sistema propuesto fue evaluado extensivamente por medio de simulación.

El capítulo 6 propone una estructura para filtrado adaptable ya sea analógico o digital, para aplicaciones en las cuales se requieren sistemas de bajo orden con altas velocidades de convergencia, y baja sensibilidad al ruido aditivo. En el algoritmo propuesto la adaptación de los coeficientes se lleva a cabo usando una red neuronal de Hopfield modificada. El sistema se evalúa asumiendo una realización analógica y se compara con el funcionamiento obtenido por una estructura transversal adaptada usando un algoritmo tipo RLS. Se comparan ambos sistemas operando como identificadores de sistemas en ambientes estacionarios y no estacionarios. Ambos sistemas se evalúan también operando como predictores lineales. Los resultados muestran que la estructura propuesta se comporta en general mejor que la convencional cuando ambas operan en condiciones similares.

El capítulo 7 propone una estructura para igualación de canales rápidamente variantes con el tiempo, tal como los que se encuentran en la telefonía celular. La estructura propuesta, emplea un algoritmo de adaptación similar al propuesto en el capítulo 6, el cual está basado en una red neuronal de Hopfield. El sistema propuesto, al igual que los anteriores fue evaluado por simulación y su funcionamiento comparado con el de otros sistemas propuestos hasta la fecha.

El capítulo 8 presenta las conclusiones de la presente investigación doctoral e indica algunas líneas de investigación que podrían derivarse a partir de la presente tesis.

Finalmente el capítulo 9 presenta la bibliografía usada durante la realización de la presente investigación doctoral y la lista de publicaciones, tanto en revistas, capítulos de libros y memorias de congresos que se generaron durante su realización.

# CAPITULO 2

## Sistemas Adaptables

Los sistemas adaptables se pueden dividir en dos grupos principales: a) Los filtros adaptables los cuales son sistemas lineales. b) Las redes neuronales artificiales que son sistemas esencialmente no lineales. Ambos han sido objeto de intensa investigación durante los últimos años, debido a su gran potencial en la solución de una enorme variedad de problemas prácticos en diversas ramas de la ciencia e ingeniería [2.1], [2.4], [2.5], [2.22], [2.35]-[2.40]. En general, aunque estos dos campos se han desarrollado en forma independiente, tienen un gran número de aspectos comunes entre si, los cuales permiten que un número significativo de resultados encontrados en un campo pueda ser fácilmente usados en el otro.

El primer elemento común que aparece tanto en los filtros adaptables como en las redes neuronales artificiales es el Combinador Lineal Adaptable (ALC) [2.35]. Este, con sus entradas conectadas a una línea de retardo se convierte en uno de los elementos primordiales de un filtro adaptable [2.35], mientras que con sus entradas conectadas a una línea de retardo y su salida a un elemento no lineal, el ALC se convierte en un elemento básico, o neurona de una red neuronal de tiempo retardado o "Time Delay Neural Networks" (TDNN) [2.37] o en un igualador de canales de comunicación de datos. Finalmente, con su salida conectada a un elemento no lineal, y sus entradas en paralelo, como se muestra en la figura 2.1, el ALC se convierte en un elemento básico, o neurona, de una red neuronal de tipo perceptrón así como de algunas otras arquitecturas similares [2.35]-[2.40].

Otro elemento en común, del cual se hablará posteriormente, es el algoritmo de adaptación LMS (Least Mean Square) el cual se usa con mucha frecuencia para entrenar, o adaptar, tanto los filtros adaptables como las redes neuronales [2.1]-[2.22], [2.32], [2.33].

## 2.1 El Combinador Lineal Adaptable

El combinador lineal adaptable es el elemento básico tanto de los filtros adaptables como de las redes neuronales. La salida de este sistema (Fig. 2.1) esta dada por el producto punto del vector de entrada y del vector de coeficientes. Esta es entonces sustraída de la señal de referencia para producir la señal de error, la cual es entonces usada para optimizar el vector de coeficientes del sistema. Esta optimización generalmente se lleva a cabo por medio de un algoritmo de adaptación el cual minimiza, en la mayoría de los casos, el valor cuadrático medio de la diferencia entre la señal de referencia y la señal de salida del sistema.

Existe una amplia variedad de algoritmos de adaptación que pueden ser empleados para ajustar los coeficientes del sistema, siendo el mas usado el algoritmo LMS [2.1]-[2.5], o alguna variación del mismo [2.32], [2.33].

Seguidamente procederemos a revisar brevemente los conceptos básicos inherentes al filtrado adaptable y las estructuras principales empleadas para este fin.

## 2.2. Filtros Adaptables

Los filtros adaptables han gozado de un gran éxito en diversas aplicaciones comerciales tales como: Igualación de canales de comunicación [2.4], [2.34] en modems de alta velocidad para reducir o evitar la interferencia entre símbolos. Algunas de estas estructuras, como se mostrará posteriormente, se pueden ver también como redes neuronales de tiempo retardado (TDNN). Por otro lado los sistemas de comunicación de larga distancia, los sistemas de teleconferencia, así como los sistemas telefónicos a manos libres se encuentran equipados, en la mayoría de los casos, con canceladores de eco [2.3], [2.6], [2.13]-[2.21], [2.27]-[2.33], los cuales permiten una comunicación de alta calidad. Otras aplicaciones en las cuales los filtros adaptables han tenido gran impacto son: Cancelación de ruido, electrocardiografía fetal, predicción lineal la cual es ampliamente usada en los sistemas modernos de compresión de voz y estimación espectral, estimación de tiempos de retardo, etc. [2.4], [2.5].

Los filtros adaptables mas usados son aquellos de respuesta a impulso finita (FIR) [2.1]-[2.5] los cuales consisten de un combinador lineal adaptable (ALC) conectado a una línea de retardo, cuya salida es una combinación lineal de las muestras presente y pasadas de la secuencia de entrada. Así variando los pesos, o coeficientes, del filtro, la respuesta a impulso de este es directamente controlable. Ya que la respuesta en frecuencia del sistema es la transformada de Fourier de la respuesta a impulso, controlando la respuesta a impulso del sistema se puede controlar la respuesta en frecuencia del mismo. Así los pesos son ajustados, usualmente, en forma tal que el valor cuadrático medio de la diferencia entre la señal de referencia y la señal de salida (figura 2.2) sea minimizado [2.5]. Durante las dos últimas décadas, se han propuesto

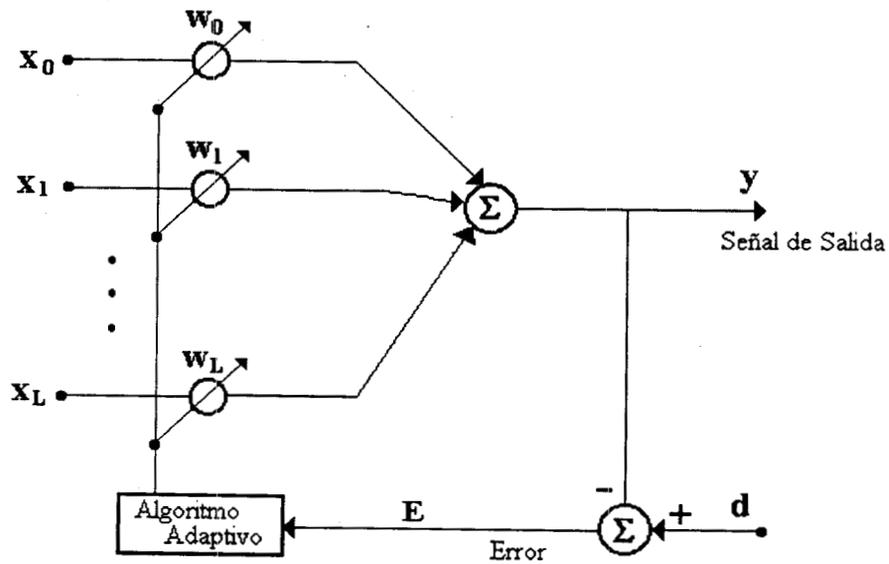


Figura 2.1. El Combinador Lineal Adaptable (ALC)

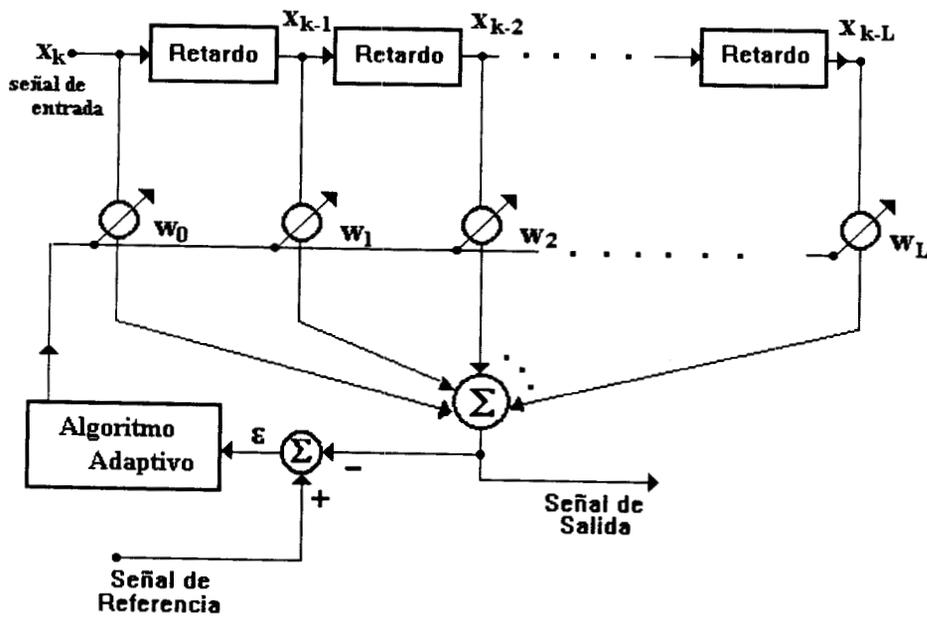


Figura 2.2. Filtro Transversal Adaptable.

una gran cantidad de algoritmos de adaptación, sin embargo de todos ellos el mas ampliamente usado sigue siendo el algoritmo LMS (Least Mean Square), o alguna variación de este, los cuales se basan en la búsqueda del mínimo en la dirección opuesta al gradiente de la superficie de error [2.1]-[2.21].

Seguidamente presentaremos una breve revisión de las estructuras para filtrado adaptable mas empleadas, tales como estructuras transversales de respuesta a impulso finita, FIR, estructuras de respuesta a impulso infinita, IIR, estructuras en el dominio de frecuencia, etc.

### **2.2.1. Principales Estructuras de Filtrado Adaptable.**

Esta sección presenta una revisión de las principales estructuras de filtrado adaptable, empleadas en la solución de diversos problemas prácticos.

#### **2.2.1.1. Filtro Adaptable Transversal.**

El filtro adaptable mas empleado en diversas aplicaciones es el filtro transversal FIR, el cual es entrenado usando ya sea algoritmos basados en la búsqueda del gradiente tal como el algoritmo LMS o alguna variación de este [2.2]-[2.4], [2.27]-[2.29], [2.32], [2.33], o en menores ocasiones el algoritmo recursivo de mínimos cuadrados (RLS) [2.1], [2.4], [2.28], [2.29]. Una de las principales razones de esto es el hecho de que esta estructura es incondicionalmente estable y converge en todos los casos al mínimo global de la superficie de error, esto es converge invariablemente a la solución óptima. En esta estructura, como se muestra en la figura 2.3, la salida del filtro es una combinación lineal de las muestras pasadas de la señal de entrada. Seguidamente señal de salida del filtro se compara con la señal de referencia,  $d(n)$  para formar la señal de error,  $e(n)$ , la cual es entonces usada para adaptar los coeficientes del filtro. No existe una forma general que nos permita seleccionar  $d(n)$ , ya que la señal de referencia depende directamente de la aplicación específica, por lo cual se proporcionarán varios ejemplos que ilustrarán como seleccionar  $d(n)$  para diversas aplicaciones de interés práctico. El algoritmo LMS funciona bastante bien cuando la densidad de potencia espectral de las señales de entrada es relativamente plana y la potencia del ruido aditivo es relativamente baja. Sin embargo, la densidad de potencia espectral de la mayoría de las señales de interés no es plana, además de que en muchas ocasiones las estadísticas de las señales involucradas cambian con el tiempo e invariablemente, se tiene la presencia del ruido aditivo. Esto dificulta la aplicación de filtros adaptables transversales con algoritmos LMS con factores de convergencia constantes, en aplicaciones prácticas que requieran algoritmos de adaptación con relativamente altas velocidades de convergencia y baja sensibilidad al ruido aditivo.

Varias estructuras alternas han sido propuestas las cuales intentan resolver algunas de las limitaciones que presentan los filtros adaptables transversales con algoritmos basados en la

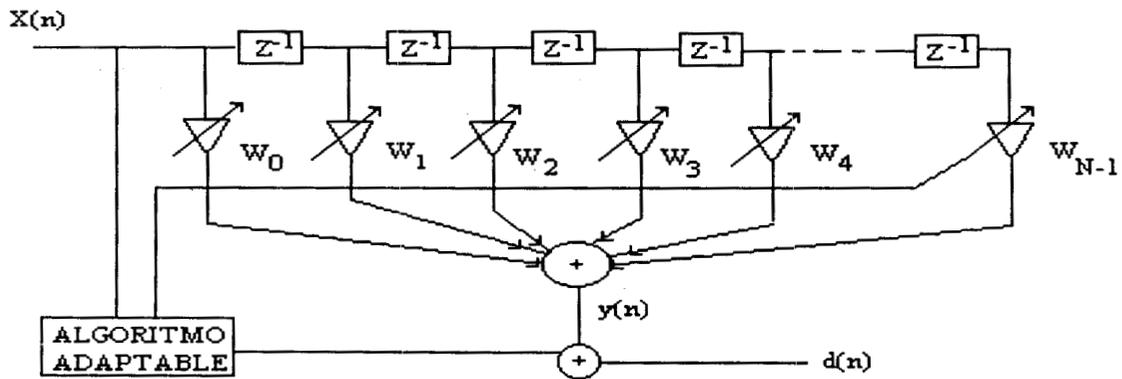


Figura 2.3. Filtro adaptable transversal de respuesta a impulso finita

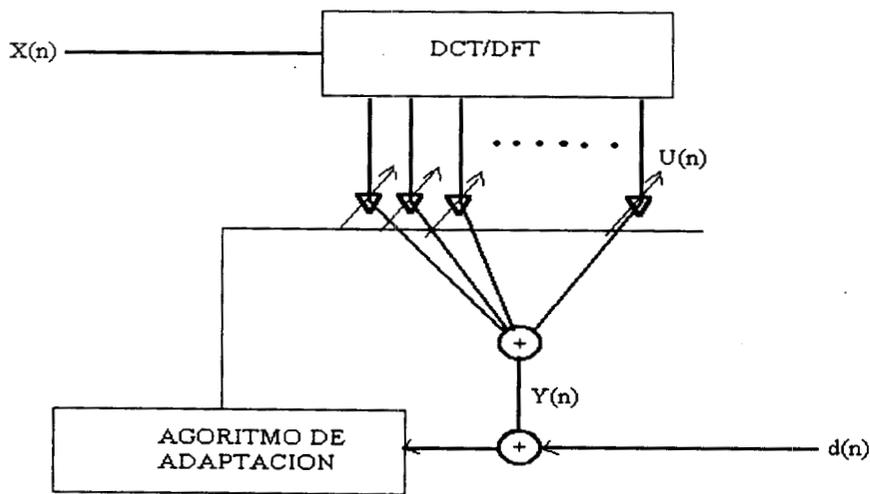


Figura 2.4. Filtro adaptable ortogonalizado de respuesta a impulso finita

búsqueda del gradiente usando factores de convergencia constantes. Entre ellas tenemos los filtros adaptables transversales con algoritmos de adaptación del tipo RLS [2.1], [2.4], [2.30], [2.31].

A diferencia del algoritmo LMS el cual controla el descenso en la dirección inversa del gradiente de la superficie de error usando un factor de convergencia escalar, el algoritmo RLS (Recursive Least Square) rota el vector gradiente de manera que este, con signo negativo, apunte al mínimo de la superficie de error con lo cual se obtienen altas velocidades de convergencia. Este algoritmo tiene una convergencia inicial extremadamente rápida, sin embargo el número de operaciones necesarias para filtrado y adaptación de los coeficientes del filtro es de orden  $N^2$ , donde  $N$  es el orden del filtro. Esta complejidad computacional que pudiera ser aceptable en algunas aplicaciones de tiempo real, pudiera a su vez ser muy grande para otras aplicaciones tales como cancelación de eco en sistemas telefónicos donde filtros con aproximadamente 100 coeficientes o más son requeridos, y más aun para la cancelación de ecos acústicos en sistemas de teleconferencia en donde serían necesarios filtros transversales con varios miles de coeficientes para atenuar el eco a niveles aceptables.

Explotando la propiedad de invarianza al corrimiento de los filtros transversales, el algoritmo RLS puede ser modificado de manera que su complejidad computacional puede ser reducida a orden  $N$ ,  $O(N)$  [2.1], [2.4], [2.5], [2.29]. Esto ha hecho posible la aparición de algoritmos del tipo RLS cuya complejidad es del mismo orden que la del algoritmo LMS. Figuras típicas son del orden de  $10N$  multiplicaciones por periodo de muestreo. Sin embargo esta complejidad computacional aun es considerablemente mayor que la del algoritmo LMS, la cual es  $2N$ , especialmente para filtros de orden alto. Aunado a lo anterior, el hecho de que la mayoría de estos algoritmos son numéricamente inestables ha limitado grandemente su uso en ciertas aplicaciones.

### **2.2.1.2. Estructuras Ortogonalizadas para Filtrado Adaptable**

Con el fin de obtener una mayor velocidad de convergencia con un costo computacional razonable, para aplicaciones en tiempo real, han sido propuestas las llamadas estructuras orthogonalizadas [2.8], [2.9], [2.25]-[2.27]. Estas estructuras resultan del análisis de la convergencia del algoritmo LMS, en la cual se hace evidente la relación existente entre la velocidad de convergencia del algoritmo LMS y la dispersión de los valores propios de la matriz de autocorrelación de la señal de entrada. Un algoritmo orthogonalizado involucra la construcción de un operador lineal el cual mapéa el vector de entrada  $\mathbf{X}(n)$  a un vector  $\mathbf{U}(n)$  cuyos elementos son ortogonales entre si. De esta forma, si el vector  $\mathbf{U}(n)$  es usado como vector de entrada de un filtro transversal adaptado con el algoritmo LMS, es relativamente sencillo predecir, a partir del análisis de convergencia del algoritmo LMS, que la estructura orthogonalizada (es decir operador lineal + normalización de los valores propios + algoritmo LMS) convergirá, en el sentido de mínimos cuadrados, a la misma velocidad que el algoritmo

LMS cuando la entrada de este es una secuencia de ruido blanco. Sin embargo solamente en un número limitado de aplicaciones se conoce de antemano la matriz de autocorrelación de la señal de entrada, o equivalentemente el operador que decorrelaciona entre sí las muestras de la señal de entrada. Así por lo general se emplea ya sea una estructura lattice o un operador fijo, tal como la transformada discreta coseno (DCT) o la transformada discreta de Fourier (DFT), para aproximadamente decorrelacionar la señal de entrada. Finalmente se aplica el algoritmo LMS asumiendo que la correlación entre las muestras de la señal de entrada ha sido totalmente eliminada. La estructura se muestra en la figura 2.4. Los algoritmos ortogonalizados proporcionan, en la mayoría de los casos de interés práctico, velocidades de convergencia cercanas a la que proporcionaría el algoritmo LMS operando con ruido blanco como señales de entrada.

Los algoritmos ortogonalizados presentan una complejidad computacional menor que la del algoritmo RLS, aunque velocidad de convergencia también es menor. Típicamente los algoritmos ortogonalizados requieren aproximadamente  $5N$  multiplicaciones por período de muestreo, siendo  $N$  el orden del filtro.

### **2.2.1.3. Filtrado Adaptable en el Dominio de Frecuencia**

En paralelo con los esfuerzos realizados para acelerar la velocidad de convergencia de los filtros transversales con algoritmos basados en la búsqueda del gradiente, muchos esfuerzos también han sido realizados para reducir la complejidad computacional de los filtros adaptables transversales, especialmente cuando filtros adaptables de alto orden son requeridos. Con esta finalidad se ha tomado ventaja de la transformada rápida de Fourier para calcular las operaciones de convolución y correlación requeridas, tanto para la operación de filtrado como para la adaptación de los coeficientes del filtro. El primer sistema de este tipo propuesto es algoritmo FLMS (Fast LMS), [2.10] el cual reduce la complejidad computacional de  $O(N)$  a  $O(\log_2(N))$ , donde  $N$  es el orden del filtro. En este sistema, con el fin de reducir la complejidad computacional mediante el uso de la transformada de Fourier, los coeficientes se mantienen constantes durante  $N$  periodos de muestreo. Esto permite el uso del método "Overlap-Save" con 50% de traslape tanto en operaciones de filtrado como de adaptación, reduciendo considerablemente la complejidad computacional del algoritmo. Aquí la complejidad computacional es proporcional al logaritmo natural de  $N$ , ( $O(\ln(N))$ ), en lugar de ser proporcional a  $N$ , ( $O(N)$ ), como en el algoritmo LMS, donde  $N$  es el orden del filtro. Sin embargo su velocidad de convergencia es en general menor, debido a que los coeficientes se adaptan únicamente al final de cada bloque de  $N$  periodos de muestreo usando un gradiente promediado, mientras que su condición de estabilidad permanece igual a la del algoritmo LMS. Otra limitación que presenta este tipo de algoritmos es el retardo de bloque introducido el cual es, en la mayoría de los casos, igual al orden del filtro.

Con el fin de resolver el primer problema, esto es incrementar la velocidad de convergencia, han sido propuestos algoritmos de bloque ortogonalizados con complejidad computacional

$O(\log_2(N))$  [2.12], los cuales proporcionan velocidades de convergencia, en muchos casos de interés práctico, iguales a las que proporciona el algoritmo FLMS cuando su señal de entrada es ruido blanco. Otra opción para incrementar la velocidad de convergencia del algoritmo FLMS la proporciona el "Jumping algorithm", JALG [2.6], [2.7]. En este, a partir de la prueba de convergencia del algoritmo LMS se deriva un conjunto de factores de convergencia óptimos, los cuales son obtenidos posteriormente, en forma aproximada, usando una transformación ortogonal tal como la transformada discreta coseno o la transformada discreta de Fourier. Este algoritmo proporciona velocidades de convergencia mucho mayores que aquellas proporcionada por el algoritmo FLMS, teniendo ambos una complejidad computacional similar. Tanto los algoritmos de bloque ortogonalizados, descritos anteriormente, como el "Jumping algorithm", JALG presentan velocidad de convergencia mayores que el del algoritmo FLMS, sin embargo éstos también introducen retardos de bloque proporcionales al orden del filtro.

Con la finalidad de reducir el retardo de bloque introducido por los algoritmos anteriormente mencionados, han aparecido en la literatura algoritmos tales como el FBAF [2.11]. En estos, las transformadas de Fourier de  $2N$  puntos, empleadas para realizar las operaciones de filtrado y adaptación, son reemplazadas, cada una, por la suma de  $M$  transformadas de Fourier de  $L$  puntos cada una como se muestra en la figura 2.5. Este hecho permite reducir el retardo de bloque e incrementar la velocidad de convergencia del algoritmo FLMS, aunque se incrementa un tanto su complejidad computacional.

#### **2.2.1.4. Filtrado Adaptable en Subbandas**

Otro tipo de estructuras para filtrado adaptable recientemente propuestas son aquellas que realizan el filtrado en subbandas [2.13]-[2.19], la cual se muestra en la figura 2.6. En este tipo de estructuras, tanto las señales de referencia como de entrada son divididas en  $N$  subbandas por medio de bancos de filtros pasa banda los cuales tienen un cierto grado de traslape con los filtros adyacentes. Subsecuentemente las señales de salida de cada filtro son submuestreadas por un factor  $M$ , el cual es menor aunque muy cercano al número de subbandas. Subsecuentemente en cada subbanda es insertado un filtro transversal adaptable operando a  $f_s/M$  Hz cuyos coeficientes son adaptados para minimizar el valor cuadrático medio del error de salida en cada subbanda. Finalmente los errores de salida son alimentados a otro banco de filtros pasabanda los cuales sintetizan la señal de salida del sistema. Los sistemas en subbandas han recibido mucha atención en los últimos años debido principalmente a que:

1. La frecuencia de muestreo es reducida por un factor  $M$ , siendo este cercano al número de subbandas, lo que permite que los procesos de filtrado y adaptación se lleven a cabo a una frecuencia de  $f_s/M$ , donde  $f_s$  es la frecuencia de muestreo. Este hecho reduce la complejidad computacional del sistema por un factor  $M$ .



2. La velocidad de convergencia puede ser incrementada de una manera importante ajustando los factores de convergencia en cada subbanda, de acuerdo a la energía de la señal de entrada en cada subbanda.
3. La correlación existente entre muestras sucesivas de las señales submuestreadas se debilita conforme se incrementan los factores de submuestreo. Así para factores de submuestreo relativamente grandes, la velocidad de convergencia llega a ser independiente de las estadísticas de las señales de entrada. Resultados experimentales muestran que para factores de submuestreo mayores o iguales a 16 la velocidad de convergencia del sistema con ruido blanco o señales de voz, como señales de entrada, es prácticamente la misma [2.18], [2.19].
4. La posibilidad de usar bandas específicas para obtener un mejor funcionamiento de dispositivos auxiliares, tales como los detectores de "double-talk", en cancelación de eco [2.19], [2.30].

Además de la reducción en la complejidad computacional y al incremento en la velocidad de convergencia, evitar la distorsión en la señal de salida y reducir el retardo introducido por el sistema son aspectos muy importantes que se deben considerar. Sin embargo cuando factores de submuestreo relativamente grandes son empleados, es difícil de reducir la potencia del error de salida en forma considerable, al mismo tiempo que se reduce el tiempo de retardo introducido por el sistema, debido a que una atenuación considerable de la potencia del error de salida requiere filtros pasa banda de alto orden. Por otro lado, una considerable reducción de la potencia del error de salida, puede ser obtenida relajando la condición de perfecta reconstrucción de los filtros QMF en los filtros de análisis y síntesis sin incrementar el orden de éstos. Sin embargo esto implica distorsionar la señal de salida del sistema.

Para reducir el tiempo de retardo del sistema en ciertas aplicaciones tales como cancelación de eco, sin distorsionar la señal de salida del mismo, se propuso un sistema para filtrado en subbandas en el cual se calculan dos diferentes error de salida. [2.18]-[2.21]. El primero para ser usado cuando el nivel del ruido aditivo es bajo, es calculado sustrayendo la señal sintetizada a partir de las señales de salida submuestreadas en cada subbanda de una replica de la señal de referencia, sintetizada también a partir de un banco de filtros para banda. La segunda por su parte, la cual será usada cuando el nivel de la señal útil (o ruido aditivo en ciertos casos) es alto, se obtiene sustrayendo la señal sintetizada a partir de las señales submuestreadas en cada subbanda, de una versión retrasada de la señal de referencia. Así la primera señal no es sensible a la distorsión introducida por los filtros pasa banda cuando la señal útil esta ausente (voz del interlocutor cercano en cancelación de eco). Por otro lado cuando la segunda señal de salida es empleada, el nivel de la señal de error crece, sin embargo en este caso la señal útil no se distorsiona. Este es el primer intento que se realiza para reducir el retardo introducido en los sistemas de filtrado adaptable en subbandas [2.18]-[2.21].

### 2.2.1.5 Filtros Adaptables de Respuesta a Impulso Infinita

Estructuras para filtrado adaptable de respuesta a impulso infinita basadas ya sea en el método de la ecuación de error o en el método del error de salida, representan otra alternativa para los filtros transversales. Entre estos, las estructuras basadas en el método de la ecuación de error (Fig. 2.7) [2.5], [2.22], [2.24] son los más usados debido a que su superficie de error es unimodal y casi cualquier algoritmo de adaptación para filtros transversales podría ser empleado. Sin embargo en presencia de altos niveles de ruido aditivo sus coeficientes llegan a ser sesgados. Esto limita su uso en algunas aplicaciones prácticas tales como cancelación de eco y ruido.

Otras alternativas son los sistemas IIR basados en el método del error de salida. El funcionamiento de estos sistemas (Fig. 2.8) es menos sensible a la presencia de ruido aditivo, sin embargo su superficie de error cuadrático medio es multimodal y su velocidad de convergencia es en general menor. Esto junto con el problema de mantener el sistema estable durante la adaptación hace difícil su empleo en aplicaciones que requieren filtros de orden relativamente alto.

Recientemente han sido propuestas estructuras de respuesta a impulso infinita usando realizaciones en paralelo [2.22]-[2.24] las cuales tienen pruebas de estabilidad trivial y una complejidad computacional de orden  $N$ , donde  $N$  es el número de coeficientes presentes en la función de transferencia del sistema. Sin embargo, aunque estos sistemas proporcionan altas velocidades de convergencia guardando una complejidad computacional relativamente baja, el sistema podría converger a cualquiera de los mínimos locales de la superficie de error.

Algunos intentos han sido hechos con el fin de solucionar algunos de los problemas mencionados anteriormente. Una de las propuestas más exitosas que han aparecido con este fin en la literatura recientemente, consiste en modelar la respuesta a impulso del sistema como una combinación lineal de un conjunto de funciones exponenciales ortogonales [2.25], [2.26]. Este sistema no presenta problemas de estabilidad, ya que sus polos son fijos, y su superficie de error es unimodal. Resultados experimentales muestran que este sistema presenta mejor funcionamiento que los filtros transversales con un menor número de coeficientes y por lo mismo, una menor complejidad computacional. Sin embargo sus polos fijos podrían llevar a un funcionamiento subóptimo.

Por lo descrito anteriormente se puede concluir que, aunque los sistemas de respuesta a impulso infinita son potencialmente la mejor alternativa, mucho trabajo resta por hacer antes de que estas estructuras puedan ser empleadas en aplicaciones prácticas.

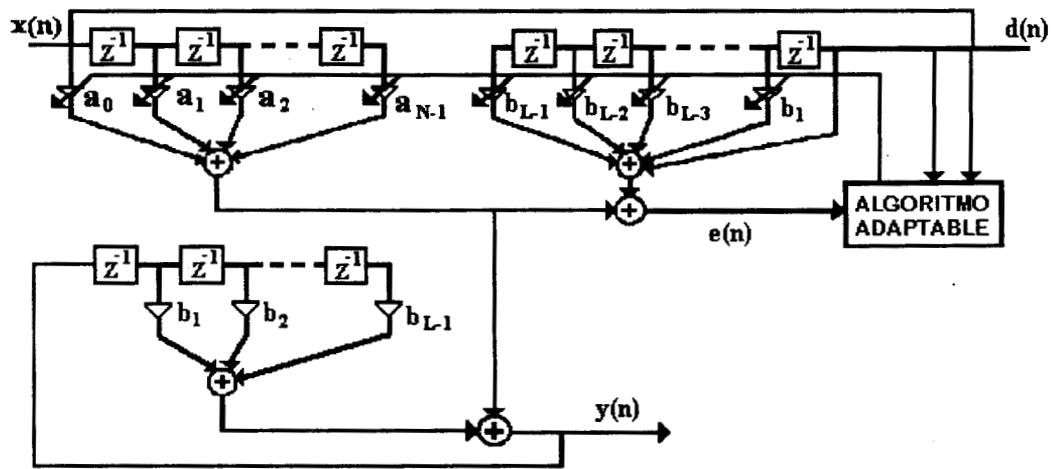


Figura 2.7. Filtro adaptable de respuesta a impulso infinita basado en el método de la ecuación de error.

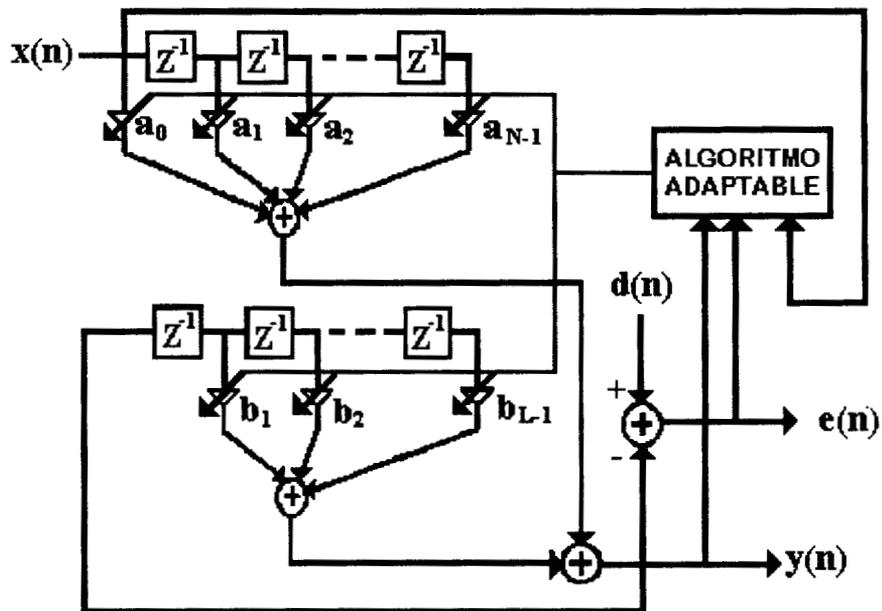


Figura 2.8. Filtro adaptable de respuesta a impulso infinita basado en el método del error de salida.

## **2.2.2 Aplicaciones de Filtrado Adaptable**

Seguidamente se mencionarán algunas de las aplicaciones mas comunes de los filtros adaptables con el fin de mostrar su utilidad en la solución de diversos problemas prácticos , así como ilustrar el proceso de selección de la señal de referencia.

### **2.2.2.1 Cancelación de Eco**

Un sistema telefónico de larga distancia consiste básicamente de una porción de 2 hilos , conocida como circuito de abonado, la cual comunica al abonado con la central telefónica local y el circuito de larga distancia propiamente dicho, el cual consiste de un canal de transmisión y otro de recepción, cada uno de los cuales consta de dos hilos. Se usa un transformador híbrido para acoplar los circuitos de larga distancia con el circuito de abonado, el cual idealmente aísla los canales de transmisión y recepción del circuito de larga distancia. Sin embargo debido al desacoplamiento de impedancias, estos no están completamente aislados, de manera que una porción de la señal recibida se retransmite en forma de eco, tal como se muestra en la figura 2.9.

Un problema similar se presenta en los sistemas de teleconferencia con el llamado eco acústico el cual se produce debido al acoplamiento que existe entre el micrófono y el altavoz en el sistema de teleconferencia. Esto da como resultado que una replica retardada y distorsionada de la señal producida por el altavoz se retroalimente hacia el micrófono.

En ambos casos se produce un deterioro en el sistema de comunicación, lo cual dio como resultado la aparición de los canceladores de eco. Estos han mostrado ser la mejor forma de resolver este problema [2.1], [2.3], [2.6], [2.13]-[2.21], [2.27]-[2.33]. El principio básico de cancelación de eco, el cual se ilustra en la figura 2.10, consiste en generar una réplica del eco el cual se sustrae de la señal a ser transmitida, resultando el llamado eco residual el cual consiste de la parte de la señal de eco que no pudo ser cancelada mas la voz del interlocutor cercano, en caso de que esta se encuentre presente [2.1], [2.4], [2.20], [2.31] . El eco residual se utiliza entonces para adaptar los parámetros del cancelador en forma tal que la potencia del eco residual es progresivamente minimizada.

Básicamente un cancelador de eco consiste de dos partes fundamentales. Un filtro adaptable, el cual genera una replica del eco la cual es sustraída de la señal transmitida, y un sistema conocido comúnmente como detector de " double talk", el cual evita la distorsión debida a la presencia de la voz del interlocutor cercano o en ausencia de la vos del interlocutor lejano. La primera componente, esto es la estructura del filtro adaptable junto con su algoritmo de adaptación, han sido mas estudiados que la segunda componente. Estas investigaciones han dado como resultado la aparición de diversas estructuras, tales como filtros transversales, estructuras en subbandas, estructuras en el dominio de la frecuencia, etc., así como diversos

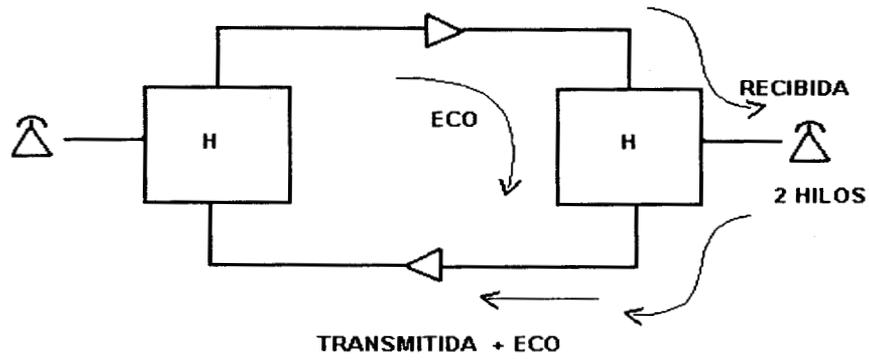


Figura 2.9. Eco en sistemas telefónicos debido a la conversión de 2 a 4 hilos.

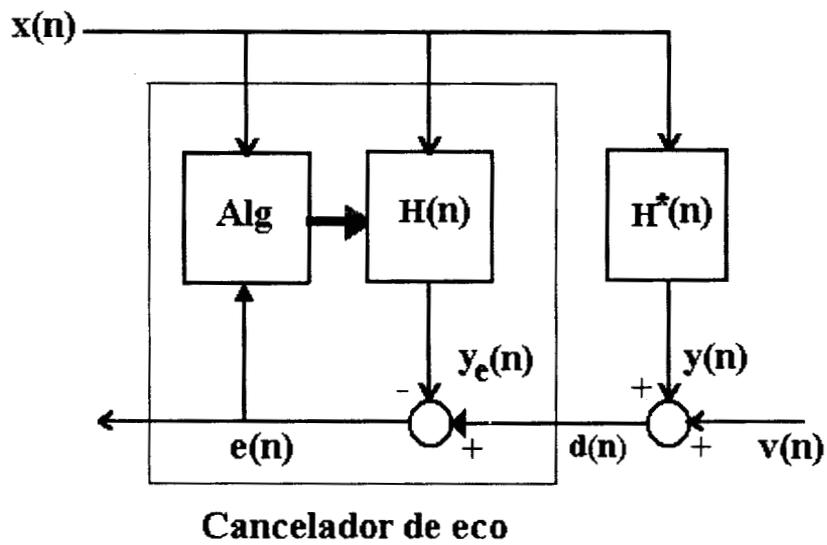


Figura 2.10. Configuración de básica usada en cancelación de eco.

algoritmos de adaptación, en su mayoría basados en la búsqueda del gradiente descendente.

Por su parte la segunda componente, a pesar de su importancia, ha recibido mucho menor atención que la primera. Así la realización de investigaciones tendientes al desarrollo de mecanismos altamente confiables que eviten la distorsión debida a la presencia simultanea de ambos interlocutores, detector de "double-talk" [2.28]-[2.33], especialmente cuando se emplean algoritmos basados en la búsqueda del gradiente descendente, es de gran importancia.

Bajo la condición de "double-talk" en la cual la señal local es añadida al eco, la estimación del canal de eco llega a ser errónea. El detector de "double-talk" detecta el inicio y el final del período de "double-talk" e inhibe la adaptación del cancelador, evitando así un funcionamiento inadecuado del sistema.

### **2.2.2.2 Predicción Lineal**

En varias situaciones es necesario estimar valores futuros de cierto tipo de señales útiles, a partir de muestras pasadas de la misma señal. Esto se puede conseguir, siempre y cuando las muestras de las señales deseadas se encuentren correlacionadas entre si, usando un filtro adaptable operado como predictor lineal. Estructura de predictor lineal se muestra figura 2.11. Aquí la señal de entrada es retarda T períodos de muestreo, donde T es la diferencia entre el instante actual y el instante en el cual se desea estimar el valor de la señal. Así por ejemplo si se desea estimar el valor que la señal tomaría cinco periodos de muestreo a partir del instante actual T sería igual a cinco, asumiendo sin perdida de generalidad, el periodo de muestreo igual a uno. La señal no retardada sirve como señal de referencia. El sistema converge y produce, en el sentido de mínimos cuadrados, el mejor estimado de la señal no retardada. Los coeficientes óptimos son entonces copiados a un "filtro esclavo" cuya entrada es no retardada y cuya salida es el mejor estimado, en el sentido de mínimos cuadrados, de una muestra T unidades de tiempo en el futuro [2.4], [2.5], [2.12], [2.17].

### **2.2.2.3 Igualación de Canales de Comunicación**

Los canales de comunicación, en general, tienen respuesta en frecuencia cuya magnitud es no plana en la banda de interés y su fase es no lineal degradando considerablemente las señales transmitidas a través de ellos. Esto, en comunicación digital, causa un fenómeno llamado "interferencia entre símbolos" el cual considerablemente degrada los sistemas de comunicación digital al introducir errores de detección. Un método comúnmente usado para combatir este problema es el uso de igualadores adaptables, los cuales reducen considerablemente este fenómeno filtrando las señales recibidas y modificando sus coeficientes en forma tal que el error de detección sea minimizado. Aquí la señal de entrada del sistema es la señal recibida

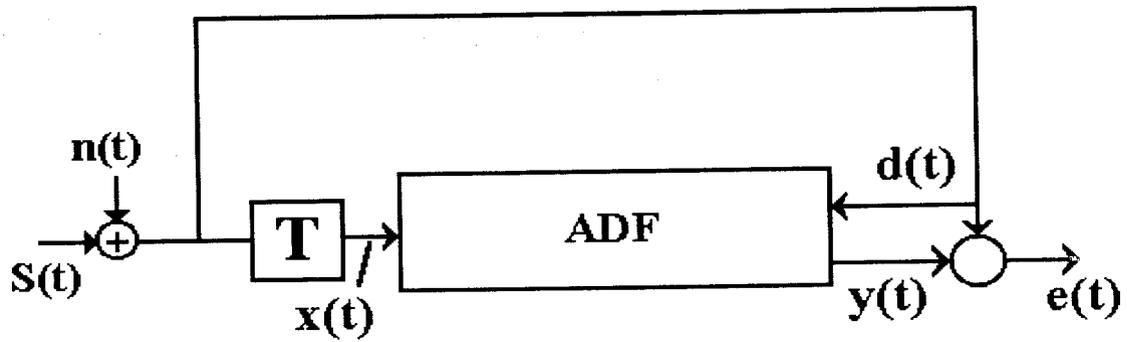


Figura 2.11. Predicción lineal adaptable.

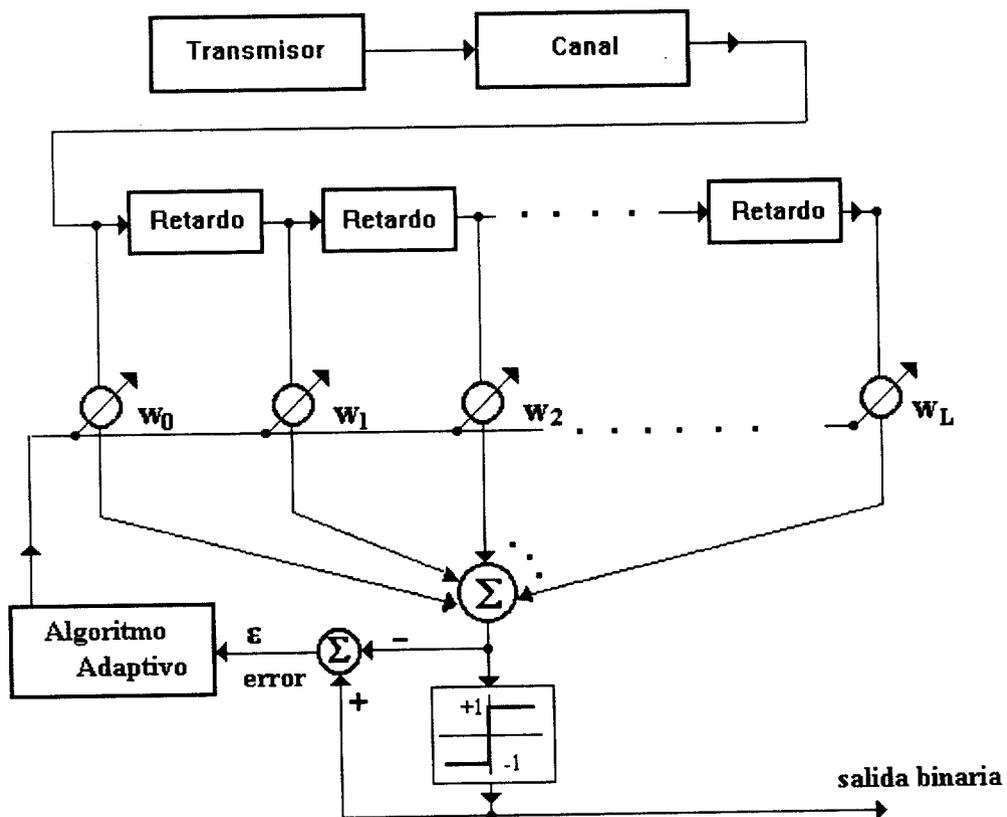


Figura 2.12. Configuración básica de un igualador transversal adaptable.

mientras que la referencia es, durante el periodo de entrenamiento, una secuencia conocida de antemano y durante la operación normal los símbolos detectados.

Un igualador consiste, básicamente, de uno o mas filtros adaptables cuya salida se encuentra conectada a un elemento no lineal. La estructura de igualador general se muestra en la figura 2.12. Es decir un igualador podría pensarse, en cierta forma, también como una red neuronal de tiempo retardado cuyos coeficientes se adaptan usando un algoritmo adaptable, ya sea basado en la búsqueda del gradiente, del tipo Newton-Rapson, etc [2.4], [2.5], [2.34].

Intensa investigación se llevado a cabo durante las últimas décadas en el campo del filtrado adaptable. Durante este tiempo se han propuesto una gran cantidad de algoritmos de adaptación y estructuras, las cuales intentan resolver diversos problemas como mencionados anteriormente. Sin embargo, pese a este considerable esfuerzo y notorios avances obtenidos, aun permanecen varios problemas sin solución entre los cuales tenemos: Algoritmos de adaptación que proporcionen bajos niveles de error con moderada complejidad computacional y rápida convergencia. Este sigue siendo un problema fundamental pues, en general los algoritmos con baja complejidad computacional y relativamente bajos niveles de error convergen lentamente [2.1]-[2.3]. De manera que el desarrollo de algoritmos con complejidad baja, bajos niveles de errores de estimación y convergencia moderadamente rápida es un problema fundamental. Otro problema de gran importancia es el desarrollo de nuevas estructuras que permitan reducir la complejidad de los sistemas adaptables actuales, sin degradar su funcionamiento, de manera que estos se puedan emplear en la solución de problemas en tiempo real.

## 2.3 Redes Neuronales

En las últimas dos décadas se ha observado un importante interés en el desarrollo de sistemas que pretenden observar un comportamiento inteligente. El desarrollo de las redes neuronales da respuesta a esta inquietud convirtiéndose en un campo con enormes posibilidades de desarrollo [2.35]-[2.40], estas posibilidades se amplían con las investigaciones que se realizan en nuevos algoritmos que proponen soluciones importantes a problemas como: la clasificación o reconocimiento de patrones, los cuales se pueden clasificar en reconocimiento de objetos [2.41], [2.42], reconocimiento de caracteres manuscritos [2.43]-[2.45], reconocimiento de fonemas [2.46]-[2.48], reconocimiento de firmas [2.49], [2.50], remoción de ruido [2.36], [2.51], optimización [2.52], [2.53], restauración y compresión de imágenes [2.54], [2.55], filtros adaptables [2.56], [2.57] entre otros. La realización de redes neuronales mediante estructuras de silicio con integración a gran escala (V.L.S.I.) propondrá soluciones cada vez mas cercanas con las estructuras y algoritmos que propone la teoría general de las redes neuronales.

La búsqueda de una inteligencia no biológica es una vieja aspiración del hombre moderno. Al menos desde la década de los sesentas hay esfuerzos sistemáticos que proponían a la inteligencia artificial [2.58] como la alternativa. Este campo que recibió importantes apoyos en países desarrollados como es el caso de los Estados Unidos de Norteamérica. La Inteligencia Artificial (I. A.) propone, básicamente, la solución de problemas tales como la identificación y el reconocimiento de patrones, la estimación o predicción de eventos, etc., los cuales son problemas que se resuelven con gran eficacia por la inteligencia biológica, mediante computadoras digitales. Esta ha desarrollado un intenso trabajo de investigación aportando importantes conceptos los cuales han ayudado entender mejor los mecanismos del pensamiento humano.

Otra alternativa importante en la búsqueda de inteligencia no biológica lo han sido las redes neuronales. En estas el concepto del que se parte es la movilidad de la interconexión de elementos que conforman la arquitectura de la red, proponiendo un camino de desarrollo que parte de los principios básicos de funcionamiento de las neuronas biológicas. Partiendo de estos conceptos se ha empezado a construir una teoría axiomática y sistematizada, la cual intenta desarrollar inteligencia no biológica o sistemas con la capacidad de aprender, obteniéndose hasta la fecha resultados sorprendentes los cuales, en muchos casos, han superado ya a las soluciones propuestas por la I.A.

Las redes neuronales artificiales son sistemas que se constituyen de un gran número de unidades de procesamiento que se conectan entre si de acuerdo con un patrón preestablecido, en las cuales existe una relación funcional que liga a las entradas con las salidas de la misma. Debido a su capacidad de aprendizaje, es posible entrenar a las redes neuronales para que puedan almacenar, ordenar o reconocer información. En este sentido la propuesta de las redes neuronales es en esencia la búsqueda de una arquitectura que trabaje de manera análoga a como

lo hace el cerebro, guardada toda proporción, pues sabemos que la mente humana tiene alrededor de cien millones de neuronas y el número de conexiones que puede establecer cada neurona es de hasta diez mil. En relación a estas cifras es claro que las redes neuronales artificiales están en el inicio de su infancia, esperando que el campo madure con resultados que seguramente sorprenderán.

### 2.3.1 Neuronas

Una neurona en general consiste de un Combinador Lineal Adaptable (ALC) cuya señal de salida se encuentra conectada en cascada con un elemento no lineal [2.35]-[2.40], el cual produce la decisión o salida de la neurona. Aquí, al igual que en los filtros adaptables, las neuronas son entrenables usando un algoritmo adaptable, o de aprendizaje, siendo el algoritmo LMS, o algoritmo de retropropagación [2.35]-[2.37], el mas ampliamente usado. El ALC es un modelo de una neurona, estando el vector de entrada relacionado a las entradas dendríticas, el vector de coeficientes a las sinapsis y la salida del elemento no lineal, cuya entrada es una suma pesada del vector de entrada, y la salida del ALC con la salida axonal.

De lo anterior se desprende que existe una gran similitud entre el funcionamiento de una neurona y el de un filtro adaptable. Sus principales diferencias radican en el elemento no lineal que aparece en las neuronas, así como en el hecho de que las neuronas se encuentran, en general, conectadas varias de ellas formando redes, las cuales se conocen como redes neuronales, mientras que los filtros adaptables consisten, en la mayoría de los casos, de un solo ALC.

Hoy en día las redes neuronales han despertado un gran interés entre la comunidad científica debido a su gran potencial en la solución de problemas tales como reconocimiento de patrones, compresión de imágenes, restauración de imágenes, control, etc. [2.41]-[2.57]. Aunque las redes neuronales no han tenido aun el impacto de los filtros adaptables, su potencial es enorme y se espera que en breve estos sistemas se puedan aplicar con gran éxito en la solución de diversos problemas de interés práctico.

Los puntos comunes de los filtros adaptables y las redes neuronales sugieren que los dos campos tienen mucho que ver entre sí, permitiendo de esa manera que avances logrados en un campo puedan ser, en muchos casos, aplicados en el otro.

Las neuronas, por su parte, son los elementos básicos que forman las redes neuronales, se muestra en la figura 2.13. Aquí, vector  $x=(x_1, x_2, \dots, x_n)$  es entrada y  $w$  es el vector de pesos de conexión, y es el producto punto del vector de entrada  $x$  y el vector de pesos,  $u$  es salida de la neurona, el cual es valor de la función no lineal  $f(y)$ .

Las funciones no lineales que se usan normalmente en una neurona, son tres, las cuales se

Las funciones no lineales que se usan normalmente en una neurona, son tres, las cuales se muestran en la figura 2.14. La función “hard limit” se usa cuando se requiere que la salida de la red sea binaria. Esta se calcula en siguiente manera:

$$f_h(y) = \begin{cases} 1 & y > 0 \\ 0 & y \leq 0 \end{cases} \quad (2.1)$$

La función sigmoideal se usa ampliamente en una red de multicapas, debido a que esta es diferenciable. La función sigmoideal esta dada por la siguiente ecuación.

$$f_s(y) = \frac{1}{1 + e^{-\beta y}} \quad (2.2)$$

Dependiendo del valor  $\beta$ , la función sigmoideal se acerca una función lineal de 0.0 a 1.0, con un valor de  $\beta$  pequeño, o se acerca a una función “hard limit”, con un valor de  $\beta$  grande, como se muestra figura 2.15. La función no lineal conocida como “threshold logic” la cual se muestra en la figura 2.14c se calcula como sigue:

$$f_l(y) = \begin{cases} 0 & y < 0 \\ y & 0 < y < 1 \\ 1 & 1 < y \end{cases} \quad (2.3)$$

### 2.3.2 Principales Estructuras de Redes Neuronales

Durante última década se han propuesto varias estructuras de redes neuronales para diversas aplicaciones practicas. Estas estructuras de redes neuronales se pueden clasificar tomando en cuenta su manera de aprendizaje y su estructura, por ejemplo si la salida de la red es retroalimenta a la misma red o no, [2.36], [2.37]. Aquí, las redes neuronales se clasificarán por su manera de aprendizaje. Para entrenar la red neuronal, existe dos manera de aprendizaje, las cuales son aprendizaje con supervisor y aprendizaje sin supervisor. En el aprendizaje con supervisor, se presenta a la entrada de la red un vector de datos que consiste de datos de entrenamiento y la salida deseada de la red durante el entrenamiento, y se modifican los pesos de las conexiones de la red de manera que la salida de la red se aproxima a salida deseada. Mientras que en el aprendizaje sin supervisor, se presenta solamente el vector de datos de entrada y la red modifica entonces sus pesos de una manera autoorganizativa, de acuerdo a la similitud existente entre algunos datos de entrada. La redes neuronales representativas del

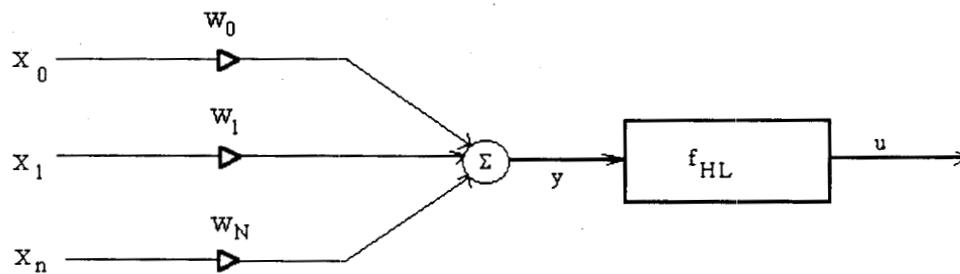


Figura 2.13. Configuración básica de un neurona.

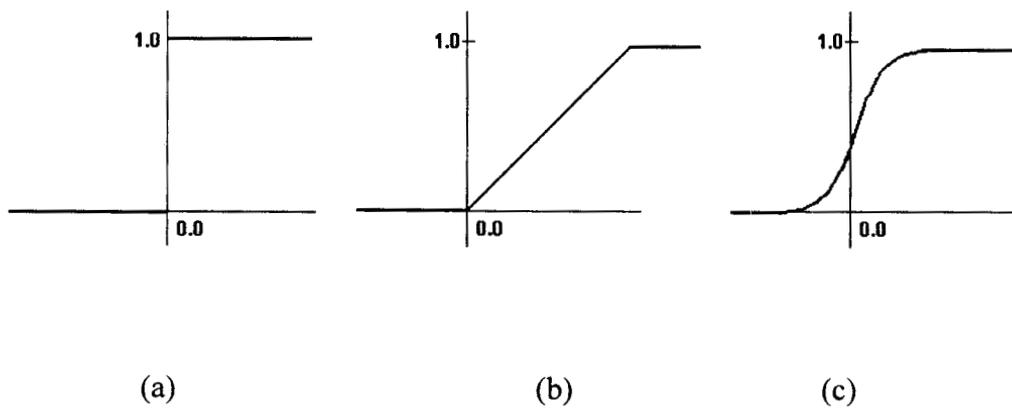


Figura 2.14. Algunos tipos de funciones no lineales usadas como etapa de salida en las neuronas artificiales. (a) "Hard limit", (b) Función lineal a tramos, (c) umbral lógico o "Threshold logic".

aprendizaje con supervisor son las redes Adaline y Madaline, el Perceptron Multicapas, las Redes Neuronales de Tiempo Retardado (TDNN), y la Red de Hopfield. Mientras que las redes neuronales mas representativas del aprendizaje sin supervisor son la red de Kohonen y la red Neocognitron. La figura 2.16 muestra la clasificación de las redes neuronales representativas de estos dos grandes grupos.

### 2.3.2.1 Redes Neuronales con Supervisor

Como se mencionó anteriormente, las redes Adaline y Madaline, el Perceptron Multicapas, la Red Neuronal de Tiempo Retardado y la Red de Hopfield, etc. se pueden clasificar en el grupo de redes neuronales con supervisor. A continuación se presenta un breve descripción de cada una de ellas.

#### Redes Adaline y Madaline

Las redes Adaline y Madaline fueron desarrolladas por Bernard Widrow en el año de 1963 [2.35]. El termino Adaline viene de ADaptive LINear Element, y su estructura se muestra en la figura 2.17. En esta figura,  $x_1, x_2, \dots, x_n$  son los datos de entrada,  $x_0$  es la entrada de sesgo, cuyo valor es siempre igual a 1,  $w_0, w_1, \dots, w_n$  son los pesos de las conexiones de la red, y es el producto punto del vector de datos de entrada y el vector de pesos de la red. La salida de la red se obtiene entonces después de aplicar una función no lineal "hard limit". Por lo tanto el valor de salida de la red es binario.

Salida de la red de Adaline se calcula usando la siguiente ecuación.

$$y = W'X \quad (2.4)$$

$$u = f_h(y) \quad (2.5)$$

Aquí,  $X$  es vector de entrada y  $W$  es vector de pesos de las conexiones. El primer elemento del vector  $X$ ,  $x_0$  es siempre igual a 1. En la red Adaline, los pesos de las conexiones se adaptan usando un algoritmo de aprendizaje basado, generalmente, en el método de gradiente, esto es el algoritmo LMS o alguna variación del mismo, de manera que su salida sea el mejor estimado de la salida deseada en el sentido de mínimos cuadrados. Este esquema de adaptación de la red Adaline se muestra en la figura 2.18. El algoritmo de LMS se mencionará con detalle en el siguiente capítulo (capítulo 3). La red Adaline, por su estructura sencilla, tiene algunas limitaciones en su funcionamiento, por ejemplo, en la red Adaline no puede resolver el problema de XOR, porque la partición del espacio de la señal de entrada no se puede clasificar linealmente, como muestra en la figura 2.19. Para resolver este problema, se propuso la red

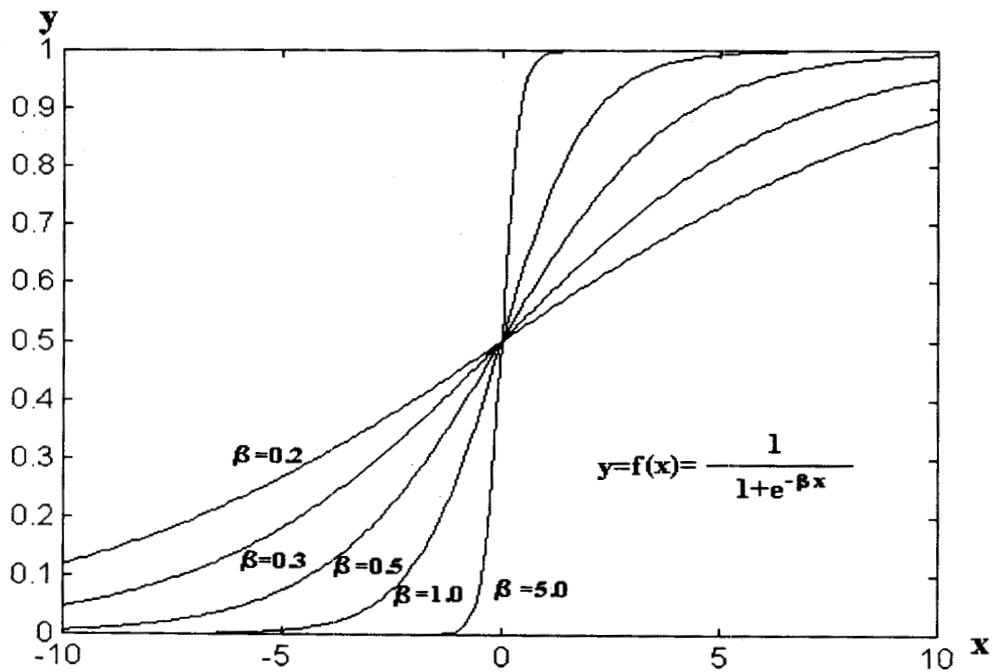


Figura 2.15. Función no lineal sigmoidal conocida como “threshold logic” con diferentes valores de  $\beta$ .

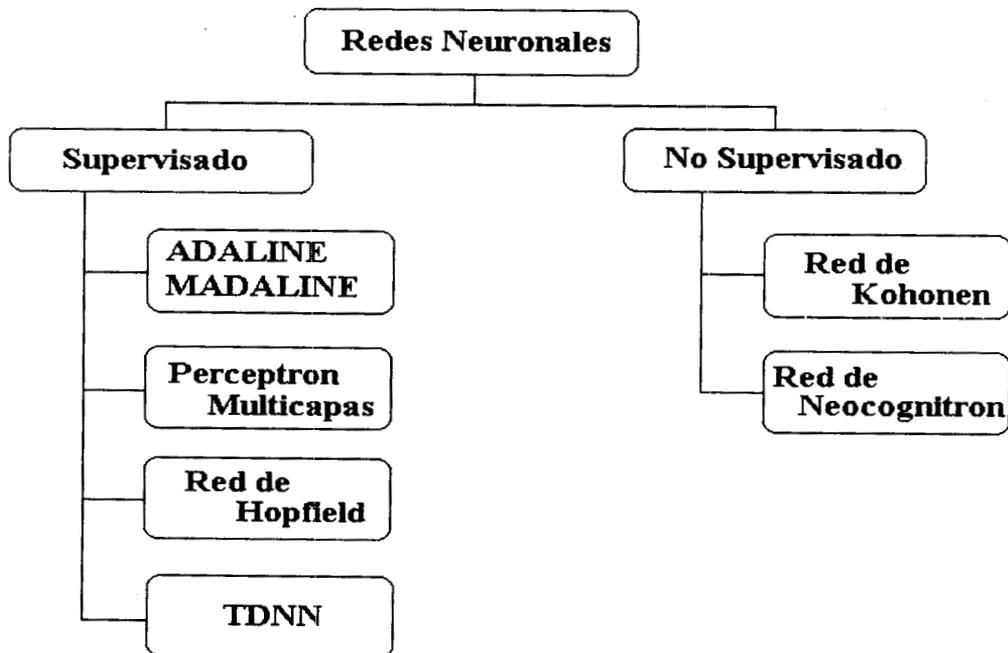


Figura 2.16. Clasificación de las redes neuronales atendiendo a su tipo de entrenamiento, ya sea con supervisor o sin supervisor.

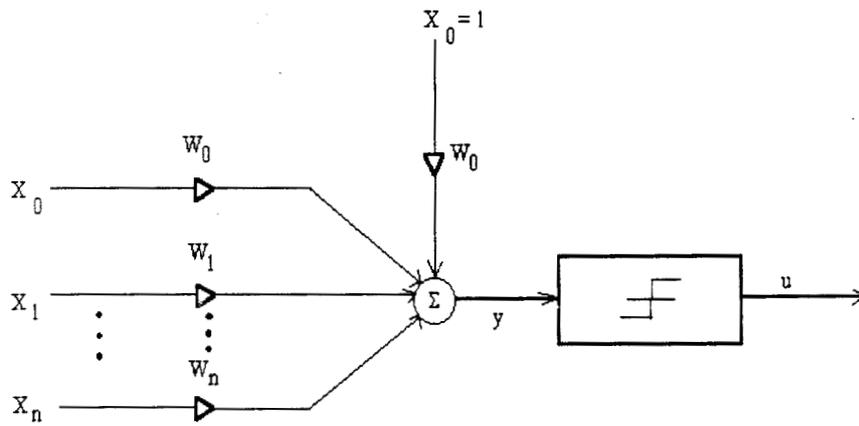


Figura 2.17. Configuración básica de un Elemento lineal adaptable, "ADaptive LINear Element" (ADALINE).

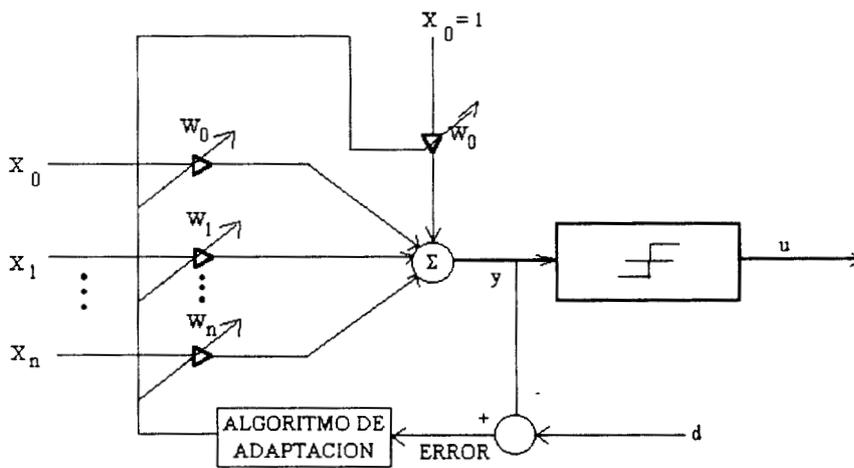


Figura 2.18. Método de entrenamiento de un elemento lineal adaptable (ADALINE).

de Madaline, cuyo termino viene de Many Adaline. La estructura de la red Madaline se parece al perceptron multicapas, mostrado en la figura 2.20. La única diferencia entre la red Adaline y el perceptron multicapas es la función no lineal usada en cada red. Ya que mientras en la red Madaline se usa función de “hard limit”, en la red perceptron multicapas se emplea una función sigmoideal. Debido a la función no lineal usada en la red Madaline, ésta presenta dificultades para entrenamiento.

### Perceptron Multicapas

El perceptron multicapas con su algoritmo de aprendizaje (retropropagación) se han usado en la solución de varios problemas prácticos, tales como, reconocimiento de caracteres manuscritos [2.43], [2.44], reconocimiento de objetos [2.42], y reconocimiento de firmas [2.49], [2.50].

La estructura del perceptron multicapas se muestra en la figura 2.21. En la figura, la red tiene una capa de entrada, dos capas ocultas y una capa de salida, esta red se llama red Perceptron de 3 capas. El numero de capas ocultas de la red y el numero de neuronas en cada capa oculta, son determinadas por el usuario dependiendo de la aplicación, sin embargo no existe método de determinar los valores óptimos. Se puede probar que la red perceptron de dos capas ( capa de entrada , una capa oculta y capa de salida) puede clasificar cualquier número de grupos usando una frontera de forma arbitraria. [2.37], [2.59]. Sin embargo la red perceptron de 3 capas con pocas neuronas en sus dos capas oculta puede funcionar igual que la red perceptron de 2 capas con una gran cantidad de neuronas en su capa oculta [2.36], [2.37]. El perceptron multicapas entrenado adecuadamente tiene un excelente funcionamiento cuando se usa en un importante número de aplicaciones prácticas [2.44], Sin embargo, la complejidad temporal del entrenamiento es sumamente alta, esto dificulta su uso en algunas aplicaciones prácticas. Debido a esto, se han propuestos varios métodos para reducir dicha complejidad, los cuales son “Prunning” , “weight sharing”, etc. Estos métodos propuestos tratan de reducir numero de conexiones entre las neuronas, ya que después de aplicar cualquiera de ellos las neuronas de la red ya no están completamente conectadas unas con otras, sino las conexiones son locales.

La salida de la red Perceptron Multicapas que se muestra en la figura 2.21 se calcula por medio de las ecuaciones (2.6)-(2.8).

$$y_l = f\left(\sum w''_{k,l} x''_k - \theta''_l\right) \quad 0 \leq l \leq M - 1 \quad (2.6)$$

$$x''_k = f_s\left(\sum_{j=0}^{N_1-1} w'_{j,k} x'_j - \theta'_k\right) \quad 0 \leq k \leq N_2 - 1 \quad (2.7)$$

$$x'_j = f_s\left(\sum w_{i,j} x_{ij} - \theta_j\right) \quad 0 \leq j \leq N_1 - 1 \quad (2.8)$$

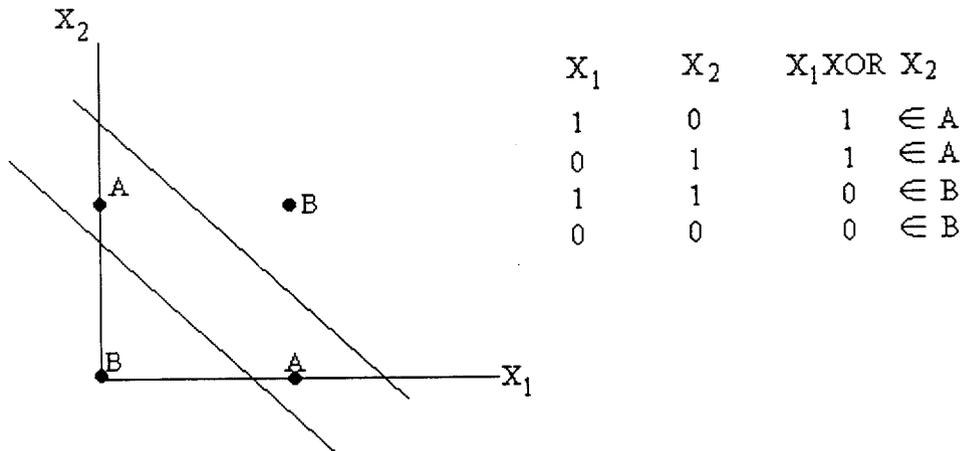


Figura 2.19. Como se muestra el problema XOR ilustrado en esta figura no puede ser resuelto usando un sistema lineal adaptable (ADALINE).

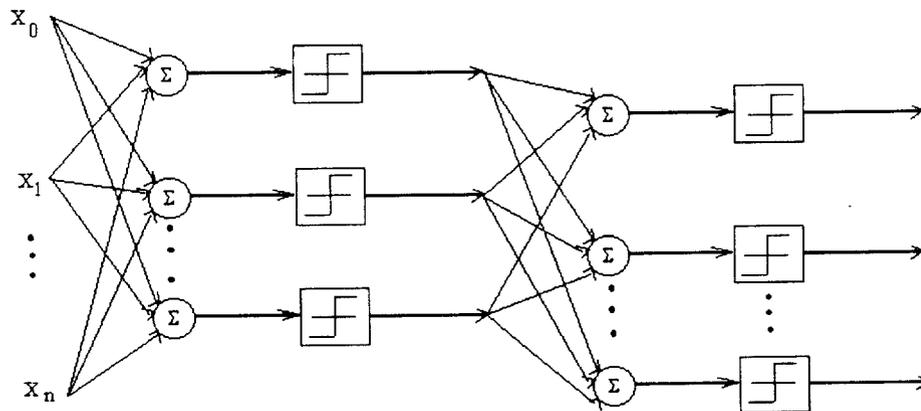


Figura 2.20. Red Neuronal Madaline 0 "Many adaline" la cual permite, entre otras cosas resolver el problema XOR. La no linealidad usada en esta red es un "hard limit".

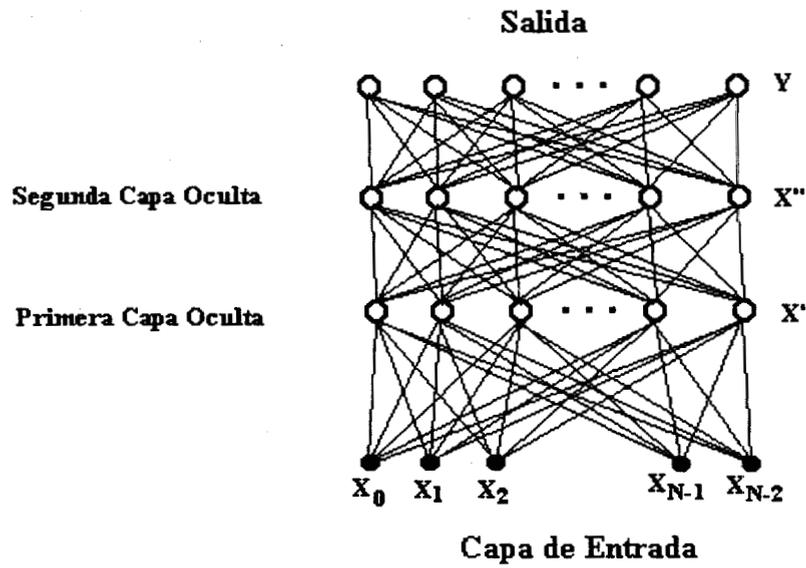


Figura 2.21. Perceptron multicapas con dos capas ocultas.

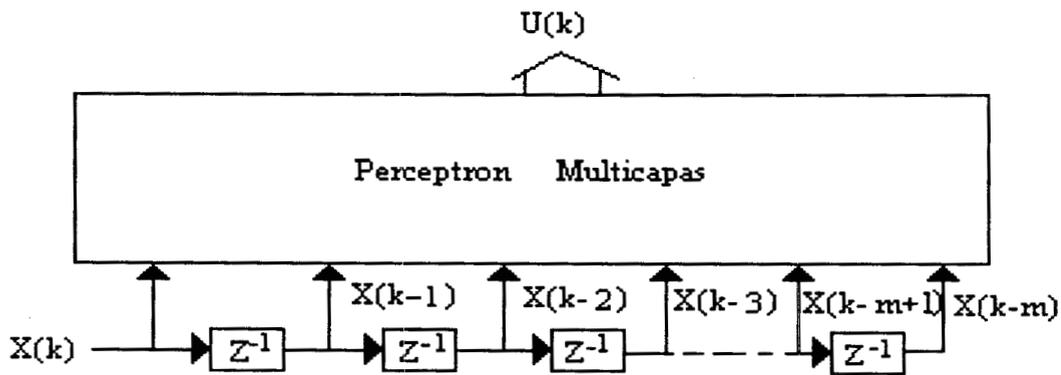


Figura 2.22. Red neuronal de tiempo retardado o "Time delay neural network" DTNN.

La función  $f_s(\alpha)$  es en general alguna función no lineal, aunque en ocasiones podría ser lineal. La función  $f_s(\alpha)$  de las ecuaciones (2.7) y (2.8) es la función sigmoide, la cual es la más comúnmente usada en la mayoría de las aplicaciones prácticas.

### **Red Neuronal de Tiempo Retardado**

La red neuronal de tiempo retardado tiene misma estructura que la red perceptrón multicapas, aunque aquí la entrada de la red es una secuencia temporal de datos en lugar de patrón estático como se muestra en la figura 2.22. En esta figura se observa que la señal de entrada pasa por una línea de retardo, seguidamente la secuencia temporal de la señal de entrada se aplica a la red perceptrón multicapas. Esta red neuronal puede manejar información temporal, por lo tanto se usa ampliamente para el reconocimiento de fonemas o identificación del hablante.

Para el entrenamiento de la red, normalmente se usa el algoritmo de retropropagación como en la red perceptrón multicapas.

### **Red Neuronal de Hopfield**

La red de Hopfield fue desarrollada por Hopfield en el año de 1982. Los pesos de las conexiones de la red se calculan de antemano usando correlación de los patrones representativos de cada clase. Por lo mismo la red neuronal de Hopfield no necesita entrenamiento. La estructura de la red neuronal de Hopfield se muestra en la figura 2.23. De la figura, la salida de la red neuronal de Hopfield se retroalimenta a todas las neuronas, por lo tanto, la red neuronal de Hopfield se puede clasificar como una red neuronal dinámica.

En la figura, el vector  $\mathbf{X}=(x_0, x_1, \dots, x_{n-1})$  es el vector de entrada,  $\mathbf{U}=(u_0, u_1, \dots, u_{n-1})$  es el vector de estado de las neuronas y  $\mathbf{Y}=(y_0, y_1, \dots, y_{n-1})$  es el vector de salida, cuyos valores se calculan en las ecuaciones (2.9), (2.10).

$$u_i(k) = \sum_{j=1}^N w_{ij} y_j(k) + x_i \quad (2.9)$$

$$y_i(k+1) = f_h(u_i(k)) \quad (2.10)$$

donde  $f_h(\alpha)$  es la función no lineal "hard limit".

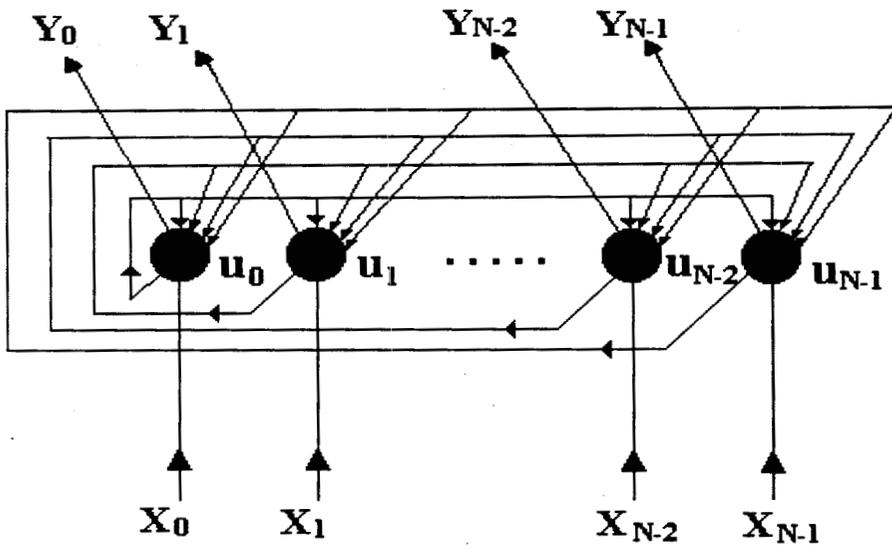


Figura 2.23. Red neuronal de Hopfield

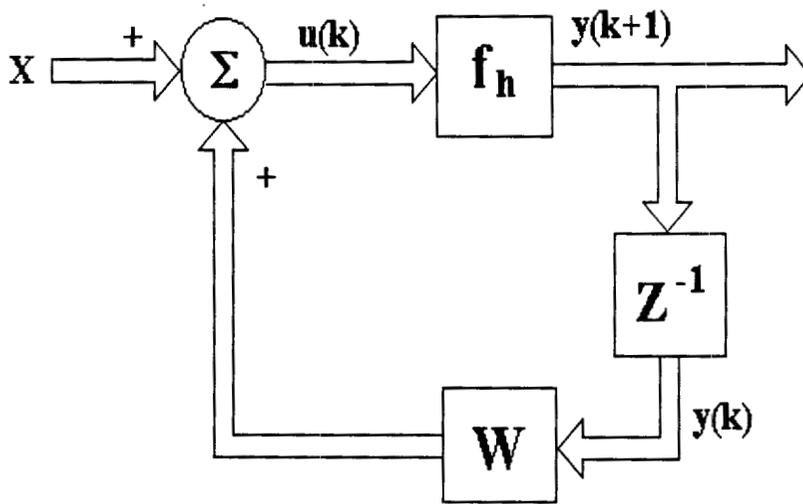


Figura 2.24. Diagrama que ilustra el funcionamiento de la red neuronal de Hopfield.

El diagrama del sistema se muestra en la figura 2.24. La red de Hopfield se usa para resolver problemas de optimización, tal como el problema de agente viajero.

### 2.3.2.2 Red Neuronal sin Supervisor

Mayoría de las redes neuronales propuestas pertenecen al grupo de redes neuronales con supervisor. Sin embargo algunas redes neuronales como red de Kohonen y la red Neocognitron pertenecen al grupo de redes neuronales sin supervisor. En las redes neuronales sin supervisor, los valores de los pesos de las conexión se adaptan solo con los vectores de entrada, sin la necesidad de una salida deseada. En la mayoría de los casos los pesos de las conexiones se adaptan usando la similitud existente entre los datos de entrada por medio de aprendizaje por competencia o aprendizaje de elite. Por el hecho de que estas redes organizan el valor de sus pesos automáticamente sin supervisor externo, se conocen como redes neuronal autoorganizativas.

#### Red de Kohonen

La red de Kohonen, desarrollada por T. Kohonen en 1984 como mapas característicos autoorganizativos [2.46], es una de las redes mas representativa del grupo de redes neuronales sin supervisor. La estructura de la red de Kohonen, se muestra en la figura 2.25. Como se muestra la figura, la red neuronal de Kohonen consiste de la capa de entrada y la capa de salida, y cada unidad de entrada se conecta con todas las unidades de la capa de salida. En la red de Kohonen la adaptación de los pesos de las conexiones se realiza localmente. Después de la presentación del dato de entrada, la red calcula la distancia entre la entrada y la salida de todas neuronas de la capa de salida.

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{i,j}(t))^2 \quad (2.11)$$

donde  $x_i(t)$  es valor de  $i$ -esimo elemento del vector de entrada,  $w_{i,j}(t)$  es el peso de la conexión que va desde el  $i$ -esimo nodo de la capa de entrada hasta  $j$ -esimo nodo de la capa de salida, y  $d_j$  es la distancia entre entrada y  $j$ -esimo nodo de la capa de salida. Los pesos de las conexiones han sido inicializadas de antemano con valores aleatorios pequeños. La adaptación se realiza entonces en los pesos que se conectan con el nodo de salida que tenga la distancia mínima y con los nodos alrededor de este. Por ejemplo el nodo  $j^*$  tiene distancia mínima, adaptación de los pesos se calcula en siguiente ecuación.

$$w_{i,k}(t+1) = w_{i,k}(t) + \eta(t)(x_i(t) - w_{i,k}(t)) \quad (2.12)$$

$$k \in NC_{j^*}(t) \quad (2.13)$$

donde termino  $\eta(t)$  es una función monotonamente decreciente en tiempo.  $NC_{j^*}(t)$  significa una área vecindad del nodo  $j^*$  y cuyo tamaño decreciente en tiempo.

### Red Neuronal Neocognitron

La mayoría de las redes neuronales propuestas no están basados en una investigación biológica del cerebro humano. La red neuronal Neocognitron, la cual es una red específica para el reconocimiento de caracteres manuscritos, ha sido desarrollada como un modelo del cerebro humano por K. Fukushima en 1980 [2.45]. La red Neocognitron consiste de 9 capas incluyendo una capa de entrada. La estructura de la red se muestra en la figura 2.26. La capa  $U_0$  es la capa de entrada en la cual se presenta un carácter manuscrito. Las capas  $U_s$  y  $U_c$  construyen un modulo, el cual funciona como un extractor de características de una parte pequeña de los caracteres manuscritos tales como un segmento, curva, etc. Las capas  $U_{s1}, U_{s2}, U_{s3}, U_{s4}$  tienen neuronas de tipo S, las cuales coinciden con neuronas sencillas o con neuronas complejas de bajo orden. Las capas  $U_{c1}, U_{c2}, U_{c3}, U_{c4}$  tienen neuronas de tipo C, las cuales coinciden con neuronas complejas o neurona complejas de alto orden, según la clasificación del sistema visual propuesto por Hubel y Wiesel. [2.60].

En cada capa del Neocognitron existen varios planos, los cuales funcionan como extractores de diferentes características de una parte de los datos de entrada. Las características extraídas en capa  $U_s$  se transmiten a la siguiente capa  $U_c$ , absorbiendo sus posiciones y combinando las características sencillas en un caracter mas complejo. Las capas menos profundas o sea las mas cercanas a la capa de entrada extraen las características mas sencillas, por ejemplo una parte de un segmento o una parte de una curva, etc. Por su parte las capas mas profundas tienen la capacidad de extraer características combinadas y mas amplias. Finalmente la ultima capa  $U_{c4}$  puede extraer las características completas del patrón de entrada.

El entrenamiento de la red neuronal Neocognitron consiste en la modificación de las conductancias de las sinapsis las cuales son fijas o variables. Con este fin se presenta el patrón de entrenamiento en la primera capa la cual equivale a la retina en la sistema visual humano, y los patrones de entrenamiento para cada una de las capas, siendo necesario escoger cuidadosamente los patrones de entrenamiento para entrenar cada una de estas, ya que estos patrones de entrenamiento considerablemente afectan al funcionamiento de la red. La selección de éstos patrones dependen del tipo de objetos que se quiere reconocer, o sea los patrones de entrenamiento usados para cada una de las capas para el reconocimiento de dígitos y para el reconocimiento de caracteres manuscritos son diferentes.

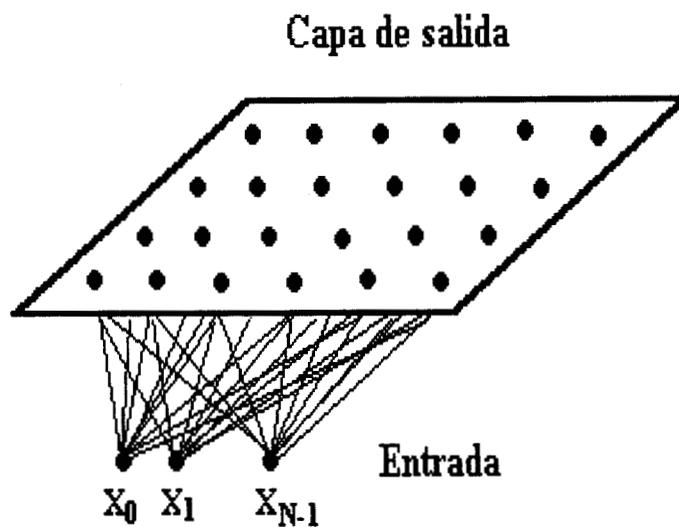


Figura 2.25. Red neuronal de Kohonen

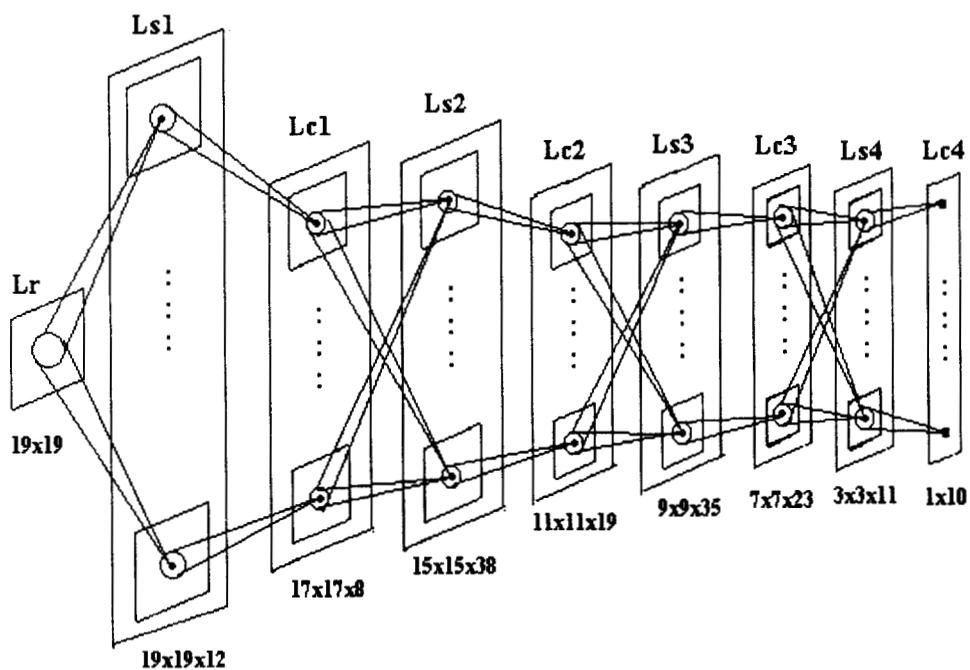


Figura 2.26. Red neuronal neocognitron

### **2.3.3 Aplicación de Redes Neuronales**

La red neuronal misma es un tema de investigación amplio, que puede comprender varias líneas de investigación. Por ejemplo las propiedades de convergencia de la red, la complejidad computacional (temporal y espacial) y la representación interna de la red que tienen las capas ocultas, etc. son temas que se han investigado ampliamente en diversos países. Además de estas investigaciones básicas, las redes neuronales se han aplicado en la solución de diversos problemas prácticos en diversos campos de la ciencia e ingeniería, tales como el procesamiento de señales, procesamiento de imágenes y control, etc. como una herramienta o un método nuevo. En esta sección, se presentan dos de las principales aplicaciones de redes neuronales: El reconocimiento de patrones y el problema de optimización.

#### **2.3.3.1 Reconocimiento de patrones**

Una de las principales aplicaciones prácticas de las redes neuronales está en el reconocimiento de patrones, entre los que incluyen aplicaciones tales como: El reconocimiento de caracteres manuscritos, el reconocimiento de objetos, el reconocimiento de firmas, el reconocimiento de rostros, el reconocimiento de huellas digitales, reconocimiento de fonemas y la identificación de hablantes.

Entre los tipos de reconocimiento de patrones mencionados anteriormente, el reconocimiento de caracteres manuscritos ha sido un tema de investigación muy intenso, debido a sus potenciales aplicaciones, tales como sistema automático de distribución de cartas usando el código postal y compresión de datos de facsímil, etc.

Para el reconocimiento de caracteres manuscritos, se ha usado principalmente la red neuronal perceptrón multicapas entrenada usando el algoritmo de retropropagación [2.43], [2.44]. El funcionamiento de esta red se ha investigado ampliamente obteniéndose como resultado que, después de gran cantidad de entrenamiento, la red reconoce aproximadamente 95% de los patrones de entrada. También se ha investigado la compresión de información de documentos manuscritos con el fin de transmitirse con un menor costo y tiempo, para lo cual también se ha usado esta red neuronal la cual reconoce las letras manuscritas y las convierte a código ASCII. [2.61].

El reconocimiento de rostro, el reconocimiento de huellas digitales y la identificación del hablante tienen potenciales aplicaciones en sistemas de seguridad, los cuales permiten el acceso a determinados lugares exclusivamente algunas personas (cuarto, edificio, etc.). Para llevar a cabo esta aplicación se ha investigado la red de perceptrón multicapas entrenada usando el algoritmo de retropropagación, así como la red de tiempo retardado, y la red neuronal de

Hopfield. [2.47], [2.48].

También el reconocimiento de fonemas tiene, potencialmente, bastantes aplicaciones, tales como venta o reservación automática por teléfono, etc. Para llevar a cabo esta aplicación se ha investigado ampliamente la red neuronal de tiempo retardado y la red neuronal recurrente, ambas entrenadas usando el algoritmo de retropropagación. Al igual que sucede con el reconocimiento de caracteres manuscritos, después de un tiempo de entrenamiento suficientemente largo, la red reconoce aproximadamente 80 – 90 % de las palabras aisladas que se encuentran dentro de un vocabulario limitado. Sin embargo, todavía el funcionamiento de la red neuronal es muy bajo cuando se requiere reconocer palabras aisladas pertenecientes a vocabularios consistentes de un gran número de palabras, así como vos continua tal como una oración completa. [2.47].

El reconocimiento de objetos puede ser de gran utilidad para detección de fallas de cualquier objeto fabricado dentro de una línea de producción, combinado con algunos otros métodos del campo de procesamiento de imágenes. Hasta ahora no existen muchos artículos publicados sobre este tema, debido a la alta complejidad computacional que se requiere para llevar a cabo estas aplicaciones usando redes neuronales. [2.42]

El reconocimiento de firmas también ha sido un tema de intensa investigación durante los últimos años, en el cual detección de firmas falsificadas es mucho mas importante que el reconocimiento mismo de esta. Hasta ahora no existe ningún método que sea suficientemente robusto para este fin desde el punto de vista de su funcionamiento.

### **2.3.3.2 Problema de Optimización**

Mayoría de los problemas óptimos, tales como el problema del agente viajero, el problema de la mochila, etc. pertenecen al problema de NP-Completo, [2.62], siendo necesario, para resolver estos problemas, realizar una búsqueda exhaustiva. Cuando una red neuronal dinámica, como la red neuronal de Hopfield, tenga un valor de la función de energía que sea mínimo, se puede decir la red convergió, y que el valor de la salida de la red es el valor de la solución óptima del problema. Se han reportado resultados exitosos acerca de la realización del problema del agente viajero usando la red neuronal de Hopfield. [2.52]

# CAPITULO 3

## Algoritmos de Adaptación

Los algoritmos de adaptación más comúnmente usados en filtrado adaptable son el “Least Mean Square” (LMS) y el algoritmo “Recursive Least Square” (RLS). Ambos tienen características de convergencia y complejidades computacionales diferentes, que los hacen atractivos cada uno, para aplicaciones diferentes. Así mientras el algoritmo LMS debido a su baja complejidad, alta sensibilidad al ruido aditivo y baja velocidad de convergencia es atractivo para aplicaciones de tiempo real que requieren filtros de alto orden operando en ambientes estacionarios o lentamente variantes con el tiempo y altas relaciones señal a ruido. Por su parte el algoritmo RLS presenta alta complejidad computacional, sin embargo su velocidad de convergencia es alto, mientras que su sensibilidad al ruido aditivo es baja, lo cual lo hace atractivo en aplicaciones en tiempo real donde el filtro debe operar en ambientes no estacionarios con relaciones señal a ruido bajas, siendo el orden de los filtros requeridos también bajos. En este capítulo se presenta una revisión de estos dos algoritmos haciendo énfasis en las ventajas que ambos ofrecen.

Finalmente se presenta también una revisión del algoritmo de retropropagación, ampliamente usado para entrenar redes neuronales multicapas, tales como el perceptron multicapas o la red TDNN.

### 3.1 Introducción

Los filtros adaptables mas comúnmente usados son los filtros transversales de respuesta a impulso finita cuya configuración básica se muestra en la figura 3.1, y su señal de salida se calcula en la siguiente ecuación:

$$y(n) = \mathbf{W}^T \mathbf{X}(n) \quad (3.1)$$

donde

$$\mathbf{W} = [w_0, w_1, \dots, w_{n-1}]^T \quad (3.2)$$

es el vector de coeficientes y

$$\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (3.3)$$

es el vector de entrada.

Como se mencionó en el capítulo 2 los coeficientes del filtro serán modificados de forma tal que alguna función de la señal de error, usualmente el valor cuadrático medio, sea minimizado. Este capítulo presenta algoritmos de adaptación para estructuras de filtros adaptables transversales, los cuales introducen ciertas simplificaciones que permiten llevarlos a la práctica.

Iniciaremos la presentación de estos algoritmos con el algoritmo LMS ( Least Mean Square ) ya que este es, debido a su baja complejidad computacional y robustez el mas ampliamente usado.

### 3.2 Algoritmo de LMS

Entre los algoritmos mas comúnmente usados tenemos aquellos basados en la búsqueda del gradiente, de los cuales el algoritmo LMS es su mejor exponente [3.1]-[3.3]. Lo anterior se debe principalmente a su baja complejidad computacional y robustez.

Un algoritmo basado en la búsqueda del gradiente descendente esta dado por

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu \nabla \quad (3.4)$$

donde  $\mathbf{W}(n)$  es el vector de coeficientes dado por (3.2) y

$$\nabla = E \left[ \frac{\partial e^2(n)}{\partial w_0}, \frac{\partial e^2(n)}{\partial w_1}, \dots, \frac{\partial e^2(n)}{\partial w_{n-1}} \right]^T \quad (3.5)$$

Es el gradiente de la superficie de error cuadrático medio y  $\mu$  es un factor que controla la estabilidad y velocidad de adaptación del algoritmo.

De la figura 3.1 tenemos que

$$e(n) = d(n) - \mathbf{W}^T \mathbf{X}(n) \quad (3.6)$$

de manera que

$$\nabla = -2E[e(n)x(n), e(n)x(n-1), \dots, e(n)x(n-N+1)]^T \quad (3.7)$$

El cálculo del gradiente no es tan simple en la práctica debido a la presencia del operador de expectativa. Esta dificultad radica principalmente en el hecho de que la superficie de error, o en su defecto la función de error, es desconocida de antemano en la práctica y debe ser estimada a partir de los datos de entrada. Una solución a este problema que fue propuesta por Widrow [3.3], consiste en reemplazar el gradiente dado por (3.5), por el gradiente instantáneo dado por:

$$\hat{\nabla} = 2 \left[ \frac{\partial e^2(n)}{\partial w_0}, \frac{\partial e^2(n)}{\partial w_1}, \dots, \frac{\partial e^2(n)}{\partial w_{n-1}} \right]^T \quad (3.8)$$

el cual de (3.6) se puede reescribir como:

$$\hat{\nabla} = -2[e(n)x(n), e(n)x(n-1), \dots, e(n)x(n-N+1)]^T \quad (3.9)$$

$$\hat{\nabla} = -2e(n)\mathbf{X}(n) \quad (3.10)$$

Finalmente substituyendo (3.10) en (3.4) obtenemos:

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu e(n)\mathbf{X}(n) \quad (3.11)$$

La ecuación (3.11) se conoce como el algoritmo LMS o algoritmo de Widrow-Hopf. Este algoritmo puede fácilmente llevarse a la práctica ya que todas las cantidades en el involucradas se encuentran disponibles.

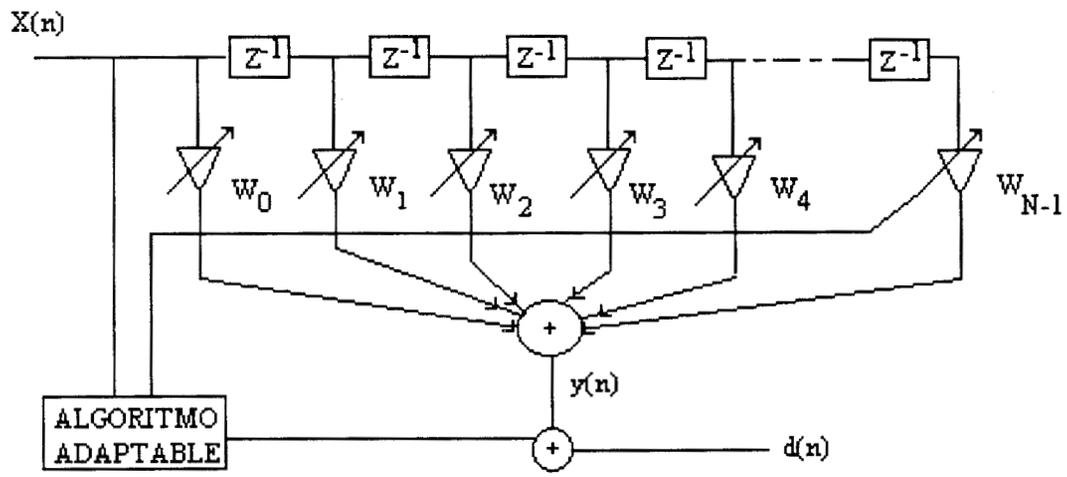


Figura 3.1. Filtro adaptable de respuesta a impulso finita (FIR-ADF)

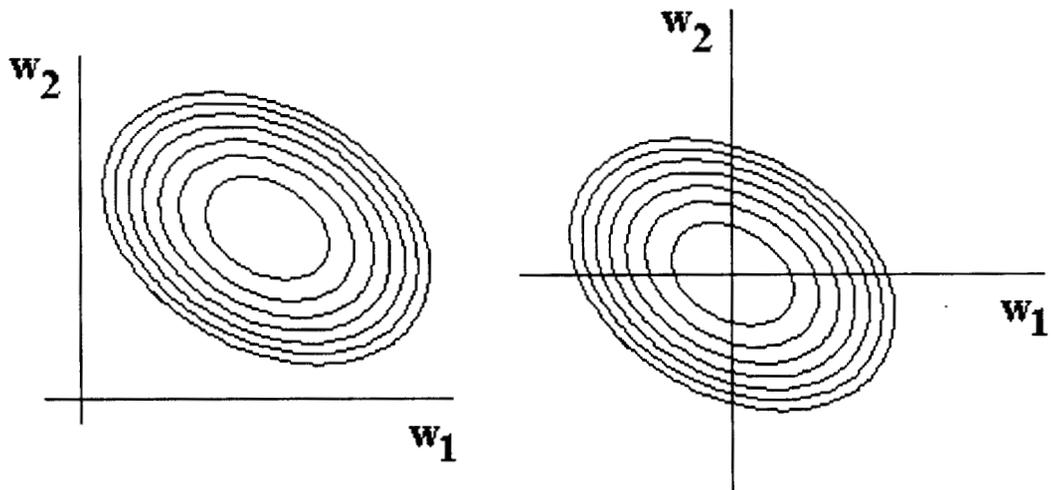


Figura 3.2. Proceso de traslación de coordenadas usado durante la prueba de convergencia del algoritmo LMS

### 3.2.1 Complejidad computacional y Característica de convergencia

El algoritmo LMS, cuando es usado para adaptar los coeficientes de un filtro transversal, requiere  $N$  multiplicaciones para calcular la salida del filtro, y  $N+1$  multiplicaciones para calcular  $2\mu e(n)\mathbf{X}(n)$  ya que el factor  $2\mu$  permanece constante durante toda la operación del sistema y  $2\mu e(n)$  es común para todos los coeficientes del filtro. Así la complejidad del filtro, medido en términos del número de multiplicaciones es

$$2N + 1 \text{ mult. / periodo de muestreo} \quad (3.12)$$

Considere la señal de error la cual esta dada por (figura 3.1)

$$e(n) = d(n) - \mathbf{X}^T(n)\mathbf{W}(n-1) \quad (3.13)$$

donde  $\mathbf{X}(n)$  y  $\mathbf{W}$  están dados por (3.2) y (3.3) respectivamente. Sustituyendo (3.13) de la expresión correspondiente al algoritmo LMS

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu e(n)\mathbf{X}(n) \quad (3.14)$$

De (3.13) y (3.14) obtenemos

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu\mathbf{X}(n)[d(n) - \mathbf{X}^T(n)\mathbf{W}(n-1)] \quad (3.15)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu d(n)\mathbf{X}(n) - 2\mu\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1) \quad (3.16)$$

tomando el valor esperado de (3.16) obtenemos:

$$E[\mathbf{W}(n)] = E[\mathbf{W}(n-1)] + 2\mu E[d(n)\mathbf{X}(n)] - 2\mu E[\mathbf{X}(n)\mathbf{X}^T(n)]E[\mathbf{W}(n-1)] \quad (3.17)$$

Aquí nosotros hemos asumido que los coeficientes del filtro no están correlacionados con la señal de entrada. Esto, aunque estrictamente hablando no se satisface completamente pues los coeficientes dependen de  $\mathbf{X}(n)$  a través de (3.11), se puede asumir para valores pequeños de  $\mu$  debido a la lenta trayectoria que de esto resulta.

Por otro lado la matriz de autocorrelación y el vector de correlación están dado por

$$\mathbf{R} = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(1) & \dots\dots\dots & \gamma_{xx}(L-1) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \dots\dots\dots & \gamma_{xx}(L-2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{xx}(L-1) & \gamma_{xx}(L-2) & \dots\dots\dots & \gamma_{xx}(0) \end{bmatrix} = E[\mathbf{X}(n)\mathbf{X}^T(n)] \quad (3.18)$$

$$\mathbf{P} = E[d(n)\mathbf{X}(n)] = E[\mathbf{X}(n)d(n)] \quad (3.19)$$

Así sustrayendo la solución óptima en ambos lados de (3.17), a partir de (3.18) y (3.19) obtenemos

$$E[\mathbf{W}(n)] - \mathbf{R}^{-1}\mathbf{P} = E[\mathbf{W}(n-1)] - \mathbf{R}^{-1}\mathbf{P} + 2\mu\mathbf{R}[\mathbf{R}^{-1}\mathbf{P} - E[\mathbf{W}(n-1)]] \quad (3.20)$$

seguidamente definiendo

$$\xi(n) = E[\mathbf{W}(n)] - \mathbf{R}^{-1}\mathbf{P} \quad (3.21)$$

de (3.20) obtenemos

$$\xi(n) = (\mathbf{I} - 2\mu\mathbf{R})\xi(n-1) \quad (3.22)$$

Este proceso equivale a una translación de coordenadas se muestra figura 3.2.

Seguidamente, representamos  $\mathbf{R}$  en términos de una transformación ortogonal [3.3],

$$\mathbf{R} = \mathbf{K}^T \mathbf{Q} \mathbf{K} \quad (3.23)$$

donde

$$\mathbf{Q} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_{N-1}] \quad (3.24)$$

y  $\lambda_i$  es el  $i$ -ésimo valor propio de la matriz  $\mathbf{R}$ . Haciendo lo anterior y substituyendo (3.23) en (3.22) obtenemos:

$$\xi(n) = (\mathbf{I} - 2\mu\mathbf{K}^T \mathbf{Q} \mathbf{K})\xi(n-1) \quad (3.25)$$

$$\xi(n) = \mathbf{K}^T (\mathbf{I} - 2\mu\mathbf{Q}) \mathbf{K} \xi(n-1) \quad (3.26)$$

multiplicando en ambos lados de (3.26) por la izquierda por  $\mathbf{K}$  y definiendo:

$$\mathbf{V}(n) = \mathbf{K} \xi(n) \quad (3.27)$$

obtenemos

$$\mathbf{V}(n) = (\mathbf{I} - 2\mu\mathbf{Q})\mathbf{V}(n-1) \quad (3.28)$$

Finalmente iterando (3.28) y reescribiendo el resultado en forma matricial obtenemos

$$\begin{bmatrix} v_0(n) \\ v_1(n) \\ \vdots \\ v_{N-1}(n) \end{bmatrix} = \begin{bmatrix} (1-2\mu\lambda_1)^n & & & \\ & (1-2\mu\lambda_2)^n & & \\ & & \ddots & \\ & & & (1-2\mu\lambda_N)^n \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix} \quad (3.29)$$

Así usando el algoritmo LMS, el valor esperado del vector de coeficientes convergerá a la solución óptima sí,

$$|1 - 2\mu\lambda_i| < 1, \quad i = 1, 2, \dots, N-1 \quad (3.30)$$

esto es, el algoritmo convergerá siempre y cuando

$$0 < \mu < \frac{1}{\lambda_{\text{máx}}} \quad (3.31)$$

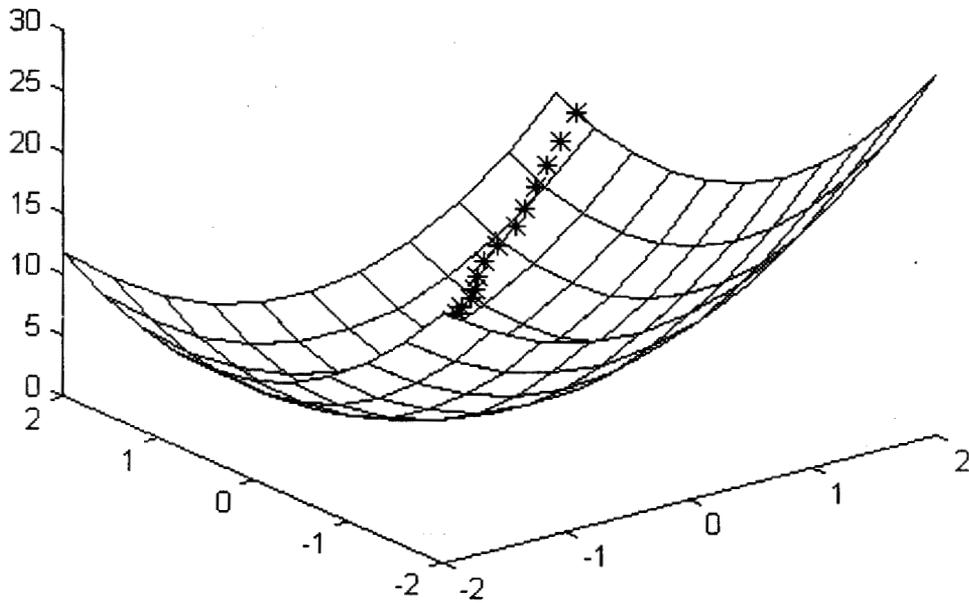
La estimación de los valores propios es un problema complejo que requiere de un número elevado de operaciones. Así, con el fin de proporcionar un rango de validez de aplicación práctica consideremos un resultado de teoría de matrices que establece que:

$$\sum_{i=0}^{N-1} E[x^2(n)] = N \overline{x^2(n)} > \lambda_{\text{máx}} \quad (3.32)$$

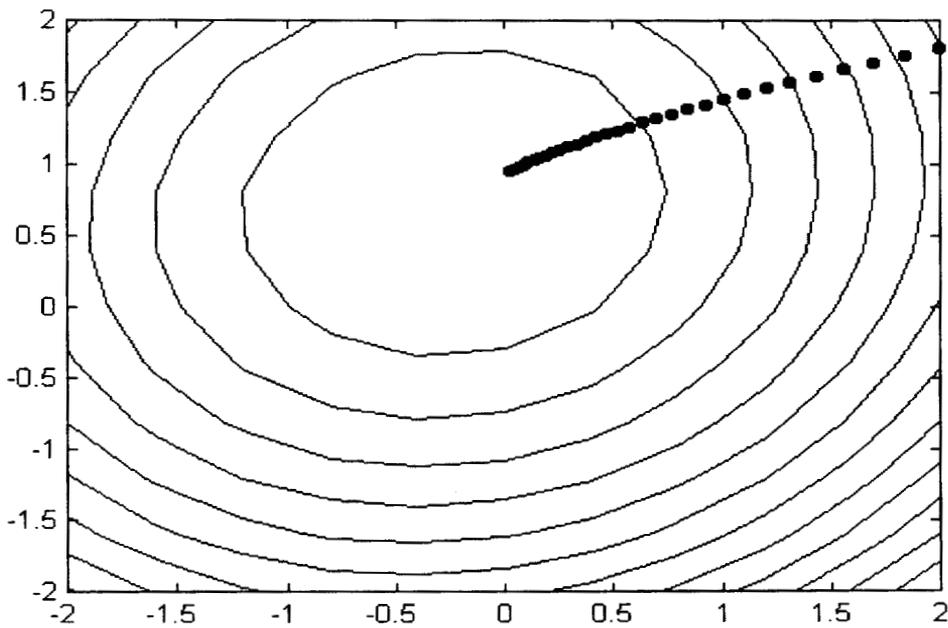
Entonces, usando (3.32) podemos obtener un rango de validez para  $\mu$  el cual puede ser fácilmente estimado. Este estará dado por [3.3]

$$0 < \mu < \frac{1}{N \overline{x^2(n)}} \quad (3.33)$$

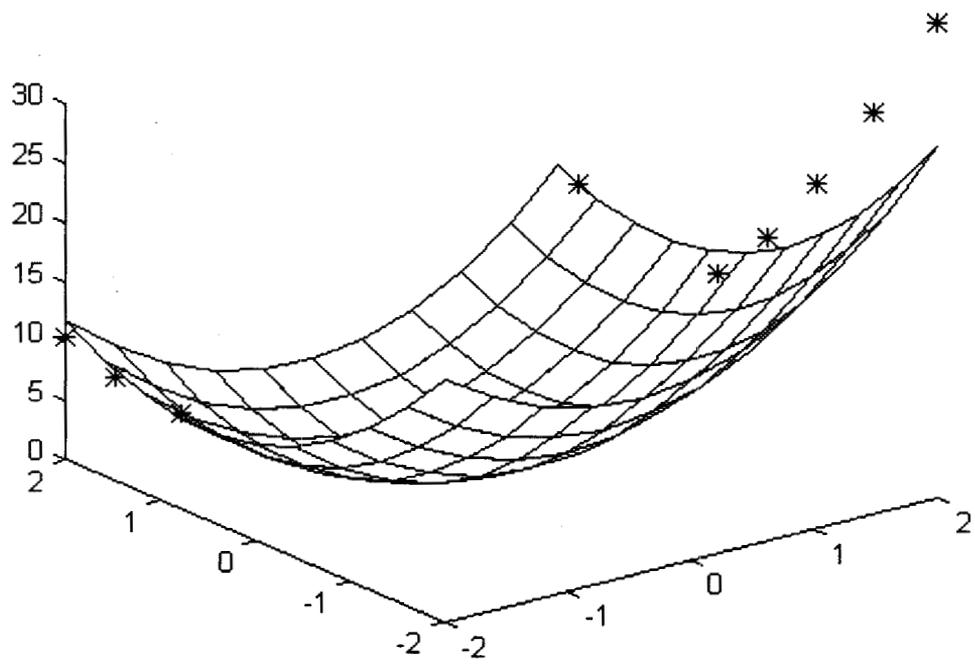
Donde  $\overline{x^2(n)}$  denota el valor cuadrático medio de  $x(n)$  estimado con un número reducido de muestras.



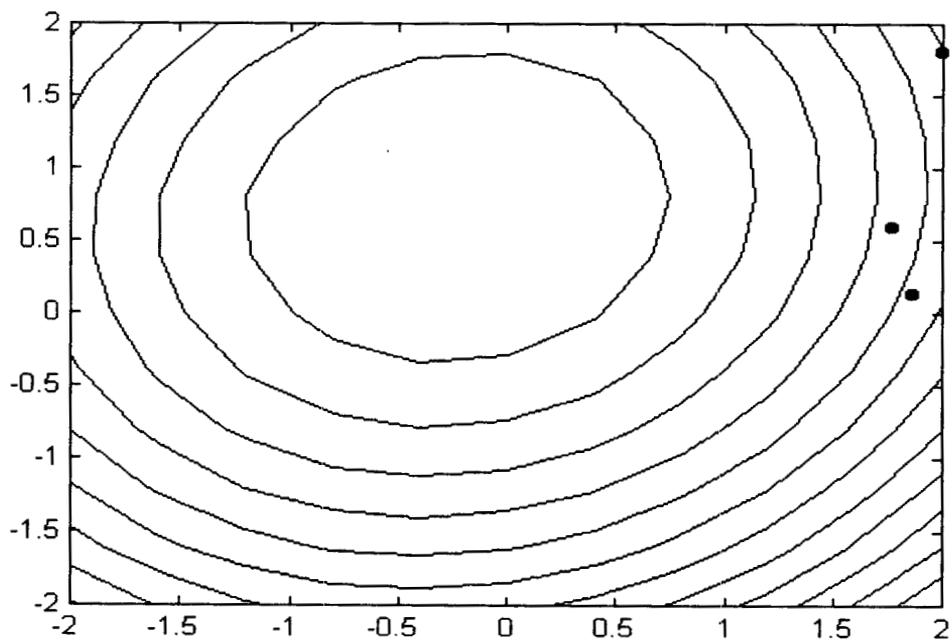
**Figura 3.3.** Superficie de error y trayectoria de los coeficientes del filtro en la dirección del gradiente desde el punto inicial hasta arribar al valor óptimo.



**Figura 3.4.** Trayectoria del vector de coeficientes del filtro adaptable LMS en la dirección del gradiente negativo desde un punto inicial arbitrario hasta el punto óptimo, con un factor de convergencia  $\mu=0.03$ .



**Figura 3.5.** Superficie de error y trayectoria de los coeficientes del filtro en la dirección del gradiente desde el punto inicial. Aquí debido a que el factor de convergencia es muy grande,  $\mu=0.82$  el algoritmo LMS no converge.



**Figura 3.6.** Trayectoria de los coeficientes de un filtro adaptable LMS usando un factor de convergencia mayor que el máximo permitido, lo que ocasiona que el algoritmo diverja.

La figura 3.3 muestra superficie de error y una trayectoria de búsqueda de gradiente de error usando algoritmo LMS, en el cual sistema tiene dos coeficientes  $w_0, w_1$ , los cuales son inicializados con números aleatorios. La figura 3.4 es contorno de superficie de error equivalente a la figura 3.3 y trayectoria de búsqueda. Aquí el algoritmo LMS termina cuando error cuadrático medio sea menor que cierta valor  $\epsilon$ , el factor de convergencia  $\mu$  es igual a 0.03, el cual cumple la condición dada por (3.33).

Las figuras 3.5 y 3.6 muestran trayectoria de búsqueda con valor de  $\mu$  es igual a 0.82 usando mismo sistema. El hecho de que el valor  $\mu$  no cumpla la condición de (3.33) ocasiona que el algoritmo no converja.

### 3.2.2 Propiedades del Gradiente Instantáneo

En esta sección analizaremos algunas propiedades del gradiente instantáneo y compararemos su comportamiento con aquel del gradiente real. Debido al hecho de que el gradiente instantáneo es una variable aleatoria, la comparación se llevara a cabo en términos estadísticos. Así, considere inicialmente la ecuación (3.16)

$$\mathbf{W}(n) = \mathbf{W}(n-1) - 2\mu\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1) + 2\mu d(n)\mathbf{X}(n) \quad (3.34)$$

la cual se puede reescribir como

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu(2\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1) - 2d(n)\mathbf{X}(n)) \quad (3.35)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu\hat{\nabla} \quad (3.36)$$

donde

$$\hat{\nabla} = 2\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1) - 2d(n)\mathbf{X}(n) \quad (3.37)$$

tomando el valor esperado (3.37) obtenemos [3.8]

$$E[\hat{\nabla}] = 2E[\mathbf{X}(n)\mathbf{X}^T(n)]E[\mathbf{W}(n-1)] - 2E[d(n)\mathbf{X}(n)] \quad (3.38)$$

$$E[\hat{\nabla}] = 2\mathbf{R}E[\mathbf{W}(n-1)] - 2\mathbf{P} \quad (3.39)$$

Así, debido a que cuando el algoritmo converge a la solución optima [3.8]

$$E[\mathbf{W}(n)] = \mathbf{W}^* \quad (3.40)$$

y

$$\nabla = 2\mathbf{R}\mathbf{W}^* - 2\mathbf{P} \quad (3.41)$$

de (3.40) y (3.41) obtenemos que

$$E[\hat{\nabla}] = \nabla \quad (3.42)$$

seguidamente considere el  $\nabla \bullet \nabla$  el cual puede reescribirse como [3.8]

$$\nabla \bullet \nabla = E[\nabla^T \nabla] \quad (3.43)$$

$$\nabla \bullet \nabla = \nabla^T \nabla \quad (3.44)$$

A continuación considere el  $\hat{\nabla} \bullet \nabla$ , el cual a partir de (3.42) esta dado por

$$\hat{\nabla} \bullet \nabla = E[\hat{\nabla}^T \nabla] \quad (3.45)$$

$$\hat{\nabla} \bullet \nabla = E[\hat{\nabla}^T] \nabla \quad (3.46)$$

$$\hat{\nabla} \bullet \nabla = \nabla^T \nabla \quad (3.47)$$

de donde

$$\hat{\nabla} \bullet \nabla = \nabla \bullet \nabla \quad (3.48)$$

Esto significa que la proyección ortogonal del gradiente instantáneo en  $\hat{\nabla}$  es igual a  $\nabla$ . En el sentido estadístico, esto significa que el gradiente instantáneo siempre contiene una componente igual al verdadero gradiente, pero su magnitud es, en general mayor que la del verdadero gradiente y su dirección se desvía de la del verdadero gradiente en un ángulo  $\theta$ , como se ilustra en la figura 3.7.

Los resultados anteriores muestra además que (3.42),

$$E[\theta] = 0 \quad (3.49)$$

### 3.3 Algoritmo LMS Normalizado

Considere el algoritmo LMS el cual esta dado por (3.11)

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{X}(n) \quad (3.50)$$

donde  $\mu$  es un factor que controla la estabilidad y velocidad de convergencia.

Considere la diferencia entre el vector optimo  $\mathbf{W}^*$  y el vector de coeficientes actual  $\mathbf{W}(n)$  dada por:

$$\mathbf{V}(n) = \mathbf{W}^* - \mathbf{W}(n) \quad (3.51)$$

Entonces asumiendo que la señal deseada esta dada por

$$d(n) = \mathbf{W}^{*T} \mathbf{X}(n) \quad (3.52)$$

Seguidamente suponga que la señal de error esta dada por

$$e(n) = d(n) - \mathbf{W}^T(n)\mathbf{X}(n) \quad (3.53)$$

Sustituyendo (3.52) en (3.53) a partir de (3.51) obtenemos

$$e(n) = \mathbf{W}^{*T} \mathbf{X}(n) - \mathbf{W}^T(n)\mathbf{X}(n) \quad (3.54)$$

$$e(n) = (\mathbf{W}^{*T} - \mathbf{W}^T(n))\mathbf{X}(n) \quad (3.55)$$

$$e(n) = \mathbf{V}^T(n)\mathbf{X}(n) \quad (3.56)$$

seguidamente hacemos una proyección del vector de diferencia  $\mathbf{V}(n)$  en el vector  $\mathbf{X}(n)$  como se muestra en la figura 3.8.

Haciendo esto descomponemos  $\mathbf{V}(n)$  como [3.7]

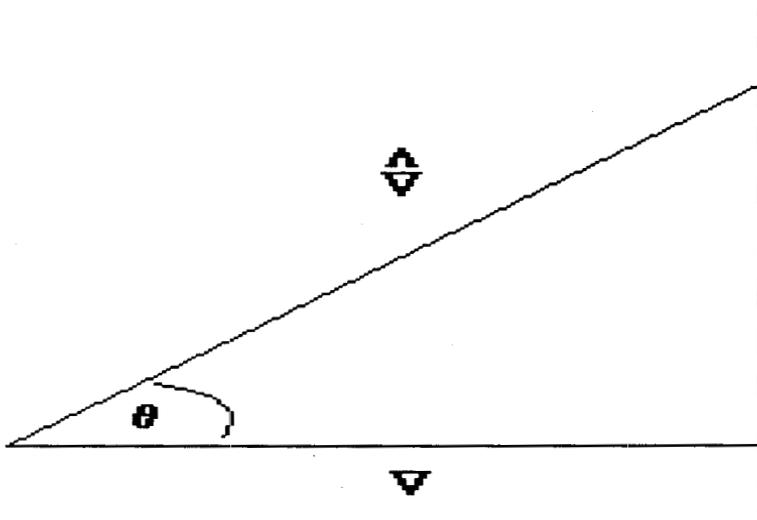
$$\mathbf{V}(n) = \mathbf{V}_o(n) + \mathbf{V}_p(n) \quad (3.57)$$

donde  $\mathbf{V}_o(n)$  es la componente de  $\mathbf{V}(n)$  ortogonal al vector de entrada  $\mathbf{X}(n)$  y  $\mathbf{V}_p(n)$  es la componente de  $\mathbf{V}(n)$  paralela a  $\mathbf{X}(n)$ . Esto implica que

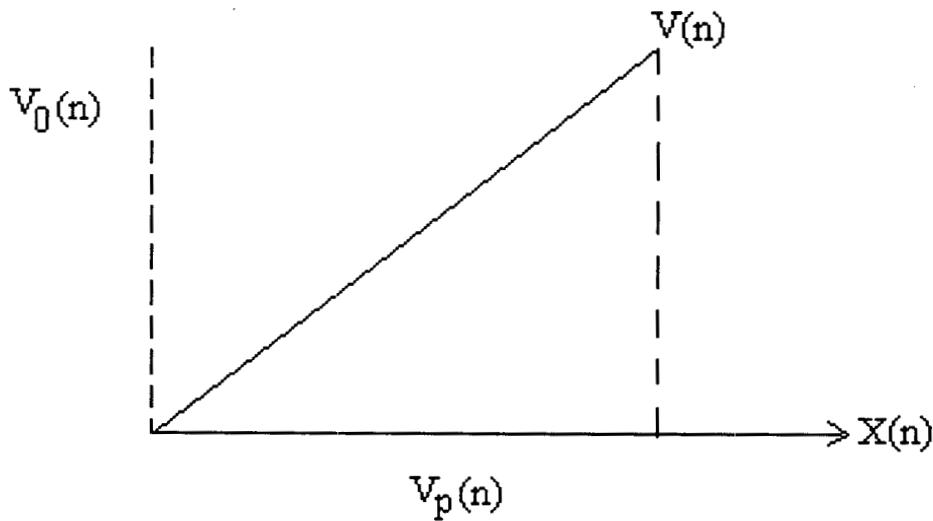
$$\mathbf{V}_p(n) = C\mathbf{X}(n) \quad (3.58)$$

donde  $C$  es un constante, y entonces sustituyendo (3.57) y (3.58) en (3.56) obtenemos

$$e(n) = [\mathbf{V}_o(n) + \mathbf{V}_p(n)]^T \mathbf{X}(n) \quad (3.59)$$



**Figura 3.7.** Ilustración de la diferencia existente entre el gradiente instantaneo y el gradiente real. El primero es siempre mayor que el segundo, aunque el gradiente instantaneo tiene siempre un componente igual al gradiente real.



**Figura 3.8.** Representación del vector de error  $V(n)$  en términos de sus componentes paralela  $V_p(n)$  y ortogonal  $V_o(n)$  al vector de entrada  $X(n)$ .

$$e(n) = [\mathbf{V}_o(n) + C\mathbf{X}(n)]^T \mathbf{X}(n) \quad (3.60)$$

Debido a que  $\mathbf{V}_o(n)$  es ortogonal a  $\mathbf{X}(n)$  es el producto escalar:

$$\mathbf{V}_o^T \mathbf{X}(n) = 0 \quad (3.61)$$

y entonces

$$e(n) = C\mathbf{X}^T(n)\mathbf{X}(n) \quad (3.62)$$

de donde despejando  $C$  obtenemos

$$C = \frac{e(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.63)$$

y así, (3.63) en (3.58) obtenemos

$$\mathbf{V}_p(n) = \frac{e(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.64)$$

El objetivo a continuación es hacer el vector  $\mathbf{V}(n)$  más ortogonal a  $\mathbf{X}(n)$ , en cada iteración, en forma tal que  $\mathbf{V}(n)$  sea más ortogonal a  $\mathbf{X}(n)$  de lo que lo fue el vector  $\mathbf{V}(n-1)$ , como se muestra figura 3.9 [3.7].

Lo anterior se logra sustrayendo el vector  $\alpha\mathbf{V}_p(n)$  del vector  $\mathbf{V}(n)$  :

$$\mathbf{V}(n+1) = \mathbf{V}(n) - \alpha\mathbf{V}_p(n) \quad (3.65)$$

Finalmente, sustituyendo (3.51)

$$\mathbf{W}^* - \mathbf{W}(n+1) = \mathbf{W}^* - \mathbf{W}(n) - \alpha \frac{e(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.66)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \alpha \frac{e(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.67)$$

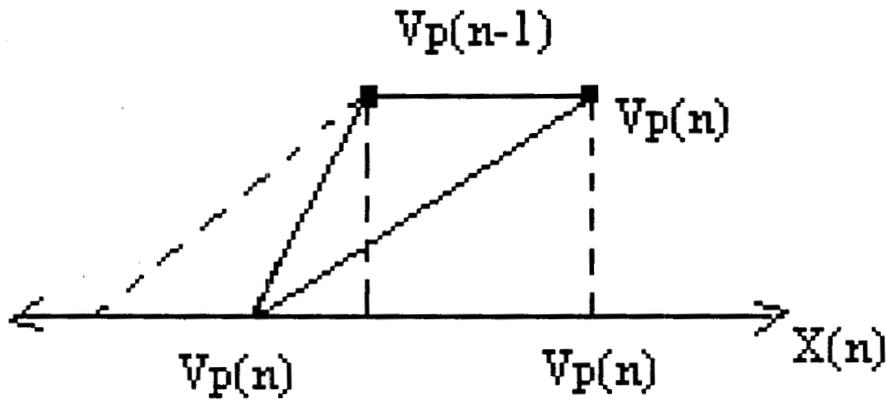


Figura 3.9. Interpretación geométrica del algoritmo LMS normalizado.

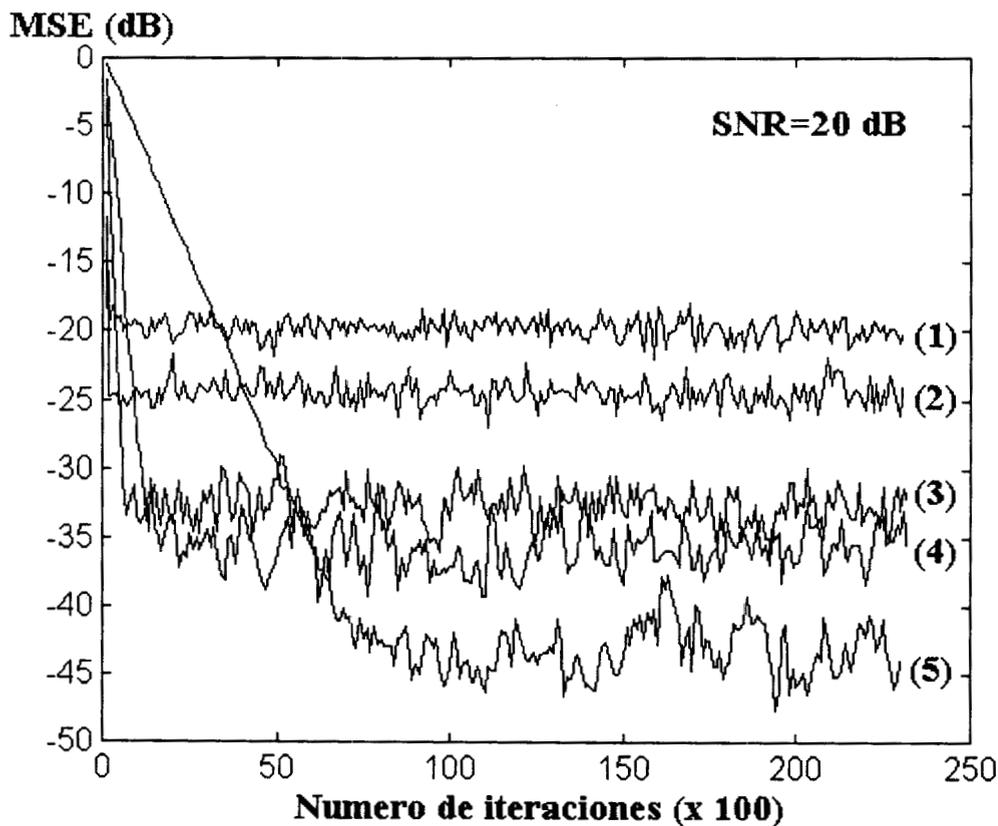


Figura 3.10. Características de convergencia del algoritmo LMS normalizado con factores de convergencia (1)  $\alpha=1.0$ , (2)  $\alpha=0.5$ , (3)  $\alpha=0.1$ , (4)  $\alpha=0.05$ , (5)  $\alpha=0.01$ . La señal de entrada es una secuencia de ruido blanco y la señal de referencia es estacionaria.

donde, con el fin de que  $\mathbf{V}(n)$  en cada iteración se vuelva más ortogonal a  $\mathbf{X}(n)$ ,  $\alpha$  debe satisfacer la condición [3.7]

$$0 < \alpha < 2 \quad (3.68)$$

Así entonces

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \beta e(n)\mathbf{X}(n) \quad (3.69)$$

donde :

$$\beta = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.70)$$

Así el algoritmo NLMS (3.69) es equivalente al algoritmo LMS (3.11) si

$$2\mu = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (3.71)$$

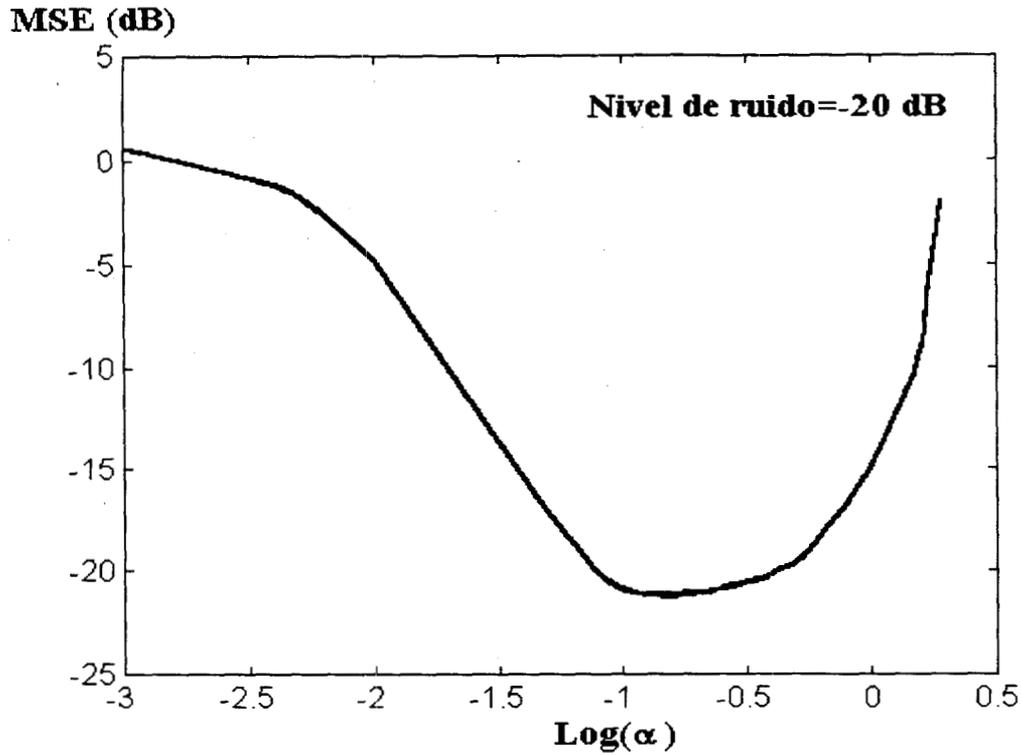
### 3.3.1 Propiedad de Convergencia con Señales Estacionarias y No Estacionarias

Hasta hace poco la mayoría de sistemas de comunicación, como por ejemplo, los sistemas telefónicos, se podían considerar estacionarios o muy lentamente variable con el tiempo. Sin embargo con el avance de la tecnología han hecho su aparición de sistemas de comunicación móvil en los cuales los canales ya no pueden considerarse estacionarios, sino por el contrario, estos varían en muchas ocasiones rápidamente con el tiempo[3.4].

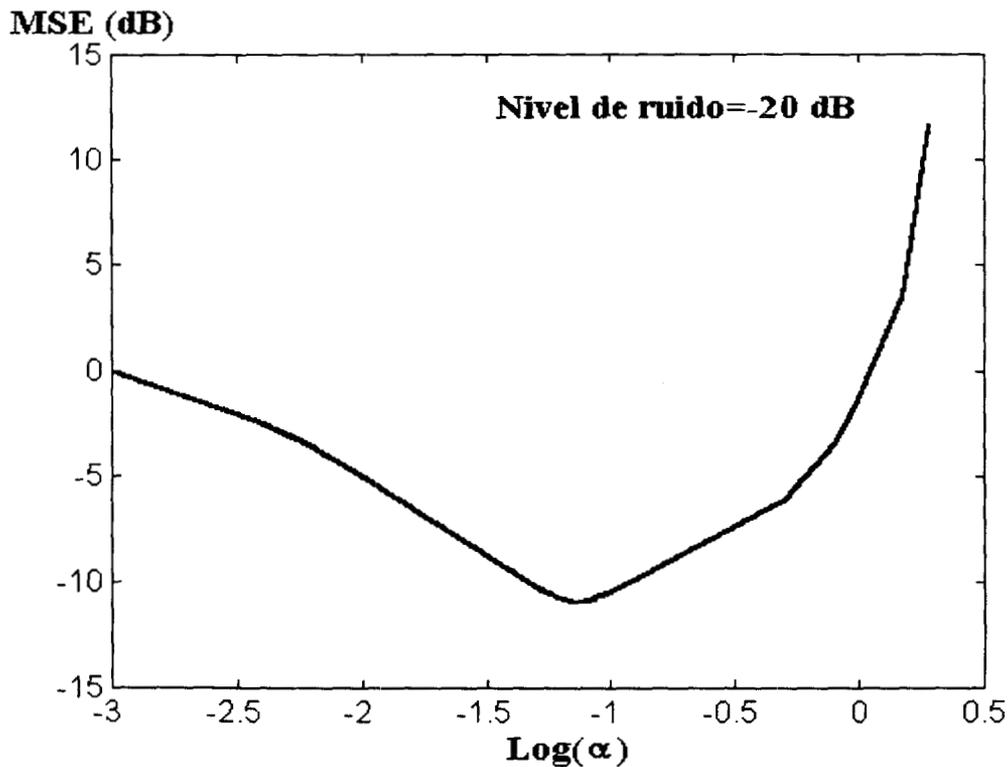
La propiedad de convergencia del algoritmo LMS y NLMS es bien conocida cuando la señal de entrada sea estacionaria o muy lentamente variable con el tiempo. La figura 3.10 muestra una comparación entre error cuadrático medio y velocidad de convergencia con diferentes factores de convergencia  $\alpha$ , cuando es requerido a identificar un sistema invariante con el tiempo. En la figura 3.10, (1)  $\alpha=1.0$ , (2)  $\alpha=0.5$ , (3)  $\alpha=0.1$ , (4)  $\alpha=0.05$  y (5)  $\alpha=0.01$ . De esta figura se tiene que con pequeños valores de  $\alpha$  el sistema se tarda converger, sin embargo error cuadrático medio es menor. La señal de entrada es ruido blanco.

Las figuras 3.11 y 3.12 muestran el funcionamiento del algoritmo tomando factores de convergencia  $\alpha$ , cuando es requerido a identificar sistemas variables con el tiempo, el cual permanece constante durante 100 muestras después de las cuales se modifica de acuerdo a la relación

$$\begin{aligned} A[i, k] &= A[i, 0]\cos(2\pi k / 120) && k \text{ impar} \\ A[i, k] &= A[i, 0]\cos(2\pi k / 200) && k \text{ par} \end{aligned} \quad (3.72)$$



**Figura 3.11.** Error cuadrático medio obtenido cuando el algoritmo LMS normalizado, con diferentes factores de convergencia  $\alpha$  es requerido a identificar un sistema variable con el tiempo. La señal de entrada es ruido blanco.



**Figura 3.12.** Error cuadrático medio obtenido cuando el algoritmo LMS normalizado, con diferentes factores de convergencia  $\alpha$  es requerido a identificar un sistema variable con el tiempo. La señal de entrada es un proceso autoregresivo con función de autocorrelación igual a la de una señal de voz actual.

Donde  $A$  es el vector de coeficientes,  $i$  es el índice del vector de coeficientes y  $k$  es el número de bloque. Esta técnica de simulación de sistemas variables con el tiempo se le conoce como método de "Snapshot".[3.4]

En la figura 3.11, la señal de entrada fue ruido blanco, mientras que en la figura 3.12, la señal de entrada el proceso AR. De las figuras se observa que cuando el algoritmo opera, en sistemas no estacionarios, no se puede afirmar que valores pequeños de  $\alpha$  lleven a un mejor funcionamiento.

### 3.4 Algoritmo de RLS (Recursive Least Square)

Otro de los algoritmos de adaptación comúnmente usados en filtrado adaptable es el algoritmo RLS (Recursive Least Square) [3.1],[3.2] en el cual se minimiza directamente el valor cuadrático medio de la señal de error por medio de un inversión de matriz. La derivación de este algoritmo, se presenta a continuación para lo cual considere la salida de un filtro de respuesta a impulso finita, la cual está dada por

$$y(n) = \mathbf{W}^T \mathbf{X}(n) = \mathbf{X}^T(n) \mathbf{W} \quad (3.73)$$

donde

$$\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (3.74)$$

$$\mathbf{W} = [w_0, w_1, w_2, \dots, w_{N-1}]^T \quad (3.75)$$

El vector de coeficientes óptimo se calculará de manera tal que el error cuadrático medio,  $E[e^2(n)]$  sea minimizado donde

$$e(n) = d(n) - y(n) \quad (3.76)$$

$$E[e^2(n)] = E[(d(n) - y(n))^2] \quad (3.77)$$

Para minimizar (3.77) podemos usar el principio de ortogonalidad en estimación de mínimos cuadrados. Esto es, seleccionamos el vector de coeficientes en forma tal que el error de salida llegue a ser ortogonal al vector de entrada. Así de (3.73) y (3.76), obtenemos [3.2]

$$E[\mathbf{X}(n)(d(n) - \mathbf{X}(n)^T \mathbf{W})] = 0 \quad (3.78)$$

de donde

$$E[\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}] = E[d(n)\mathbf{X}(n)] \quad (3.79)$$

Assumiendo que el vector de coeficientes es no correlacionado con el vector de entrada, obtenemos

$$E[\mathbf{X}(n)\mathbf{X}^T(n)]\mathbf{W} = E[d(n)\mathbf{X}(n)] \quad (3.80)$$

La cual se puede reescribir como

$$\mathbf{R}\mathbf{W} = \mathbf{P} \quad (3.81)$$

donde

$$\mathbf{P} = E[d(n)\mathbf{X}(n)] \quad (3.82)$$

es el vector de correlación entre la señal de referencia  $d(n)$  y el vector de entrada  $\mathbf{X}(n)$  y

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)] \quad (3.83)$$

es la matriz de autocorrelación de la señal de entrada. Seguidamente, asumiendo ergodicidad,  $\mathbf{P}(n)$ , es puede estimar en tiempo real de la siguiente manera

$$\mathbf{P}(n) = \sum_{k=0}^n \lambda^{n-k} d(k)\mathbf{X}(k) \quad (3.84)$$

$$\mathbf{P}(n) = \sum_{k=0}^{n-1} \lambda^{n-k} d(k)\mathbf{X}(k) + d(n)\mathbf{X}(n) \quad (3.85)$$

$$\mathbf{P}(n) = \lambda \sum_{k=0}^{n-1} \lambda^{n-k-1} d(k)\mathbf{X}(k) + d(n)\mathbf{X}(n) \quad (3.86)$$

$$\mathbf{P}(n) = \lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n) \quad (3.87)$$

Donde  $\lambda$  se conoce como factor de olvido [3.2].

En forma similar obtenemos

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{X}(n)\mathbf{X}^T(n) \quad (3.88)$$

Así, multiplicando (3.81) por la izquierda por  $\mathbf{R}^{-1}$  y sustituyendo en éste resultado (3.87) y (3.88) obtenemos

$$\mathbf{W} = [\lambda\mathbf{R}(n-1) + \mathbf{X}(n)\mathbf{X}^T(n)]^{-1}[\lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n)] \quad (3.89)$$

Seguidamente, usando el lema de inversión de matrices, el cual establece que [3.2]

$$[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}[\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}]^{-1}\mathbf{DA}^{-1} \quad (3.90)$$

Con  $\mathbf{A}=\lambda\mathbf{R}(n-1)$ ,  $\mathbf{B}=\mathbf{X}(n)$ ,  $\mathbf{C}=1$  y  $\mathbf{D}=\mathbf{X}^T(n)$  obtenemos

$$\begin{aligned}\mathbf{W}(n) &= \left[ \frac{1}{\lambda} \mathbf{R}^{-1}(n-1) - \left[ \frac{1}{\lambda} \mathbf{R}^{-1}(n-1) \mathbf{X}(n) \right] \right. \\ &\quad * \left[ \frac{1}{\lambda} \mathbf{X}^T(n) \mathbf{R}^{-1}(n-1) \mathbf{X}(n) + 1 \right]^{-1} \frac{1}{\lambda} \mathbf{X}^T(n) \mathbf{R}^{-1}(n-1) \left. \right] \\ &\quad * [\lambda \mathbf{P}(n-1) + d(n) \mathbf{X}(n)]\end{aligned}\quad (3.91)$$

$$\begin{aligned}\mathbf{W}(n) &= \frac{1}{\lambda} \left[ \mathbf{R}^{-1}(n-1) - \frac{\mathbf{R}^{-1}(n-1) \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{R}^{-1}(n-1)}{[\lambda + \mathbf{X}^T(n) \mathbf{R}^{-1}(n-1) \mathbf{X}(n)]} \right] \\ &\quad * [\lambda \mathbf{P}(n-1) + d(n) \mathbf{X}(n)]\end{aligned}\quad (3.92)$$

Seguidamente, definiendo

$$\mathbf{Q}(n) = \mathbf{R}^{-1}(n) \quad (3.93)$$

y

$$\mathbf{K}(n) = \frac{\mathbf{R}^{-1}(n-1) \mathbf{X}(n)}{\lambda + \mathbf{X}^T(n) \mathbf{R}^{-1}(n-1) \mathbf{X}(n)} \quad (3.94)$$

de (3.92) obtenemos

$$\mathbf{W}(n) = \frac{1}{\lambda} [\mathbf{Q}(n-1) - \mathbf{K}(n) \mathbf{X}^T(n) \mathbf{Q}(n-1)] [\lambda \mathbf{P}(n-1) + d(n) \mathbf{X}(n)] \quad (3.95)$$

$$\begin{aligned}\mathbf{W}(n) &= \mathbf{Q}(n-1) \mathbf{P}(n-1) + \frac{1}{\lambda} d(n) \mathbf{Q}(n-1) \mathbf{X}(n) \\ &\quad - \mathbf{K}(n) \mathbf{X}^T(n) \mathbf{Q}(n-1) \mathbf{P}(n-1) - \frac{1}{\lambda} d(n) \mathbf{K}(n) \mathbf{X}^T(n) \mathbf{Q}(n-1) \mathbf{X}(n)\end{aligned}\quad (3.96)$$

$$\begin{aligned}\mathbf{W}(n) &= \mathbf{W}(n-1) + \frac{1}{\lambda} d(n) \mathbf{Q}(n-1) \mathbf{X}(n) \\ &\quad - \frac{\mathbf{Q}(n-1) \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{W}(n-1)}{\lambda + \mathbf{X}^T(n) \mathbf{Q}(n-1) \mathbf{X}(n)} \\ &\quad - \frac{1}{\lambda} \frac{d(n) \mathbf{Q}(n-1) \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{Q}(n-1) \mathbf{X}(n)}{\lambda + \mathbf{X}^T(n) \mathbf{Q}(n-1) \mathbf{X}(n)}\end{aligned}\quad (3.97)$$

$$\begin{aligned} \mathbf{W}(n) = \mathbf{W}(n-1) + \frac{1}{\lambda} \frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{[\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)]} \\ * [\lambda d(n) + d(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n) \\ - \lambda \mathbf{X}^T(n)\mathbf{W}(n-1) - d(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)] \end{aligned} \quad (3.98)$$

$$\begin{aligned} \mathbf{W}(n) = \mathbf{W}(n-1) + \frac{1}{\lambda} \frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{[\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)]} \\ * \lambda [d(n) - \mathbf{X}^T(n)\mathbf{W}(n-1)] \end{aligned} \quad (3.99)$$

de donde obtenemos finalmente

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{K}(n)e(n) \quad (3.100)$$

donde

$$\mathbf{K}(n) = \frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)} \quad (3.101)$$

Una expresión alterna puede ser obtenida a partir de (3.95) de donde [3.1],[3.2]

$$\mathbf{Q}(n) = \frac{1}{\lambda} [\mathbf{Q}(n-1) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)] \quad (3.102)$$

Multiplicando en ambos lados por  $\mathbf{X}(n)$  por la derecha obtenemos

$$\mathbf{Q}(n)\mathbf{X}(n) = \frac{1}{\lambda} [\mathbf{Q}(n-1)\mathbf{X}(n) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)] \quad (3.103)$$

$$\mathbf{Q}(n)\mathbf{X}(n) = \frac{1}{\lambda} [\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)]\mathbf{K}(n) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n) \quad (3.104)$$

$$\begin{aligned} \mathbf{Q}(n)\mathbf{X}(n) = \mathbf{K}(n) + \frac{1}{\lambda} \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n) \\ - \frac{1}{\lambda} \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n) \end{aligned} \quad (3.105)$$

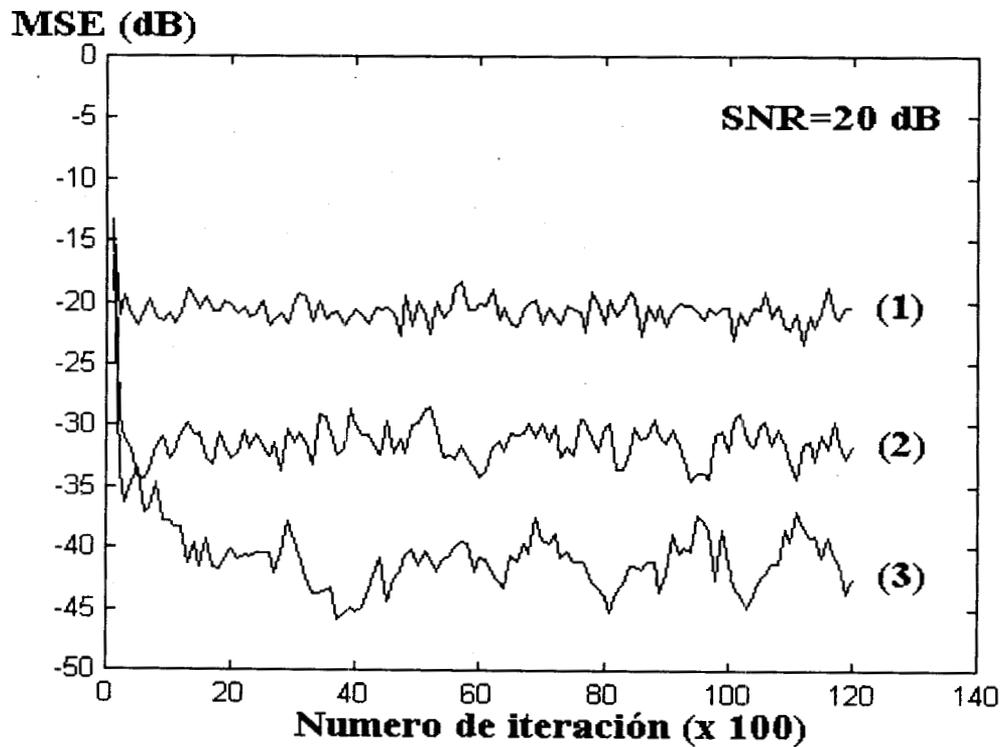


Figura 3.13. Convergencia del algoritmo RLS usando tres diferentes factores de olvido (1) 0.9, (2) 0.99 y (3) 0.999, cuando es requerido a identificar un sistema invariante con el tiempo.

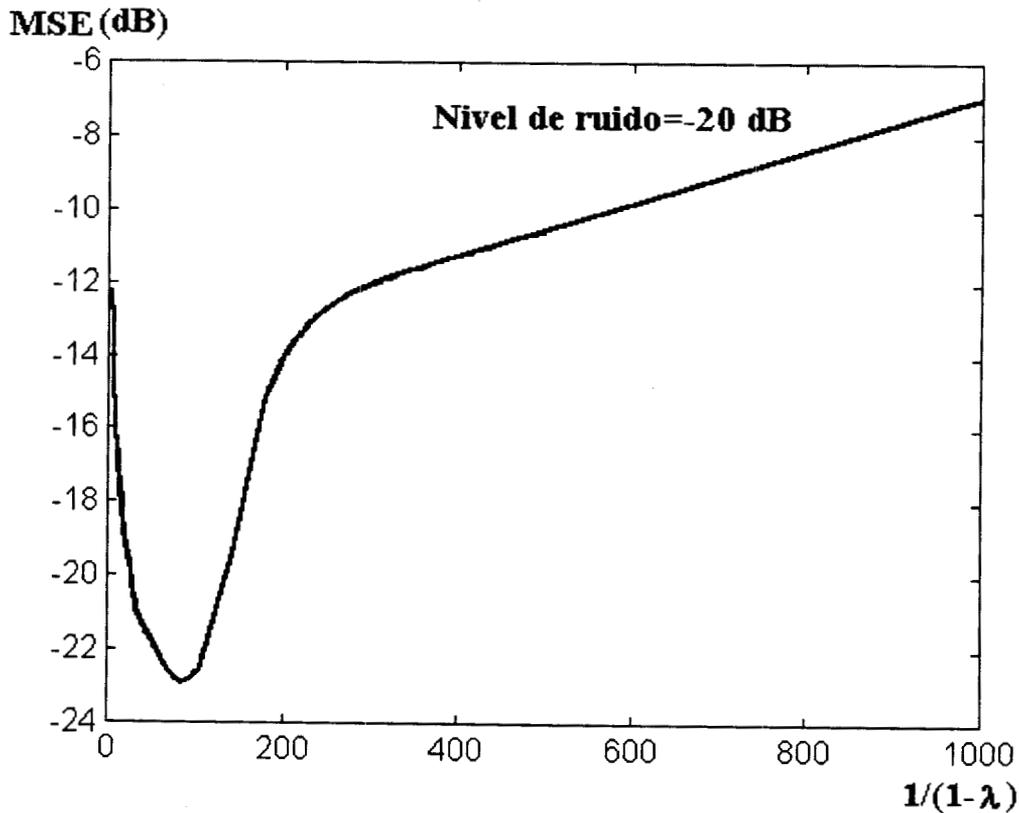
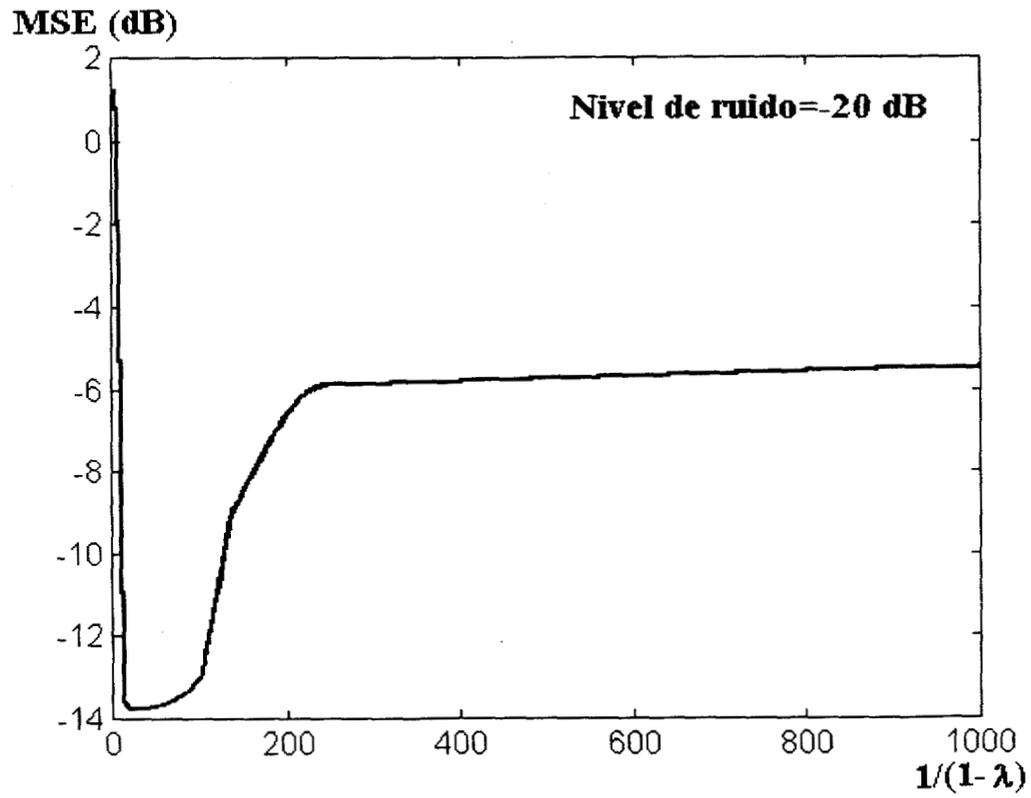


Figura 3.14. Error cuadrático medio obtenido cuando el algoritmo RLS, con diferentes factores de olvido  $\lambda$  es requerido a identificar un sistema variable con el tiempo. La señal de entrada es ruido blanco.



**Figura 3.15.** Error cuadrático medio obtenido cuando el algoritmo RLS, con diferentes factores de olvido  $\lambda$  es requerido a identificar un sistema variable con el tiempo. La señal de entrada es un proceso autogresivo cuya función de autocorrelación es idéntica a la de una señal de voz actual.

$$\mathbf{Q}(n)\mathbf{X}(n) = \mathbf{K}(n) \quad (3.106)$$

Entonces, sustituyendo (3.106) en (3.100) obtenemos

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{Q}(n)e(n)\mathbf{X}(n) \quad (3.107)$$

donde  $\mathbf{Q}(n)$  esta dado por

$$\mathbf{Q}(n) = \frac{1}{\lambda} \left[ \mathbf{Q}(n-1) - \frac{\mathbf{Q}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)} \right] \quad (3.108)$$

### 3.4.1 Propiedad de Convergencia con Señales Estacionarias y No Estacionarias

Con el fin de evaluar la propiedad de convergencia del algoritmo de RLS, cuando es requerido a identificar sistemas tanto invariantes como variantes con el tiempo.[3.4] La figura 3.13 muestra el funcionamiento del algoritmo RLS, usando distintos factores de olvido  $\lambda$  o memorias de diferente longitud, cuando es requerido a identificar un sistema invariante con el tiempo. La señal de entrada fue una secuencia de ruido blanco con una relación señal a ruido de 20dB. Como se esperaba un factor de olvido mas cercano a uno, en casos estacionarios proporciona un mejor funcionamiento.

Las figuras 3.14 y 3.15 muestran el funcionamiento cuando es requerido a identificar sistemas variantes con el tiempo. La variación empleada es la misma que se empleó con el algoritmo de LMS, y que se muestra (3.72). En la figura 3.14, la señal de entrada fue una serie de ruido blanco, mientras que en la figura 3.15, la señal de entrada fue un proceso autoregresivo (AR) de orden 5. Estas figuras muestran que a diferencia del caso estacionario, un factor de olvido cercano a uno no proporciona necesariamente el mejor funcionamiento, existe un compromiso entre el valor de factor de olvido  $\lambda$  y la habilidad de identificar un sistema variable con el tiempo.

### 3.5 Algoritmo de Retropropagación

El perceptrón multicapas entrenado usando al algoritmo de retropropagación ha sido aplicado en la solución de diversos problemas prácticos. El perceptrón multicapas consiste en la mayoría de los casos de 3 capas: Una capa de entrada cuyo número de nodos coincide con el número de elementos del patrón de entrada. Una o más capas ocultas y una capa de salida cuyo número de nodos depende de la aplicación en particular. Por ejemplo en un perceptrón multicapas aplicado el reconocimiento de dígitos manuscritos, el número de nodos de la capa de entrada es igual al número de píxeles usados para representar los dígitos, y el número de

nodos de la capa de salida debe ser 10, cada uno de los cuales coincide con uno de los posibles resultados de la red, mientras que no hay un criterio específico para determinar el número de nodos de las capas ocultas. Así el número de nodos de las capas de entrada y salida están determinadas automáticamente por la aplicación, mientras que el número de capas ocultas así como el número de nodos de cada capa oculta se deciden a partir de diversas pruebas.

Una vez decidido el número de capas, así como el número de nodos de cada capa se procede al entrenamiento de la red, el cual se lleva a cabo de manera supervisada usando el algoritmo de retropropagación, introduciremos inicialmente algunas definiciones que facilitaron su derivación [3.5],[3.6].

- $u_{l,j}$  ---- salida del nodo  $j$  en la capa  $l$ .
- $w_{l,j,i}$  ---- coeficiente que conecta al nodo  $i$  de la capa  $l-1$  con el nodo  $j$  de la capa  $l$ .
- $x_p$  ---- patrón de entrada
- $u_{0,j}$  ---- componente  $j$ -ésima del patrón de entrada
- $d_j(x_p)$  ---- respuesta deseada del  $j$ -ésimo nodo de la salida cuando el patrón de entrada es  $x_p$
- $N_l$  ---- número de nodos en la capa  $l$
- $L$  ---- número de capas
- $P$  ---- número de patrones de entrenamiento

Con estas definiciones consideremos

$$u_{0,j} = x_j \tag{3.109}$$

Donde  $x_{p,j}$  es el  $j$ -ésimo elemento del patrón de entrada  $x_p$ .

Como la capa de entrada. Además considere que la primera componente en cada capa es igual a uno, y que el primer coeficiente  $w_{l,j,0}$  les el coeficiente de sesgo. Así

$$u_{l,0} = 1.0 \tag{3.110}$$

$$u_{l,j} = f\left(\sum_{i=0}^{N_{l-1}} w_{l,j,i} u_{l-1,i}\right) \tag{3.111}$$

donde  $f(\cdot)$  es una no linealidad sigmoideal, la cual tiene una derivada simple dada por

$$f'(\alpha) = \beta f(\alpha)(1 - f(\alpha)) \tag{3.112}$$

donde:

$$f(\alpha) = \frac{1}{1 + e^{-\beta\alpha}} \quad (3.113)$$

Los coeficientes del sistema  $w_{l,j,i}$  serán seleccionados en forma tal que la suma del cuadrado de los errores  $J_p(w)$  sea minimizado, donde

$$J(w) = \frac{1}{2} \sum_{p=1}^P \sum_{q=1}^{N_L} (u_{L,q}(x_p) - d_q(x_p))^2 = \sum_{p=1}^P J_p(w) \quad (3.114)$$

donde  $N_L$  es el número de nodos de la capa de salida. Con esta definición en mente, los pesos de la red se determinan iterativamente usando una aproximación basada en la búsqueda del gradiente, de acuerdo a la siguiente relación

$$w_{l,j,i}(k+1) = w_{l,j,i}(k) - \mu \frac{\partial J_p(w)}{\partial w_{l,j,i}} \quad (3.115)$$

Seguidamente considere la derivada de  $J_p(w)$  con respecto a  $w$  la cual se puede calcular como

$$\frac{\partial J_p(w)}{\partial w_{l,j,i}} = \frac{\partial J_p(w)}{\partial u_{l,j}} \cdot \frac{\partial u_{l,j}}{\partial w_{l,j,i}} \quad (3.116)$$

donde

$$\frac{\partial u_{l,j}}{\partial w_{l,j,i}} = \frac{\partial}{\partial w_{l,j,i}} \left[ f \left( \sum_{m=0}^{N_{l-1}} w_{l,j,m} u_{l-1,m} \right) \right] \quad (3.117)$$

$$\frac{\partial u_{l,j}}{\partial w_{l,j,i}} = f' \left( \sum_{m=0}^{N_{l-1}} w_{l,j,m} u_{l-1,m} \right) \cdot \frac{\partial}{\partial w_{l,j,i}} \left[ \sum_{m=0}^{N_{l-1}} w_{l,j,m} u_{l-1,m} \right] \quad (3.118)$$

$$\frac{\partial u_{l,j}}{\partial w_{l,j,i}} = f' \left( \sum_{m=0}^{N_{l-1}} w_{l,j,m} u_{l-1,m} \right) u_{l-1,i} \quad (3.119)$$

Seguidamente de (3.112) y (3.113) obtenemos

$$\frac{\partial u_{l,j}}{\partial w_{l,j,i}} = u_{l,j}(1 - u_{l,j})u_{l-1,i} \quad (3.120)$$

A continuación considere  $\partial J_p(w)/\partial u_{l,j}$  la cual puede ser expresada en términos de las sensibilidades de los nodos de las capas altas como sigue

$$\frac{\partial J_p(w)}{\partial u_{l,j}} = \sum_{m=1}^{N_{l+1}} \frac{\partial J_p(w)}{\partial u_{l+1,m}} \cdot \frac{\partial u_{l+1,m}}{\partial u_{l,j}} \quad (3.121)$$

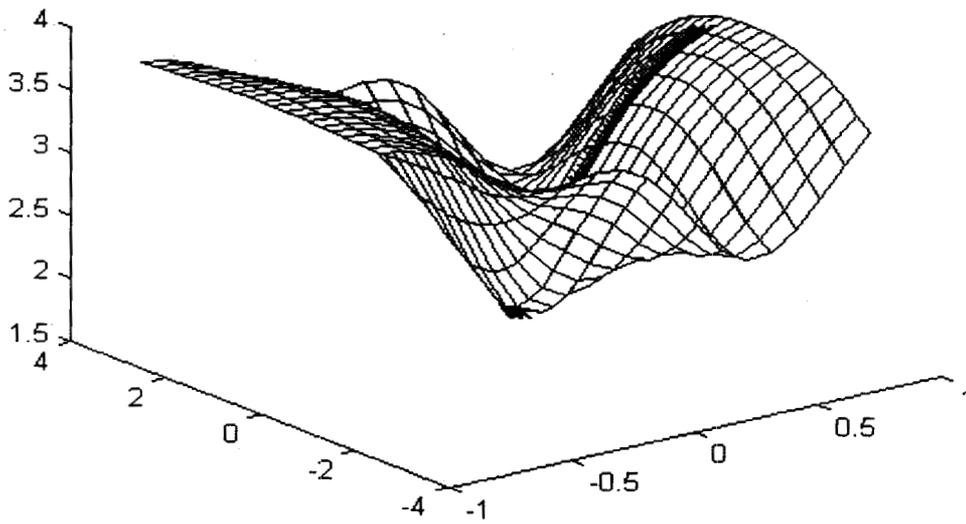
$$\frac{\partial J_p(w)}{\partial u_{l,j}} = \sum_{m=1}^{N_{l+1}} \frac{\partial J_p(w)}{\partial u_{l+1,m}} u_{l+1,m} (1 - u_{l+1,m}) w_{l+1,m,j} \quad (3.122)$$

Este proceso se continua para  $\partial J_p(w)/\partial u_{l+1,j}$  y así sucesivamente hasta alcanzar la capa de salida. Finalmente en la capa de salida

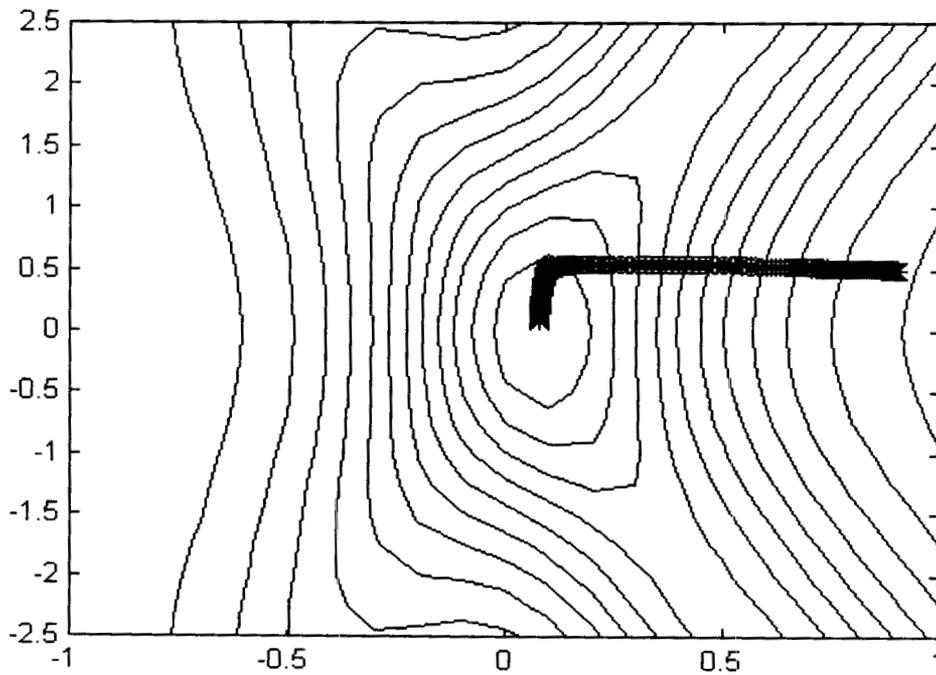
$$\frac{\partial J_p(w)}{\partial u_{L,j}} = u_{L,j}(x_p) - d_j(x_p) \quad (3.123)$$

Mientras la derivación se hace de la capa de entrada a las capas de salida, la realización de algoritmos se efectúa de la capa de salida a la de entrada. Así debido a que el error de las capas ocultas se calcula de las capas de salida hacia la capa de entrada, el algoritmo se conoce como algoritmo de retropropagación el cual esta determinado por las ecuaciones (3.115), (3.116), (3.120), (3.122), (3.123).

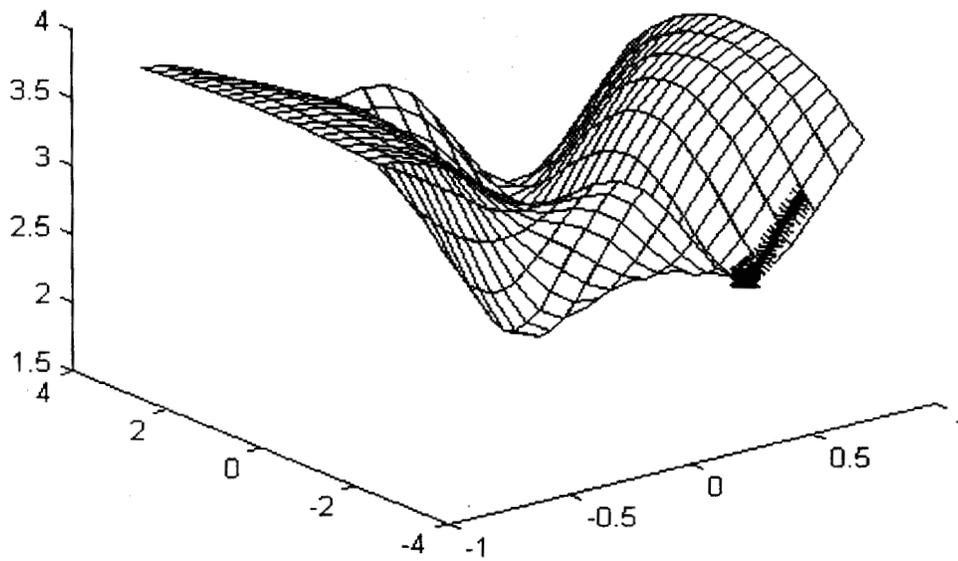
La figura 3.16 muestra el superficie de error y una trayectoria de búsqueda del algoritmo de retropropagación desde un punto de inicio, aquí el sistema empleado tiene dos pesos,  $w_0, w_1$ , los cuales se inicializan con valores aleatorios pequeños. La figura 3.17 muestra el contorno de error equivalente a la figura 3.16 y la trayectoria de búsqueda. Desde las figuras 3.16 y 3.17 tenemos que, después de varias iteraciones, el algoritmo encuentra un mínimo global. Sin embargo, de hecho de que el algoritmo de retropropagación utiliza una función no lineal, el superficie de error tiene varios mínimos locales. Las figuras 3.18 y 3.19 muestran una situación que el algoritmo converge a un mínimo local de la superficie de error. Esto depende de los valores iniciales de los pesos  $w_0, w_1$ , y los cuales calcula en manera aleatoria, por lo tanto este problema de mínimo local es un problema esencial del algoritmo de retropropagación. [3.6]



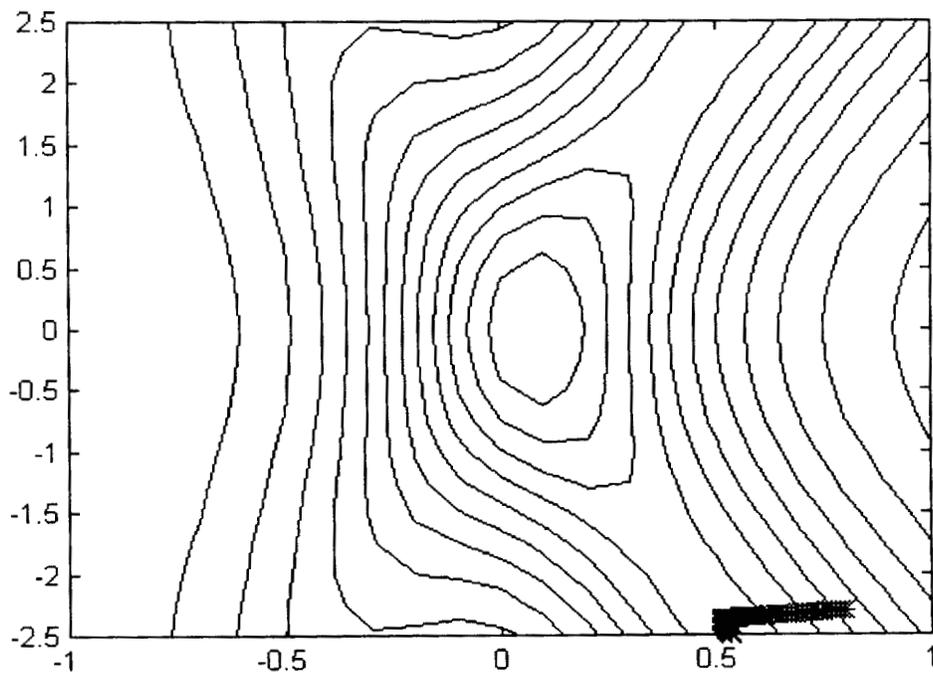
**Figura 3.16.** Superficie de error y trayectoria de los coeficientes obtenida cuando el algoritmo de retropropagación es empleado para entrenar un perceptron.



**Figura 3.17.** Trayectoria de los coeficientes de un perceptron desde un punto arbitrario hasta el mínimo global de la superficie de error.



**Figura 3.18.** Superficie de error de un perceptron y la trayectoria de sus coeficientes hasta un mínimo local de la superficie de error cuadrático medio.



**Figura 3.19.** Trayectoria de los coeficientes de un perceptron desde un punto arbitrario hasta un mínimo local de la superficie de error. Aquí el algoritmo no converge a la solución óptima.

## CAPITULO 4

### **Algoritmo LMS con Factores de Convergencia Variables en el Tiempo**

El algoritmo adaptable LMS es el mas empleado en aplicaciones de tiempo real debido principalmente a su robustez y simplicidad. Sin embargo el compromiso existente entre el factor de convergencia empleado, su velocidad de convergencia y el desajuste de la solución final limita su uso en ciertas aplicaciones prácticas. Esto se debe a que una velocidad de convergencia alta requiere un factor de convergencia relativamente grande, mientras que un desajuste pequeño de la solución final requiere un factor de convergencia pequeño, especialmente cuando la potencia del ruido aditivo es relativamente alta. Satisfacer estas dos condiciones, fundamentalmente opuestas, durante el mismo proceso de adaptación, ha sido tema de intensa investigación durante los últimos años. Esto ha dado lugar a la aparición de diversos algoritmos con factores de convergencia variables con el tiempo. Estos algoritmos de alguna manera emplean factores de convergencia grandes al principio de la adaptación para obtener velocidades de convergencia altas, reduciendo posteriormente el factor de convergencia con el fin de reducir el desajuste final. Sin embargo pese a esta intensa investigación permanecen varios problemas aun sin resolver. Este capitulo presenta un análisis del algoritmo LMS normalizado, y usando los resultados de ese análisis, propone un algoritmo LMS normalizado de paso variable que elimina algunas de las limitaciones de los esquemas existentes. Resultados obtenidos por simulación muestran que el esquema propuesto se comporta mejor que otros algoritmos existentes, especialmente cuando la potencia del ruido aditivo varía con el tiempo.

## 4.1 Introducción

El algoritmo LMS es el algoritmo de adaptación mas empleado debido principalmente a su simplicidad y robustez numérica que lo hacen apropiado para aplicaciones de tiempo real. Sin embargo el compromiso que existe entre el desajuste de la solución final y la velocidad de convergencia en ocasiones limita su uso en ciertas aplicaciones prácticas ya que un desajuste pequeño del vector de coeficientes implica un factor de convergencia pequeño, mientras que una velocidad de convergencia alta requiere de un factor de convergencia grande, lo cual traería como resultado un desajuste grande en el vector de coeficientes después de que se haya obtenido la convergencia del algoritmo.

La búsqueda de una solución al problema de obtener relativamente altas velocidades de convergencia y un pequeño desajuste de la solución final, lo cual involucra que condiciones fundamentalmente opuestas deben ser satisfechas por el factor de convergencia durante el mismo proceso de adaptación, han despertado un gran interés. Dando como resultado la aparición de un considerable número de variantes del algoritmo LMS que de alguna manera buscan satisfacer las dos condiciones. Este capítulo presenta una revisión de los principales factores que influyen tanto en la velocidad de convergencia como en el desajuste final del algoritmo LMS normalizado, y con base en estos propone un algoritmo LMS normalizado de paso variable que de alguna manera mejora las características de los algoritmos ya existentes. Se presentan además resultados experimentales que muestran el funcionamiento del algoritmo propuesto, y comparando el funcionamiento de éste con el de otros algoritmos LMS con factores de convergencia variable propuestos a la fecha, bajo diversas condiciones de operación.

## 4.2 Efecto del Factores de Convergencia en el Funcionamiento de los Algoritmos LMS

Con el fin de analizar el efecto del factor de convergencia en las características asintóticas del algoritmo de adaptación, considere el error de salida del sistema mostrado en la figura 4.1, el cual esta dado por

$$e(n) = \mathbf{X}^T(n)[\mathbf{H} - \mathbf{W}(n)] + r(n) \quad (4.1)$$

donde  $\mathbf{H}$  es el vector de coeficientes del sistema desconocido y  $r(n)$  es una señal no correlacionada con  $\mathbf{X}(n)$ , y el vector de coeficientes del sistema adaptable esta dado por

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} \mathbf{X}(n)e(n) \quad (4.2)$$

sustituyendo (4.1) en (4.2) obtenemos:

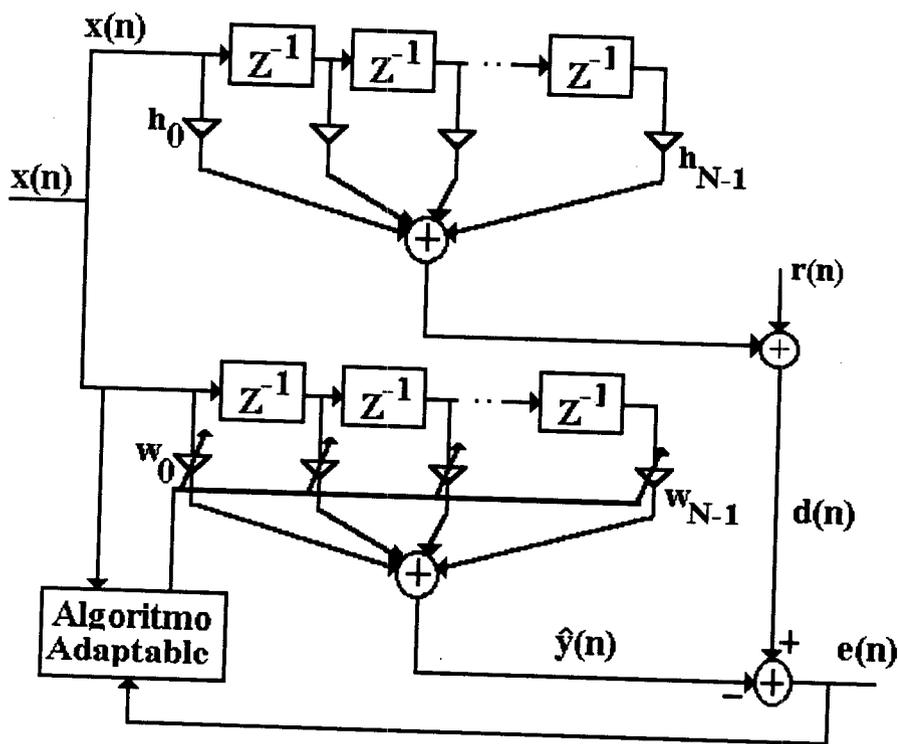


Figura 4.1. Filtro adaptable operando en una configuración de identificación de sistemas

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\alpha \mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} [\mathbf{X}^T(n)[\mathbf{H} - \mathbf{W}(n)] + r(n)] \quad (4.3)$$

Seguidamente sustrayendo  $\mathbf{H}$  en ambos lados se obtiene

$$\mathbf{W}(n+1) - \mathbf{H} = \mathbf{W}(n) - \mathbf{H} + \frac{\alpha \mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} [\mathbf{H} - \mathbf{W}(n)] + \frac{\alpha r(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (4.4)$$

a continuación defina

$$\varepsilon(n) = \mathbf{X}^T(n)[\mathbf{W}(n) - \mathbf{H}] \quad (4.5)$$

multiplique por la izquierda por  $\mathbf{X}^T(n)$ , y asuma que, cuando se ha obtenido la convergencia, [4.8]

$$\mathbf{W}(n+1) \approx \mathbf{W}(n) \quad (4.6)$$

Haciendo lo anterior, a partir de (4.4) obtenemos:

$$\varepsilon(n) = \mathbf{X}^T(n) \left[ [\mathbf{W}(n) - \mathbf{H}] + \frac{\alpha \mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} [\mathbf{H} - \mathbf{W}(n)] \right] + \frac{\alpha r(n)\mathbf{X}^T(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (4.7)$$

$$\varepsilon(n) = \left[ \mathbf{X}^T(n) - \frac{\alpha \mathbf{X}^T(n)\mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)} \right] [\mathbf{W}(n) - \mathbf{H}] + \alpha r(n) \quad (4.8)$$

$$\varepsilon(n) = (1 - \alpha)\mathbf{X}^T(n)[\mathbf{W}(n) - \mathbf{H}] + \alpha r(n) \quad (4.9)$$

$$\varepsilon(n) = (1 - \alpha)\varepsilon(n) + \alpha r(n) \quad (4.10)$$

elevando al cuadrado (4.10) y asumiendo que  $r(n)$  es no correlacionado con  $\varepsilon(n)$  obtenemos [4.8],

$$E[\varepsilon^2(n)] = (1 - \alpha)^2 E[\varepsilon^2(n)] + \alpha^2 E[r^2(n)] \quad (4.11)$$

$$(1 - 1 + 2\alpha - \alpha^2) E[\varepsilon^2(n)] = \alpha^2 E[r^2(n)] \quad (4.12)$$

$$E[\varepsilon^2(n)] = \frac{\alpha^2}{\alpha(2 - \alpha)} E[r^2(n)] \quad (4.13)$$

$$E[\varepsilon^2(n)] = \frac{\alpha}{2 - \alpha} E[r^2(n)] \quad (4.14)$$

una figura de mérito comúnmente usada es el error cuadrático medio (MSE),  $E[\varepsilon^2(n)]$ , dado por (4.14). Así expresando el MSE en dB obtenemos:

$$\text{MSE}_{\text{dB}} = 10 \log_{10} E[r^2(n)] + 10 \log_{10} \frac{\alpha}{2 - \alpha} \quad (4.15)$$

La ecuación (4.15) muestra que cuando  $\alpha = 1$  el MSE converge al nivel del ruido, mientras que para valores de  $\alpha < 1$  el MSE convergirá a un valor inferior al nivel de ruido. Así en aplicaciones en las cuales el nivel de ruido es alto, un factor de convergencia  $\alpha < 1$  podría ser necesario para obtener un MSE a niveles adecuados.

### 4.3 Constante de Tiempo

Es importante definir la constante de tiempo de la convergencia del vector de coeficientes. Así considere el vector de error dado por

$$\mathbf{V}(n) = (\mathbf{I} - \beta\mathbf{Q})^n \mathbf{V}(0) \quad (4.16)$$

La cual se puede reescribir como

$$\begin{bmatrix} v_0(n) \\ v_1(n) \\ \vdots \\ v_{N-1}(n) \end{bmatrix} = \begin{bmatrix} (1 - \beta\lambda_1)^n & & & \\ & (1 - \beta\lambda_2)^n & & \\ & & \ddots & \\ & & & (1 - \beta\lambda_N)^n \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix} \quad (4.17)$$

donde  $\mathbf{V}(n)$  es el vector de error y

$$\beta = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (4.18)$$

Seguidamente, usando los dos primeros términos de la representación en series de la función exponencial, esto es

$$e^{\left(\frac{-n}{\tau_k}\right)} = \left(1 - \frac{1}{\tau_k}\right)^n \quad (4.19)$$

y haciendo:

$$\left(1 - \frac{1}{\tau_k}\right)^n = (1 - \beta\lambda_k)^n \quad (4.20)$$

obtenemos:

$$\frac{1}{\tau_k} = \beta\lambda_k \quad (4.21)$$

$$\tau_k = \frac{1}{\beta\lambda_k} \quad (4.22)$$

Finalmente sustituyendo (4.18) en (4.22) obtenemos la constante de tiempo para a convergencia del k-esimo modo el cual esta dado por:

$$\tau_k = \frac{\mathbf{X}^T(n)\mathbf{X}(n)}{\alpha\lambda_k} \quad (4.23)$$

La ecuación (4.23) muestra que la constante de tiempo es inversamente proporcional al factor de convergencia  $\alpha$ . Así el uso de un factor de convergencia menor causará una velocidad menor de convergencia.

De las ecuaciones (4.15) y (4.23) se tiene que un factor de convergencia pequeño proporcionara un MSE por debajo del nivel de ruido lo cual es deseable en la mayoría de las aplicaciones practicas. Sin embargo, la velocidad de convergencia podría ser demasiado baja. Por el contrario una velocidad de convergencia relativamente alta se puede obtener con factores de convergencia iguales o cercanos a 1, sin embargo el valor del MSE podría ser excesivamente alto.

Satisfacer de manera simultanea estos dos requerimientos ha sido tema de intensa investigación durante los últimos años, habiendo aparecido en la literatura varios algoritmos los cuales emplean coeficientes de convergencia variables con el tiempo, los cuales intentan satisfacer de manera simultanea esos dos requerimientos. Relativamente alta velocidad de convergencia y bajo MSE. Seguidamente se propone un algoritmo LMS de con factor de convergencia variable con el tiempo el cual presenta un mejor funcionamiento que otros algoritmos similares, especialmente cuando la potencia del ruido aditivo varía con el tiempo.

#### 4.4. Algoritmo Propuesto

Con el fin de derivar el algoritmo propuesto, considere la ecuación (4.15) la cual sugiere que un valor de MSE igual al nivel de ruido se puede obtener cuando  $\alpha = 1.0$ . Este valor es adecuado cuando el nivel de ruido es bajo, ya que como se mostró anteriormente, este proporciona el menor tiempo de convergencia. Sin embargo el MSE se degrada cuando el nivel de ruido se incrementa, a menos que un factor  $\alpha$  menor sea usado, como lo sugiere la ecuación (4.15). Así con el fin de conservar un MSE pequeño aun en situaciones en las cuales el ruido aditivo se incrementa,  $(2-\alpha)/\alpha$  se debe incrementar en la misma proporción en que el nivel de ruido se incrementa.

Para lograr este objetivo, al menos aproximadamente, considere el factor de convergencia variable con el tiempo dado por

$$\alpha(n) = \frac{\overline{\varepsilon x^2(n)}}{\overline{\varepsilon x^2(n) + e^2(n)}} \quad (4.24)$$

donde  $e(n)$  es el error de identificación,  $\varepsilon$  es una constante positiva, la cual se selecciona de manera que

$$\overline{\varepsilon x^2(n)} > \overline{e^2(n)} \quad (4.25)$$

cuando la potencia de ruido aditivo es baja y así, en esas condiciones  $\alpha(n)$  se aproxime a 1.0. Es de suma importancia verificar la estabilidad del algoritmo, esto puede ser fácilmente hecho notando que

$$\overline{x^2(n)} > 0 \quad (4.26)$$

y

$$\overline{e^2(n)} > 0 \quad (4.27)$$

y por lo tanto  $0 \leq \alpha < 1.0$ , Así el algoritmo propuesto será siempre estable.

Seguidamente para analizar el efecto del factor de convergencia propuesto, defina el error de salida como

$$e(n) = \mathbf{X}^T(n)(\mathbf{H}^* - \mathbf{W}(n)) + r(n) \quad (4.28)$$

donde  $\mathbf{H}^*$  denota la solución óptima y  $\mathbf{W}(n)$  es el valor que toma  $\mathbf{W}$  al tiempo  $n$ .  $\mathbf{X}(n)$  es el vector de entrada y  $r(n)$  es la parte de la señal de referencia que no puede ser identificada por el sistema, debido a que se trata de ruido aditivo no correlacionado con  $\mathbf{X}(n)$ , o debido a que el orden del filtro no es lo suficientemente grande. Seguidamente, sustituyendo (4.28) en (4.24) obtenemos

$$\alpha(n) = \frac{\overline{\varepsilon x^2(n)}}{\overline{\varepsilon x^2(n) + [\mathbf{X}^T(\mathbf{H}^* - \mathbf{W}(n))]^2 + r^2(n)}} \quad (4.29)$$

Para analizar el comportamiento del algoritmo propuesto, después de que la convergencia ha sido obtenida, debido a posibles fluctuaciones del ruido aditivo asuma que

$$\mathbf{W}(n) \approx \mathbf{H}^* \quad (4.30)$$

por lo cual

$$\alpha(n) = \frac{\overline{\epsilon x^2(n)}}{\overline{\epsilon x^2(n)} + \overline{r^2(n)}} \quad (4.31)$$

Seguidamente analizaremos el comportamiento del algoritmo propuesto en 3 niveles de ruido aditivo que son de interés

a) **Bajo nivel de ruido.**

Asuma que cuando el nivel del ruido aditivo es bajo

$$\overline{\epsilon x^2(n)} \gg \overline{r^2(n)} \quad (4.32)$$

y sustituya (4.32) en (4.31). Haciendo lo anterior obtenemos

$$\alpha(n) \approx \frac{\overline{\epsilon x^2(n)}}{\overline{\epsilon x^2(n)}} = 1 \quad (4.33)$$

Seguidamente, sustituyendo (4.33) en (4.15) se obtiene que el error cuadrático medio cuando la potencia del ruido aditivo es moderadamente baja esta dado aproximadamente por

$$\text{MSE} = 10 \log_{10} \overline{r^2(n)} \quad (4.34)$$

Así en condiciones de bajo ruido, el MSE proporcionado por el algoritmo propuesto converge al nivel de ruido.

b) **Ruido moderado**

Seguidamente asuma que cuando el nivel de ruido es moderado

$$\overline{\epsilon x^2(n)} \approx \overline{r^2(n)} \quad (4.35)$$

Por lo cual

$$\alpha(n) = \frac{\overline{\epsilon x^2(n)}}{2\overline{\epsilon x^2(n)}} = 0.5 \quad (4.36)$$

Así, a partir de (4.15) y (4.36) se tiene que

$$\text{MSE} = 10\log_{10} \overline{r^2(n)} - 10\log_{10} \frac{2-0.5}{0.5} \quad (4.37)$$

$$\text{MSE} = 10\log_{10} \overline{r^2(n)} - 4.77 \quad (4.38)$$

De (4.38) se concluye entonces que, cuando el nivel de ruido es moderado, de acuerdo a (4.35), el algoritmo converge aproximadamente 5 dB por debajo del nivel de ruido.

### c) Nivel de ruido alto

Asuma que cuando el nivel del ruido aditivo es alto se tiene que

$$\overline{r^2(n)} \gg \overline{\epsilon x^2(n)} \quad (4.39)$$

y por lo tanto

$$\alpha(n) = \frac{\overline{\epsilon x^2(n)}}{\overline{r^2(n)}} \quad (4.40)$$

Así sustituyendo (4.40) en (4.15) obtenemos

$$\text{MSE} = 10\log_{10} \overline{r^2(n)} - 10\log_{10} \frac{2\overline{r^2(n)}}{\overline{\epsilon x^2(n)}} \quad (4.41)$$

$$\text{MSE} = 10\log_{10} \left[ \frac{\overline{\epsilon r^2(n) x^2(n)}}{2\overline{r^2(n)}} \right] \quad (4.42)$$

$$\text{MSE} = 10\log_{10} \frac{\overline{\epsilon x^2(n)}}{2} \quad (4.43)$$

Así cuando el nivel de ruido es alto, el valor asintótico del MSE será aproximadamente igual a

$$\overline{\epsilon x^2(n)} \quad (4.44)$$

De los resultados mostrados en las ecuaciones (4.34), (4.38) y (4.44) se concluye que el algoritmo propuesto, una vez obtenida la convergencia, proporciona un nivel de error menor o igual a

$$\overline{\epsilon x^2(n)} \quad (4.45)$$

De donde, por medio de  $\epsilon$  se puede determinar el nivel de señal de error.

Seguidamente considere un cambio en las estadísticas de la señal de referencia de manera que

$$\mathbf{X}^T(n)(\mathbf{H}^* - \mathbf{W}(n)) \gg \overline{r^2(n)} \approx \overline{\epsilon x^2(n)} \quad (4.46)$$

Y entonces

$$\alpha(n) = \frac{\overline{\epsilon x^2(n)}}{[\mathbf{X}^T(n)(\mathbf{H}^* - \mathbf{W}(n))]^2} \ll 1 \quad (4.47)$$

Así la velocidad de convergencia decrecería considerablemente, como se mostró en la sección 4.2. Para evitar este problema introduciremos dos condiciones que permitirán determinar si la degradación es debida a un incremento en el nivel de ruido aditivo o a cambios en las estadísticas de la señal de referencia. Así un factor  $\alpha$  constante se selecciona si

$$\overline{d^2(n)} < \overline{x^2(n)} \quad (4.48)$$

y

$$\overline{d^2(n)} < k \overline{\epsilon^2(n)} \quad (4.49)$$

Donde  $k \gg 1$  es un factor constante, en cualquier otro caso el factor variable con el tiempo dado por (4.24) es usado.

La idea detrás de esto es el hecho de que en un gran número de aplicaciones, cuando la potencia del ruido aditivo es baja se cumple (4.48). Sin embargo dependiendo de los valores de

$$\overline{x^2(n)} \text{ y } \overline{d^2(n)} \quad (4.50)$$

podría haber situaciones en las cuales, se satisfaga (4.48), aunque la potencia del ruido aditivo sea moderadamente alta. Esto podría llevar a una posible degradación del sistema. Por otro lado la condición (4.49) se cumple cuando la adaptación empieza, después de un cambio significativo en las características de  $d(n)$  o cuando el ruido aditivo se incrementa en forma considerable. Sin embargo ambas condiciones se cumplen simultáneamente solamente cuando existen cambios en las características de  $d(n)$  en situaciones de bajo ruido. Siendo este el único caso en el cual un factor de convergencia igual a uno o constante sería deseable. En cualquier otro caso un factor de convergencia variable sería preferible. Así el algoritmo propuesto estará dado por

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \beta e(n)\mathbf{X}(n) \quad (4.51)$$

Donde

$$\beta = \frac{\alpha(n)}{|\mathbf{X}(n)|^2} \quad (4.52)$$

y

$$\alpha(n) = \begin{cases} 1.0 & \text{si } \overline{x^2(n)} > \overline{d^2(n)} \text{ y } \overline{d^2(n)} < k\overline{e^2(n)} \\ \frac{\epsilon \overline{x^2(n)}}{\epsilon \overline{x^2(n)} + \overline{e^2(n)}} & \text{otro caso} \end{cases} \quad (4.53)$$

Los promedios temporales pueden ser fácilmente estimados como

$$\overline{x^2(n)} = (1-\gamma)\overline{x^2(n-1)} + \gamma x^2(n) \quad (4.54)$$

Donde  $1/\gamma$  es aproximadamente igual al número de datos usados para estimar

$$\overline{x^2(n)} \quad (4.55)$$

#### 4.5. Algoritmos del Tipo LMS con Factores de Convergencia Variables con el Tiempo Propuestos por otros Autores.

El desarrollo de algoritmos del tipo LMS con factores de convergencia variables con el tiempo, ha sido un tema de investigación activa durante los últimos años. Durante estos años se han propuesto un número importante de algoritmos que emplean factores de convergencia variables con el tiempo, como una forma de satisfacer de manera simultanea los requerimientos de altas velocidades de convergencia inicial, ec. (4.23) y razonablemente bajos errores después de que la convergencia ha sido obtenida, ec. (4.15). En esta sección se presenta una breve revisión de algunos de los algoritmos de este tipo más importantes propuestos hasta la fecha.

##### 4.5.1. Algoritmo VSLMS

Uno de los primeros, y más exitosos, algoritmos LMS con factores de convergencia variables con el tiempo, el cual ha sido ampliamente usado como parámetro para medir las características de convergencia de otros algoritmos similares propuestos recientemente, es el llamado Algoritmo VSLMS [4.10], en el cual el factor de convergencia es igual al valor cuadrático medio del error de salida, esto es

$$\alpha'(n) = \overline{e^2(n)} \quad (4.56)$$

De la definición de  $\alpha'(n)$ , es relativamente sencillo mostrar que la ecuación (4.56) se puede estimar recursivamente mediante la relación

$$\alpha'(n) = \gamma\alpha'(n-1) + (1-\gamma)e^2(n) \quad (4.57)$$

Esta selección de  $\alpha'(n)$  podría presentar dos problemas importantes. El primero relativo a la estabilidad ya que si  $\alpha'(n)$  es mayor o igual a dos el sistema llegaría a ser inestable, y segundo el factor de convergencia podría llegar a ser demasiado pequeño impidiendo que el algoritmo pudiera responder en forma adecuada a variaciones en las estadísticas de la señal de referencia. Para reducir esto problemas y obtener un funcionamiento razonablemente bueno del algoritmo cuando este opere en situaciones no estacionarias, el factor de convergencia se define de la siguiente manera.

$$\alpha(n) = \begin{cases} 1 & \alpha'(n) \geq 1 \\ \alpha_{\min} & \alpha'(n) < \alpha_{\min} \\ \alpha'(n) & \alpha_{\min} < \alpha'(n) < 1.0 \end{cases} \quad (4.58)$$

Seguidamente analizaremos el comportamiento del algoritmo asumiendo que el error se encuentra compuesto por la ecuación (4.28). De esta ecuación se obtiene que, una vez que el sistema ha convergido  $\mathbf{H}^* = \mathbf{W}(n)$ . Así

$$\overline{e^2(n)} = \overline{r^2(n)} \quad (4.59)$$

por lo cual

$$\alpha(n) = \overline{r^2(n)} \quad (4.60)$$

De la ecuación (4.60) se observa que el factor de convergencia es directamente proporcional al valor cuadrático medio del ruido aditivo. Esto permitirá obtener factores de convergencia apropiados para aplicaciones en las cuales el nivel de ruido es moderadamente bajo [4.14]. Sin embargo, este podría experimentar problemas en situaciones en las cuales el nivel de ruido es moderadamente alto o cambia con el tiempo, como sucede en aplicaciones tales como la cancelación de eco [4.14]. Así este algoritmo, que fue de los de su género que se propusieron inicialmente luce apropiado para aplicaciones en las cuales el nivel del ruido es moderadamente bajo, permaneciendo además constante.

#### 4.5.2. Algoritmo VECLMS

Uno de los últimos algoritmos LMS con factores de convergencia variables con el tiempo propuestos mas recientes es el algoritmo VECLMS [4.13]. Este intenta mejorar el comportamiento del algoritmo VSLMS en situaciones en las cuales la potencia del ruido es relativamente alta, haciendo el factor de convergencia proporcional a la correlación existente entre dos muestras consecutivas de la señal de error. Con el fin de obtener el factor de convergencia, se define

$$p(n) = \overline{e(n)e(n-1)} \quad (4.61)$$

como el valor estimado de la correlación entre  $e(n)$  y  $e(n-1)$ , donde  $e(n)$  es la señal de error al tiempo  $n$ , pudiéndose estimar  $p(n)$  en forma recursiva mediante la relación

$$p(n) = \gamma p(n-1) + (1-\gamma)e(n)e(n-1) \quad (4.62)$$

Seguidamente el factor de convergencia se define como el valor cuadrático medio de  $p(n)$ , estimándose este en la forma siguiente

$$\alpha'(n) = \gamma \alpha'(n-1) + (1-\gamma)p^2(n) \quad (4.63)$$

Esta selección de  $\alpha'(n)$  podría, al igual que en el caso anterior, presentar dos problemas importantes. El primero relativo a la estabilidad ya que si  $\alpha'(n)$  es mayor o igual a dos el sistema llegaría a ser inestable, y segundo el factor de convergencia podría llegar a ser demasiado pequeño impidiendo que el algoritmo pudiera responder en forma adecuada a variaciones en las estadísticas de la señal de referencia. Para reducir estos problemas y

obtener un funcionamiento razonablemente bueno del algoritmo cuando este opere en situaciones no estacionarias, el factor de convergencia se define de la siguiente manera.

$$\alpha(n) = \begin{cases} 1 & \alpha'(n) \geq 1 \\ \alpha_{\min} & \alpha'(n) < \alpha_{\min} \\ \alpha'(n) & \alpha_{\min} < \alpha'(n) < 1.0 \end{cases} \quad (4.64)$$

Seguidamente analizaremos brevemente el comportamiento del algoritmo asumiendo que el error se encuentra compuesto por la ecuación (4.28). De esta ecuación se obtiene que, una vez que el sistema ha convergido  $\mathbf{H}^* = \mathbf{W}(n)$ . Así

$$\overline{e(n)e(n-1)} = \overline{r(n)r(n-1)} \quad (4.65)$$

por lo cual

$$\alpha(n) = \overline{(r(n)r(n-1))^2} \quad (4.66)$$

De la ecuación (4.66) se observa que el factor de convergencia sería prácticamente independiente del ruido aditivo,  $r(n)$ , si este ruido blanco, lo cual evitaría o reduciría en forma significativa, al menos, la distorsión introducida por el incremento del ruido aditivo. Sin embargo, en muchas aplicaciones prácticas el ruido aditivo no es blanco, sino que podría presentar una alta correlación entre muestras consecutivas, provocando que  $p(n)$  fuera diferente de cero. Pudiendo llegar a tener valores cercanos a  $r^2(n)$ , con lo cual su funcionamiento podría llegar a ser muy similar al algoritmo VSLMS, descrito anteriormente. Así este algoritmo luce apropiado para aplicaciones en las cuales el nivel del ruido es moderadamente bajo o en aquellas aplicaciones en las cuales el ruido aditivo pudiera considerarse aproximadamente blanco.

### 4.5.3. Algoritmo CC\_LMS

Otro algoritmo basado en un principio similar el cual resuelve en parte algunos de los problemas que presentan los algoritmos VSLMS y VECLMS es el algoritmo CC\_LMS propuesto en [4.14]. Aquí el factor de convergencia es proporcional al valor estimado de la primera componente de la correlación cruzada entre la señal de error y la salida del filtro. esto es

$$\alpha'(n) = \overline{e(n)\hat{y}(n)} \quad (4.67)$$

donde  $e(n)$  es la señal de error y  $\hat{y}(n)$  es la señal de salida del filtro. De la definición de  $\alpha'(n)$ , esta se puede estimar recursivamente mediante la siguiente relación

$$\alpha'(n) = \gamma\alpha'(n-1) + (1-\gamma)e(n)\hat{y}(n) \quad (4.68)$$

Esta selección de  $\alpha'(n)$  podría presentar tres problemas importantes. El primero relativo a la estabilidad ya que si  $\alpha'(n)$  es mayor o igual a dos el sistema llegaría a ser inestable, segundo el factor de convergencia podría llegar a ser demasiado pequeño impidiendo que el algoritmo pudiera responder en forma adecuada a variaciones en las estadísticas de la señal de referencia y tercero,  $\alpha'(n)$  depende de la potencia de la señal de entrada. Para reducir estos problemas y obtener un funcionamiento razonablemente bueno del algoritmo cuando este opere en situaciones no estacionarias, el factor de convergencia se define de la siguiente manera.

$$\alpha(n) = \begin{cases} 1 & \frac{\alpha'(n)}{x^2(n)} \geq 1 \\ \alpha_{\min} & \frac{\alpha'(n)}{x^2(n)} < \alpha_{\min} \\ \frac{\alpha'(n)}{x^2(n)} \alpha'(n) & \alpha_{\min} < \frac{\alpha'(n)}{x^2(n)} < 1.0 \end{cases} \quad (4.69)$$

donde la potencia de la señal de entrada se puede estimar también en forma, de manera que

$$\overline{x^2(n)} = \gamma \overline{x^2(n-1)} + (1-\gamma)x^2(n) \quad (4.70)$$

Seguidamente analizaremos el comportamiento del algoritmo asumiendo que el error se encuentra compuesto por la ecuación (4.28). De esta ecuación se obtiene que, una vez que el sistema ha convergido  $\mathbf{H}^* = \mathbf{W}(n)$ ,

$$\overline{e(n)\hat{y}(n)} = \overline{r(n)\hat{y}(n)} \quad (4.71)$$

por lo cual, asumiendo que el ruido es no correlacionado con la señal de entrada, de (4.69) y (4.71) obtenemos que

$$\alpha(n) = 0 \quad (4.72)$$

De la ecuación (4.72) se observa que el factor de convergencia es independiente del ruido aditivo, por lo cual es capaz de evitar distorsión en el funcionamiento del sistema aun cuando el ruido sea altamente autocorrelacionado, evitando así el problema presente en el algoritmo VECLMS. El sistema podría, sin embargo experimentar problemas cuando la potencia del ruido varía rápidamente con el tiempo a causa de una imprecisa estimación de los parámetros involucrados en el cálculo del factor de convergencia, dados por las ecuaciones (4.68)-(4.70), los cuales se estiman usando filtros pasa bajas IIR de primer orden.

#### 4.5.4. Algoritmo ACFLMS

Otro algoritmo de paso variable propuesto es el algoritmo ACFLMS [4.12] en el cual también se modifica el factor de convergencia con el tiempo de manera tal que el factor de convergencia disminuye conforme el algoritmo converge. Aquí el factor de convergencia se obtiene a partir de la relación

$$\alpha(n) = 1.0 - \frac{1.0}{1.0 + \epsilon e(n)\hat{y}(n)} \quad (4.73)$$

donde  $e(n)$  es la señal de error y  $\hat{y}(n)$  es la señal de salida del filtro. Asumiendo, como en el caso anterior, que la señal de entrada es no correlacionada con el ruido aditivo, una vez que la convergencia ha sido obtenida,

$$\overline{e(n)\hat{y}(n)} \approx 0 \quad (4.74)$$

y entonces, de la ec. (4.73), se obtiene que  $\alpha(n)$  se aproxima a cero también. Así el algoritmo ACFLMS presenta baja sensibilidad al ruido aditivo. Este algoritmo, sin embargo, presenta los mismos problemas que el CC\_LMS descrito en la sección anterior, esto es degradación en su funcionamiento cuando la potencia del ruido aditivo varía rápidamente con el tiempo, debido a que, como en el caso anterior, el primer elemento de la correlación cruzada entre la señal de error y la salida del sistema, involucrado en la estimación del factor de convergencia, se estima usando un filtro IIR pasa bajas de primer orden.

#### 4.5.5. Algoritmo NACFLMS

Una modificación del algoritmo ACFLMS [4.12], propuesta recientemente es el algoritmo NACFLMS [4.12] en el cual también se modifica el factor de convergencia con el tiempo de manera tal que el factor de convergencia disminuye conforme el algoritmo converge. Aquí el factor de convergencia se obtiene a partir de la relación

$$\alpha(n) = 1.0 - \frac{1.0}{1.0 + \epsilon e_p(n)} \quad (4.75)$$

donde  $e_p(n)$  esta dada por

$$e_p(n) = \frac{e^2(n) \mathbf{X}(n)_{max}^2}{e^2(n)_{max} \mathbf{X}(n)^2} \quad (4.76)$$

donde  $e(n)$  es la señal y  $\mathbf{X}(n)$  el vector de entrada.

Una vez que la convergencia ha sido obtenida, esto es

$$y(n) \approx \hat{y}(n) \quad (4.77)$$

donde  $y(n)$  es la señal a ser estimada y  $\hat{y}(n)$  es la señal de salida del filtro de las ecs. (4.76) y (4.77) se obtiene

$$e_{p'}(n) = \frac{r^2(n) \mathbf{X}(n)_{max}^2}{r^2(n)_{max} \mathbf{X}(n)^2} \quad (4.78)$$

donde  $r(n)$  es la señal de ruido aditivo. Finalmente, asumiendo que la señal de entrada es estacionaria de manera que

$$|\mathbf{X}(n)|_{max} \approx |\mathbf{X}(n)| \quad (4.79)$$

de las ecs. (4.75) y (4.78) se obtiene que

$$\alpha(n) = 1.0 - \frac{1.0}{1.0 + \epsilon \frac{r^2(n)}{r^2(n)_{max}}} \quad (4.80)$$

Así cuando el ruido se incrementa de manera que  $r^2(n) = r^2(n)_{max}$  el factor de convergencia se aproxima a 1.0. Esto provocaría una mayor distorsión en las características de convergencia del algoritmo NACFLMS que los anteriores ya que un factor de convergencia cercano a uno aumenta la sensibilidad del algoritmo al ruido aditivo.

#### 4.6. Simulaciones Computacionales

La figura 4.2 muestra el factor de convergencia del algoritmo propuesto y el MSE con respecto al tiempo, obtenido cuando el algoritmo propuesto es usado para identificar un sistema desconocido de orden 260. La señal de entrada fue una secuencia de ruido blanco y la potencia del ruido aditivo variable con el tiempo. La figura 4.2 muestra que el factor de convergencia se adapta, disminuyendo o incrementando su valor de manera tal que el MSE permanezca prácticamente constante después de que la convergencia se ha obtenido. Permitiendo además que el algoritmo siga las variaciones ocurridas en las características del sistema siendo identificado. Las figuras 4.3 y 4.4 muestran las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] y VECLMS [4.13] cuando todos ellos son requeridos a identificar un sistema desconocido de orden 260. En ambos casos la señal de entrada fue una secuencia de ruido blanco con potencia igual a 0 dB. En la figura 4.3 el nivel de ruido aditivo

fue de  $-20$  dB, mientras que en la figura 4.4 el nivel de ruido aditivo fue de  $-40$  dB. La figura 4.5 muestra las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] cuando estos son requeridos a identificar un sistema desconocido de orden 260. La señal de entrada fue una señal de voz actual con potencia promedio igual a  $0$  dB y el ruido aditivo un proceso autoregresivo de orden 5 con potencia promedio igual a  $-20$  dB. La figura 4.6 por su parte muestra las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] y VECLMS [4.13] cuando son requeridos a identificar un sistema de orden 260. La señal de entrada fue un proceso autoregresivo de orden 5 cuya función de correlación es similar a la de una señal de voz actual, siendo su potencia promedio igual  $0$  dB, mientras que el ruido aditivo también fue un proceso autoregresivo con potencia promedio igual a  $-40$  dB. En todos los casos, en el algoritmo propuesto  $\epsilon=0.001$ . Las figuras 4.2-4.6 muestran que el algoritmo propuesto en todos los casos converge por debajo del nivel del ruido aditivo proporcionando un nivel de error similar, independientemente de las características de la señal de entrada. Mientras que los demás algoritmos muestran, en algunos casos, características de convergencia las cuales dependen de las estadísticas de las señales involucradas.

Las figuras 4.7 y 4.8 muestran las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] y VECLMS [4.13] cuando todos ellos son requeridos a identificar un sistema desconocido de orden 260. Aquí la potencia del ruido aditivo se incrementa bruscamente después de un cierto tiempo. En la figura 4.7 la señal de entrada fue una secuencia de ruido, mientras que en la figura 4.8 la señal de entrada fue un proceso autoregresivo cuya función de autocorrelación es idéntica a la de una señal de voz actual. En ambos casos la potencia promedio de la señal de entrada fue igual a  $0$  dB. Las figuras 4.7 y 4.8 muestran que el MSE del algoritmo propuesto permanece prácticamente constante pese al incremento en la potencia del ruido aditivo, mientras que el MSE proporcionado por los otros algoritmos se incrementa considerablemente. Este incremento en el MSE lleva a una degradación del funcionamiento del sistema.

Las figuras 4.9 y 4.10 muestran las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] y VECLMS [4.13] cuando todos ellos son requeridos a identificar un sistema desconocido de orden 260. Aquí la respuesta a impulso del sistema siendo identificado cambia bruscamente después de un cierto tiempo. En la figura 4.9 la señal de entrada fue una secuencia de ruido blanco, mientras que en la figura 4.10 la señal de entrada fue un proceso autoregresivo cuya función de autocorrelación es idéntica a la de una señal de voz actual. En ambos casos la potencia promedio de la señal de entrada fue igual a  $0$  dB. Estas figuras muestran la capacidad del algoritmo propuesto para seguir variaciones en las características de las señales involucradas en el proceso de adaptación.

Las figuras 4.11 y 4.12 muestran las características de convergencia de los algoritmos (1) TVSLMS (propuesto), (2) NLMS [4.7], (3) VSLMS [4.10], (4) NACFLMS [4.12] y VECLMS [4.13] cuando todos ellos son requeridos a identificar un sistema desconocido de orden 260. Aquí la potencia del ruido aditivo y la función de transferencia del sistema siendo identificado cambian con el tiempo. En la figura 4.11 la señal de entrada fue una secuencia de

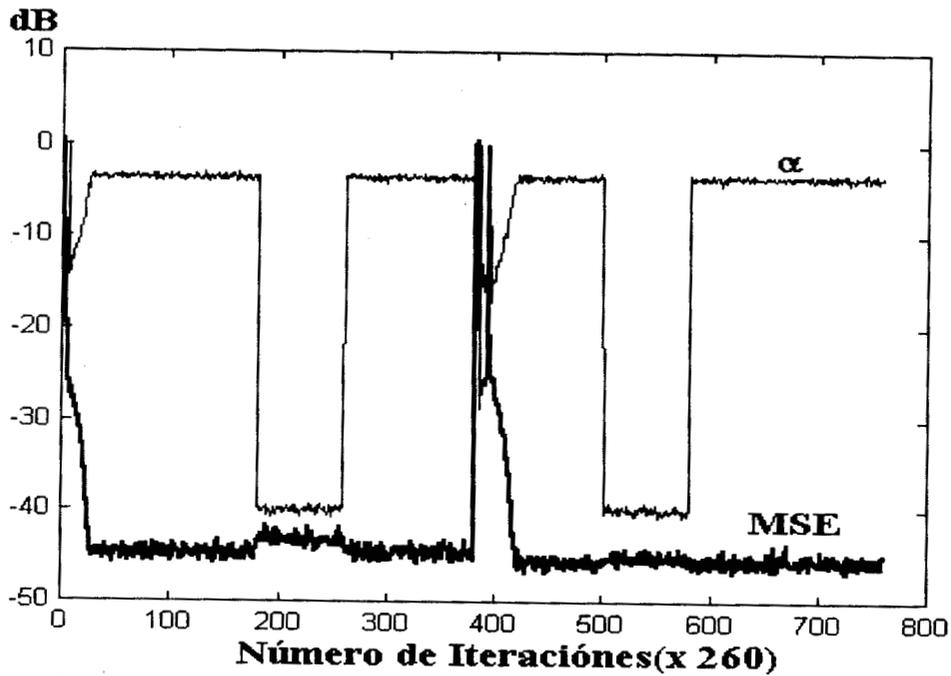


Figura 4.2. Características de convergencia del algoritmo propuesto, y el factor de convergencia, cuando es requerido a identificar un sistema desconocido variable con el tiempo de orden 260. La potencia del ruido aditivo fluctúa con el tiempo.

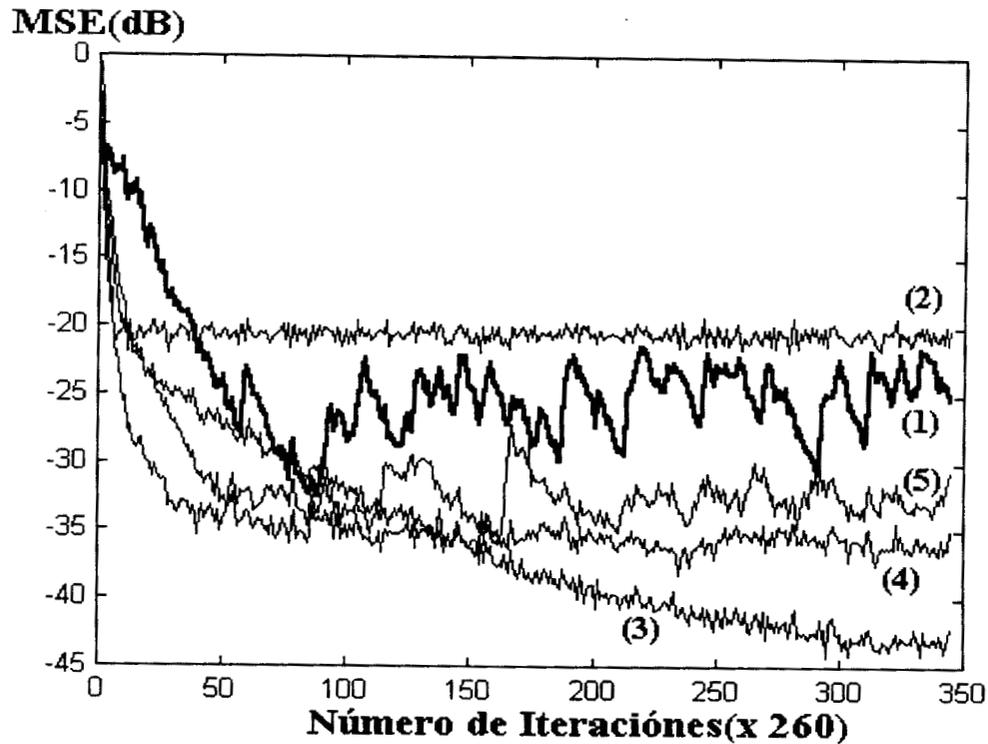
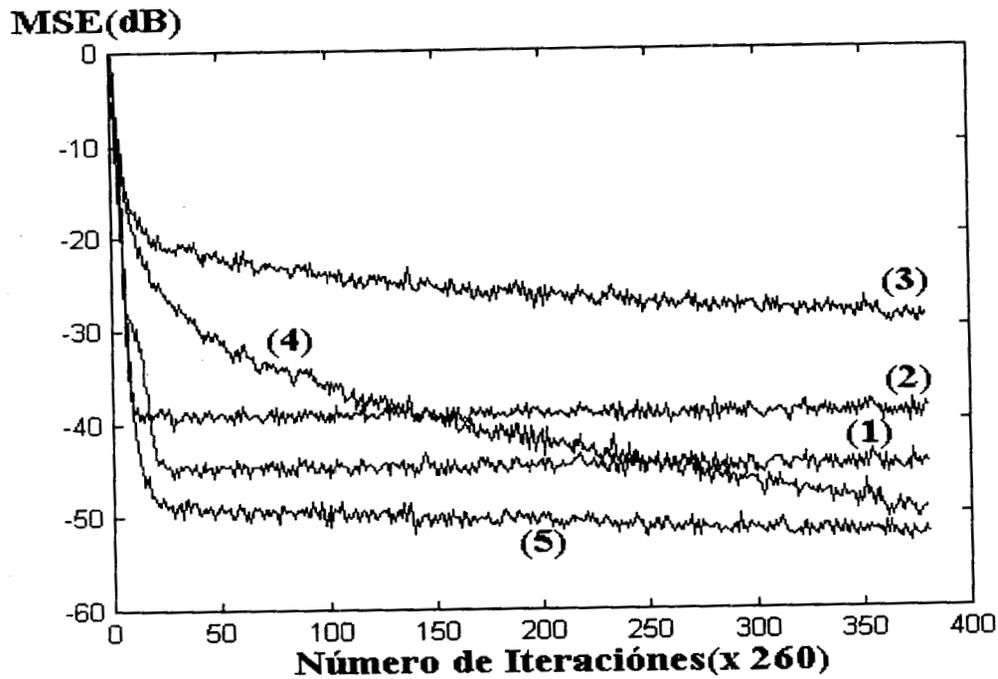
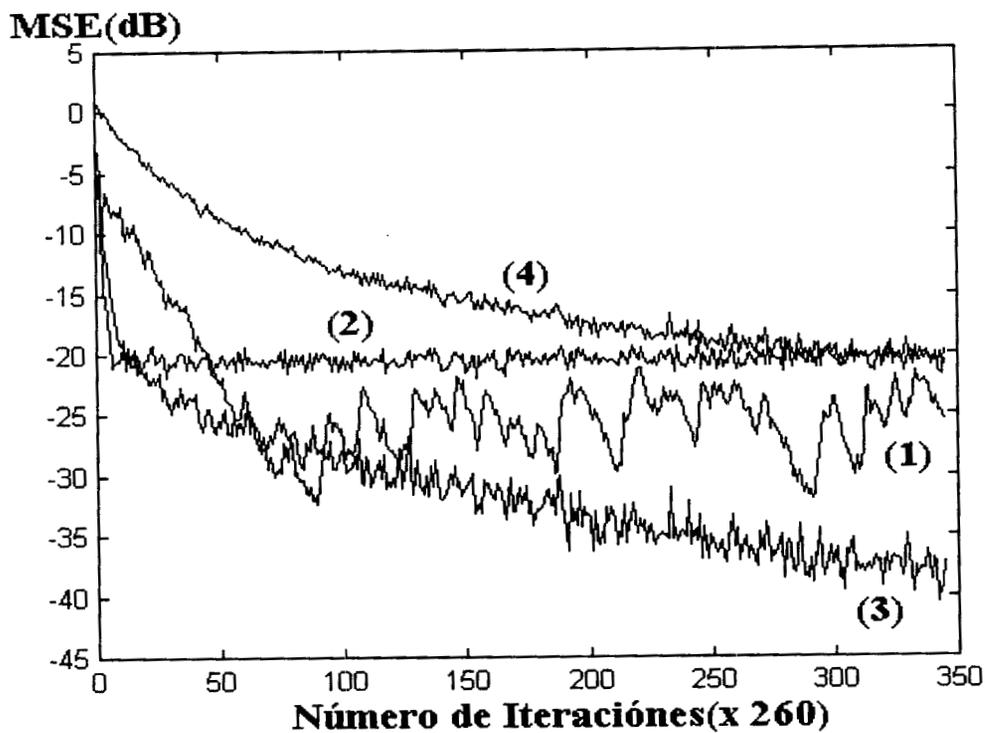


Figura 4.3. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es ruido blanco con un relación señal a ruido aditivo de 20 dB. Los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.



**Figura 4.4.** Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es ruido blanco con un relación señal a ruido aditivo de 40 dB. Los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.



**Figura 4.5.** Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es un señal de vos actual con un relación señal a ruido aditivo de 20 dB. Los algoritmos (2) NLMS [7], (3) VSLMS [10] y (4) NACFLMS[12] se muestran para su comparación.

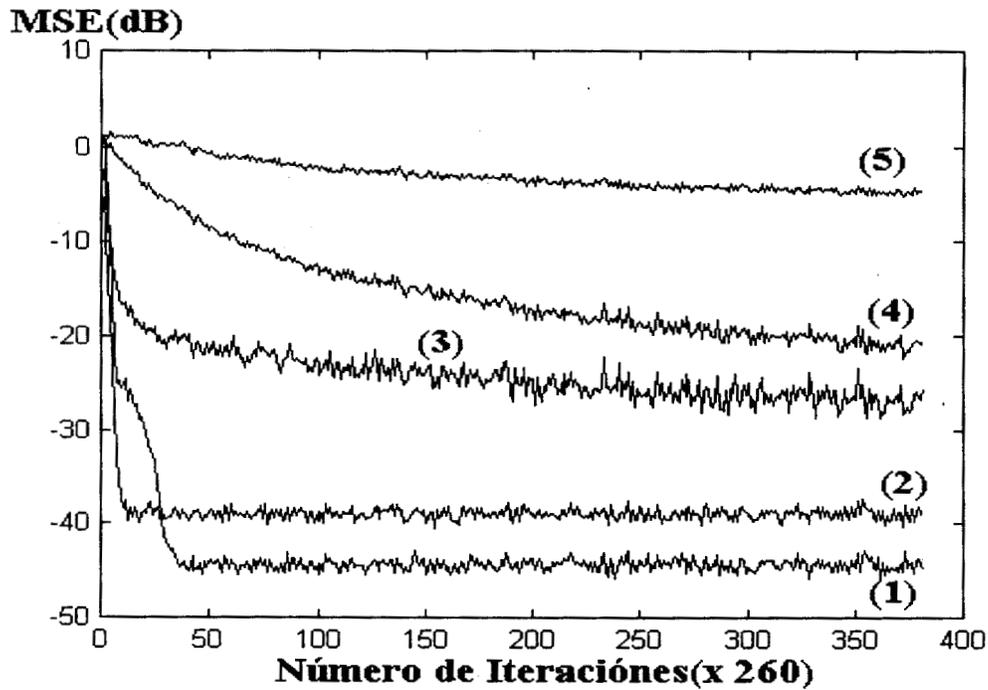


Figura 4.6. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es un proceso AR con un relación señal a ruido aditivo de 40 dB. Los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.

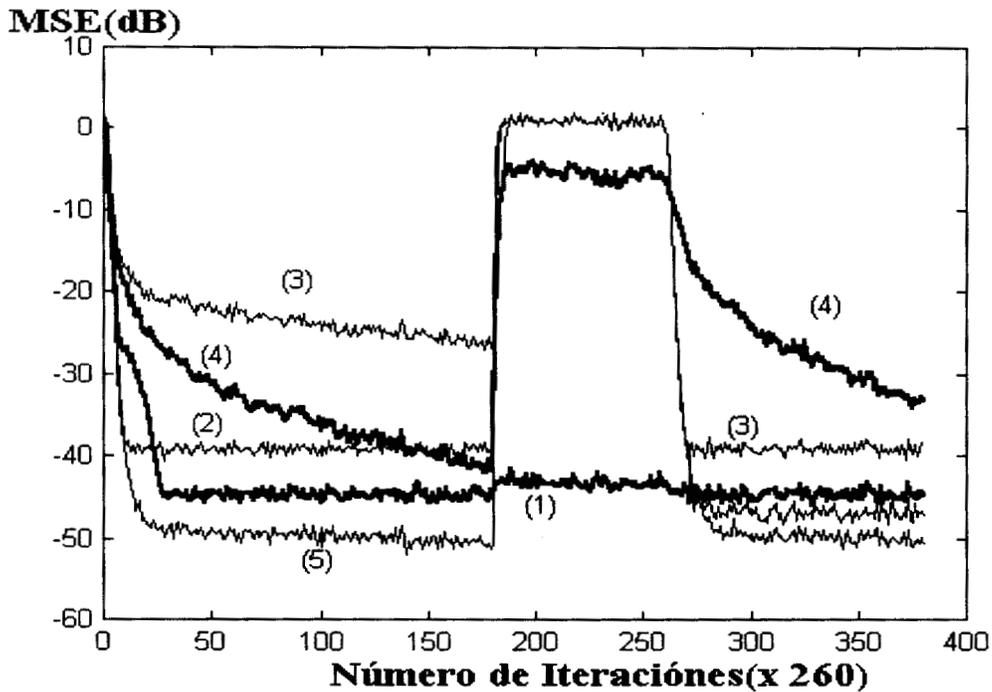


Figura 4.7. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es ruido blanco. La relación señal a ruido inicialmente es de 40 dB, seguidamente disminuye a 0 dB y finalmente se incrementa a 40 dB. La convergencia de los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.

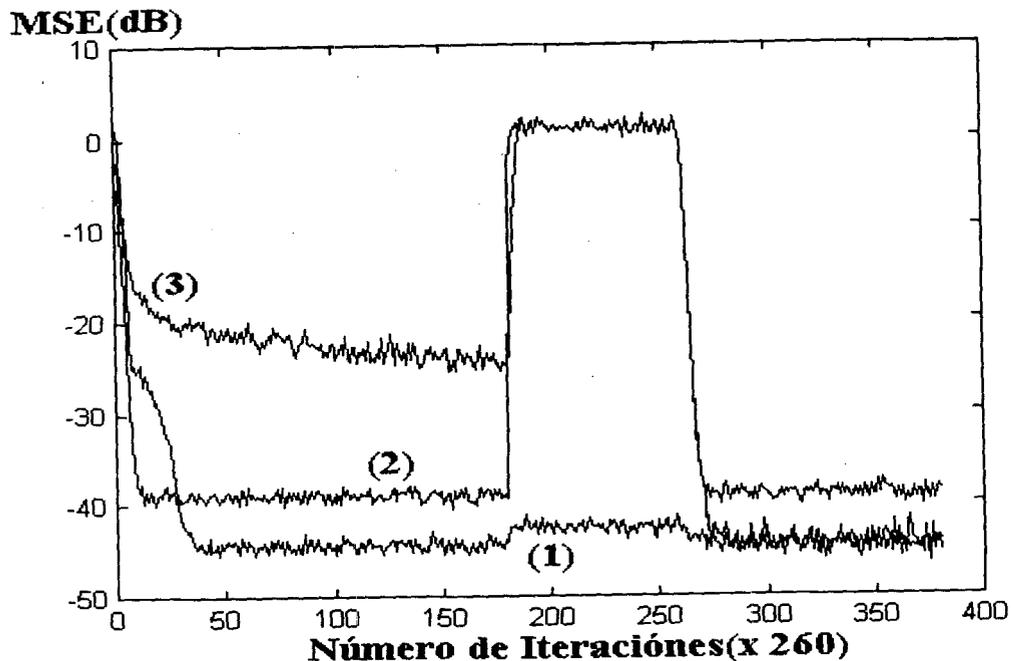


Figura 4.8. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es un proceso AR. La relación señal a ruido inicialmente es de 40 dB, seguidamente disminuye a 0 dB y finalmente se incrementa a 40 dB. La convergencia de los algoritmos (2) NLMS [7], (3) VSLMS [10] se muestran para su comparación.

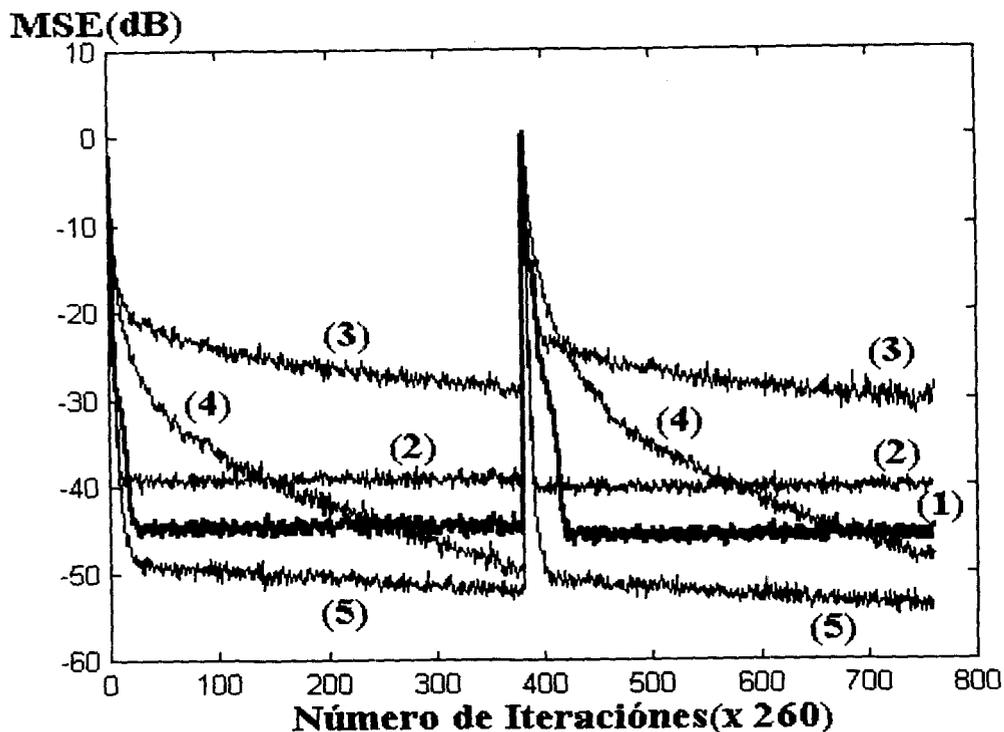


Figura 4.9. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema variable con el tiempo de orden 260. La señal de entrada es ruido blanco con una SNR de 40 dB. La convergencia de los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS [12] y (5) VECLMS [13] se muestran para su comparación.

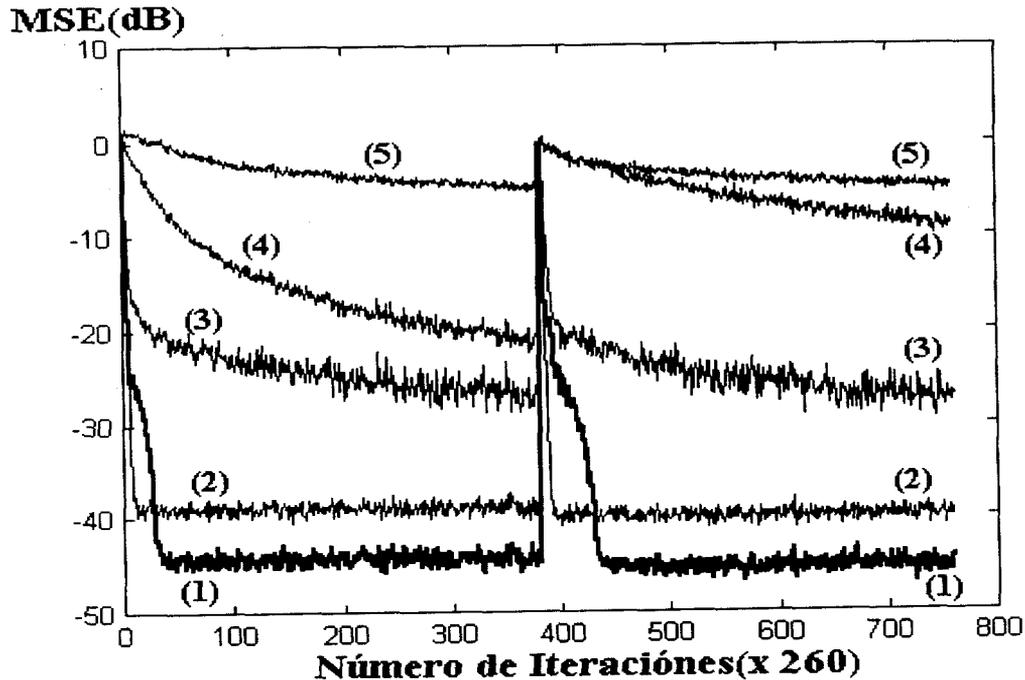


Figura 4.10. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema variable con el tiempo de orden 260. La señal de entrada es un proceso AR con una SNR de 40 dB. La convergencia de los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.

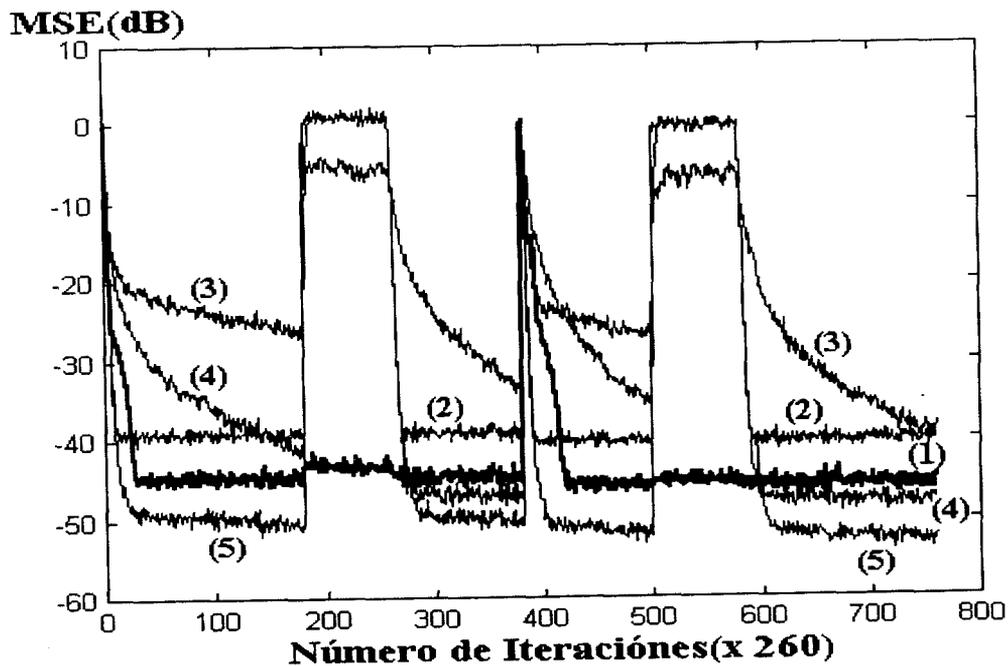
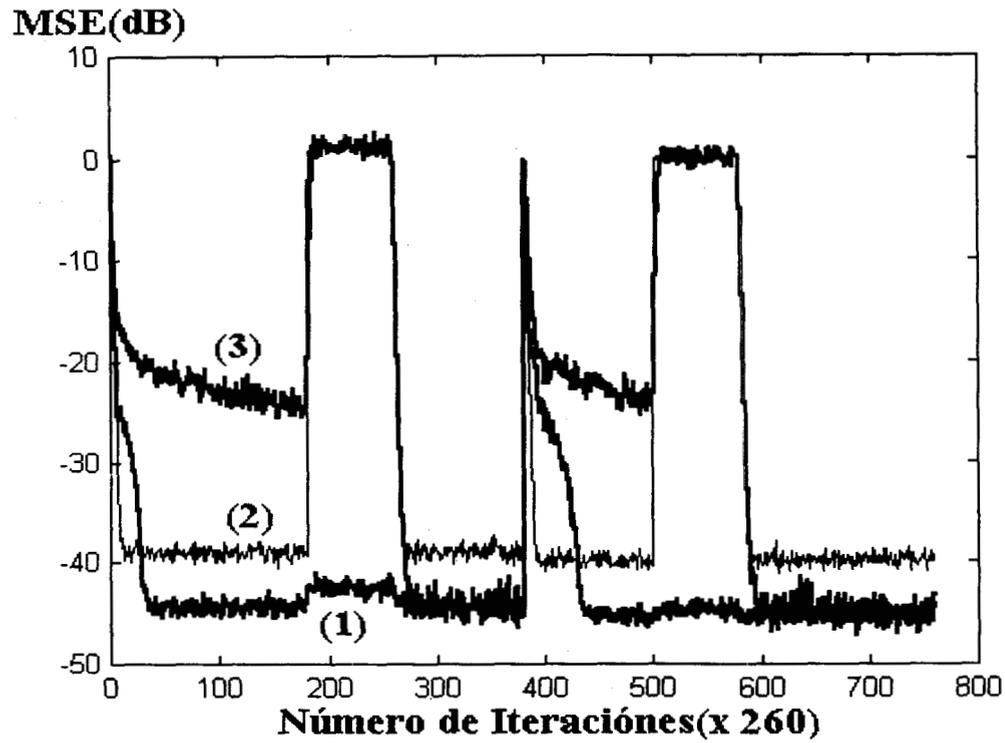
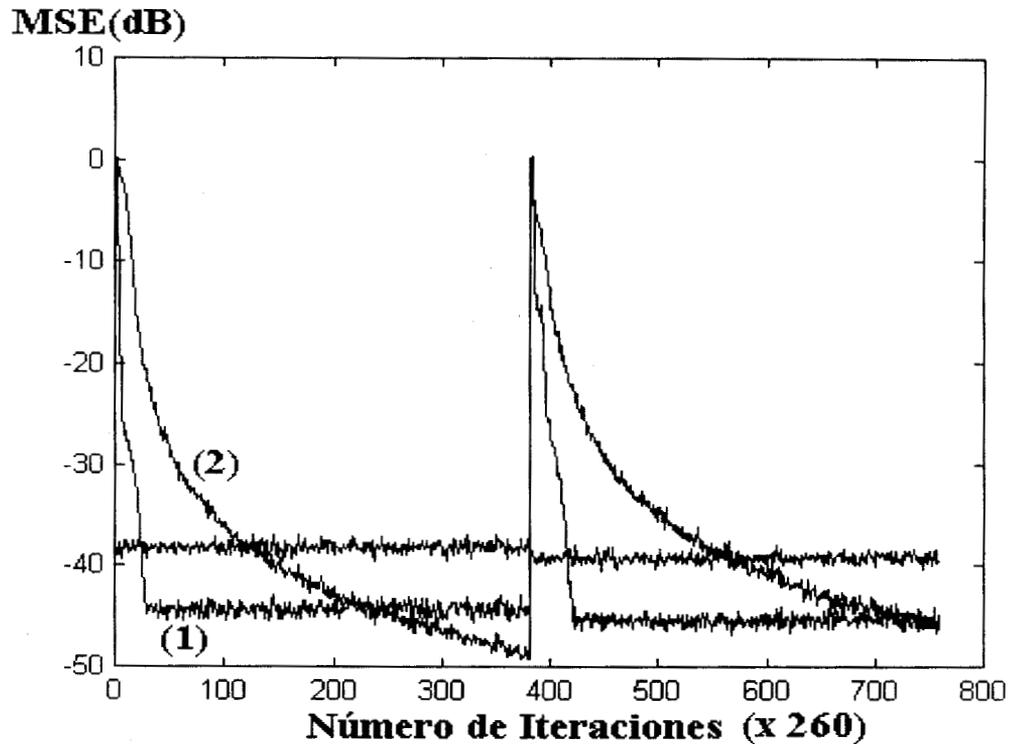


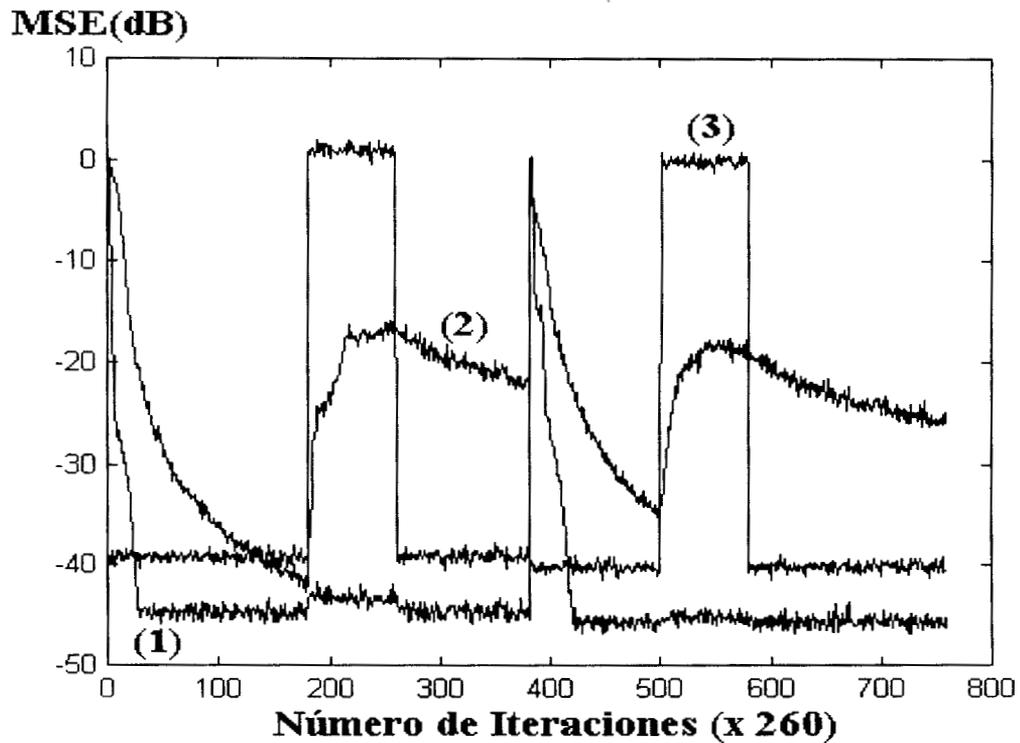
Figura 4.11. Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema variable con el tiempo de orden 260. La señal de entrada es ruido blanco. La relación señal a ruido inicialmente es de 40 dB, seguidamente disminuye a 0 dB y finalmente se incrementa a 40 dB. La convergencia de los algoritmos (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12] y (5) VECLMS [13] se muestran para su comparación.



**Figura 4.12.** Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es un proceso AR. La relación señal a ruido inicialmente es de 40 dB, seguidamente disminuye a 0 dB y finalmente se incrementa a 40 dB. La convergencia de los algoritmos (2) NLMS [7], y (3) VSLMS [10] se muestran para su comparación.



**Figura 4.13.** Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es una secuencia de ruido. La relación señal a ruido es aproximadamente 40 dB. La convergencia del algoritmo (2) CCLMS se muestra para su comparación.



**Figura 4.14.** Características de convergencia del algoritmo propuesto cuando es requerido a identificar un sistema de orden 260. La señal de entrada es una secuencia de ruido. La relación señal a ruido inicialmente es de 40 dB, seguidamente disminuye a 0 dB y finalmente es incrementa a 40 dB. La convergencia del algoritmo (2) CCLMS se muestra para su comparación.

ruido, mientras que en la figura 4.12 la señal de entrada fue un proceso autoregresivo cuya función de autocorrelación es idéntica a la de una señal de voz actual. En ambos casos la potencia promedio de la señal de entrada fue igual a 0 dB. Estas figuras muestran que el MSE del algoritmo propuesto permanece prácticamente constante pese al incremento en la potencia del ruido aditivo, mientras que el MSE proporcionado por los otros algoritmos se incrementa considerablemente. Los resultados muestran además la capacidad del algoritmo propuesto para seguir variaciones en la respuesta a impulso del sistema siendo identificado.

Las figuras 4.13 y 4.14 muestran las características de convergencia de los algoritmos (1) TVSLMS (propuesto) y CCLMS [4.14] cuando todos ellos son requeridos a identificar un sistema desconocido de orden 260. Aquí la potencia del ruido aditivo y la función de transferencia del sistema siendo identificado cambian con el tiempo. En ambas figuras la señal de entrada fue una secuencia de ruido y la potencia promedio de la señal de entrada fue igual a 0 dB. Estas figuras muestran que el MSE del algoritmo propuesto permanece prácticamente constante pese al incremento en la potencia del ruido aditivo, mientras que el MSE proporcionado por el algoritmo CCLMS se incrementa después de cierto tiempo. Los resultados muestran además la capacidad del algoritmo propuesto para seguir variaciones en la respuesta a impulso del sistema siendo identificado.

#### 4.7. Conclusiones

En este capítulo se propuso un algoritmo LMS normalizado con factor de convergencia variable, para aplicaciones en las cuales la potencia del ruido aditivo pudiera variar con el tiempo, tal como se presenta en ocasiones tanto en cancelación de ruido como de eco. El algoritmo propuesto se base en el hecho de que para evitar distorsión durante intervalos de tiempo en los cuales la potencia del ruido aditivo se incrementa, el factor de convergencia debe disminuir en la misma proporción en la que el ruido aditivo se incrementa. Simulaciones computacionales muestran que el algoritmo propuesto reduce de manera significativa, la distorsión que se presenta en los algoritmos convencionales cuando la potencia del ruido aditivo se incrementa sin reducir la capacidad del algoritmo para seguir variaciones en las características de las señales de referencia. Los resultados obtenidos indican que éste presenta mejores características que otros algoritmos propuestos hasta la fecha tales como: el NLMS [4.7], el VSLMS [4.10], el NACFLMS [4.12] y el VECLMS [4.13], CCLMS [4.14].

El algoritmo propuesto en este capítulo mejora considerablemente las características de convergencia del algoritmo NLMS en presencia del ruido aditivo, evitando degradación aun en presencia de altas cantidades de ruido aditivo. Sin embargo mantiene una complejidad computacional proporcional al orden del filtro, esto es  $O(N)$ . Aunque esta complejidad es aceptable para muchas aplicaciones existen algunas para las cuales esta complejidad podría ser muy elevada, lo cual hace necesario el desarrollo de algoritmos con una complejidad computacional menor. Con este fin, en el capítulo 5 se propone una estructura en el dominio de la frecuencia con una complejidad mucho menor para aplicaciones que requieren filtros adaptables de alto orden.

### **Algoritmo para Filtrado Adaptable en el Dominio de Frecuencia con Corto Tiempo de Retardo**

En el capítulo anterior se propuso un esquema que permite reducir en forma significativa el efecto del ruido aditivo en las características de convergencia del algoritmo NLMS. Sin embargo este algoritmo aun presenta una complejidad computacionales proporcional al orden del filtro. Aunque una complejidad de este orden podría ser aceptable para una gran cantidad de aplicaciones prácticas, ésta podría limitar el empleo de estos algoritmos en aplicaciones de tiempo real, cuando el orden del filtro es relativamente grande. Por su parte el procesamiento en bloque en el dominio de la frecuencia reduce considerablemente el número de operaciones requeridas por periodo de muestreo en aplicaciones de tiempo real. Sin embargo su velocidad de convergencia es, en general menor que la del los algoritmos en el dominio del tiempo que adaptan sus parámetros cada período de muestreo. Este hecho, así como el retardo de bloque introducido en la secuencia de salida del sistema, limita su uso en ciertas aplicaciones prácticas. Sin embargo, su potencial para manejar en tiempo real aplicaciones que requieren filtros de alto orden ha motivado la búsqueda de soluciones a los problemas antes mencionados. Esto ha dado como consecuencia la aparición de diversos esquemas que buscan incrementar la velocidad de convergencia de los algoritmos, basados en la búsqueda del gradiente, que operan en bloque o reducir su retardo de bloque. Sin embargo ninguno de los algoritmos para procesamiento en bloque publicados hasta la fecha, por otros autores, reduce el retardo de bloque al mismo tiempo que incrementa la velocidad de convergencia de los algoritmos convencionales para filtrado en bloque en el dominio de la frecuencia. Este capítulo propone un algoritmo adaptable para procesamiento en bloque el cual simultáneamente incrementa la velocidad de convergencia y reduce el tiempo de retardo introducido en la secuencia de salida del sistema, manteniendo al mismo tiempo una baja complejidad computacional. Se proporcionan además resultados obtenidos por simulación los cuales muestran las características del algoritmo propuesto.

## 5.1. Introducción.

El algoritmo LMS es ampliamente usado en diversas aplicaciones de filtrado adaptable debido a su robustez y simplicidad. Sin embargo, su velocidad de convergencia decae considerablemente cuando existe una gran dispersión en los valores propios de la matriz de autocorrelación de datos de entrada [5.1]-[5.6], [5.16], [5.22]. Esto se debe a que, a menos que la señal de entrada sea una secuencia de ruido blanco, en cuyo caso en el cual todos los valores propios son iguales, el gradiente no apunta al fondo de la superficie de error cuadrático medio. La desviación aumenta conforme la dispersión de los valores propios se incrementa, dando como resultado que la velocidad de convergencia del algoritmo LMS decrezca considerablemente.

A diferencia del algoritmo LMS el cual controla la búsqueda del vector de coeficientes óptimo a lo largo del gradiente de la superficie de error, el algoritmo RLS [5.2], [5.3], [5.16], descrito en el capítulo 3, usa el inverso de la matriz de autocorrelación para girar el vector de gradiente, en forma tal que éste apunte al mínimo de la superficie de error. Es sabido que este algoritmo proporciona la mayor velocidad de convergencia inicial. Sin embargo su complejidad computacional es extremadamente alta para muchas aplicaciones prácticas que requieren filtros de orden 50 o mayor. Últimamente se han desarrollado algoritmos RLS con mucho menor complejidad, como el llamado “Fast Kalman Algorithm”. Sin embargo éstos aun presentan problemas de estabilidad numérica [5.2], [5.3], [5.39].

Con el fin de obtener velocidades de convergencia relativamente altas con una complejidad computacional razonablemente bajas para aplicaciones en tiempo real, diversas variantes del algoritmo LMS han sido propuestas, algunas de las cuales se mencionan en el capítulo anterior, las cuales intentan mantener velocidades de convergencia relativamente altas, al mismo tiempo que buscan mantener el error residual razonablemente bajo mediante el uso de factores de convergencia variables con el tiempo [5.34]-[5.36].

Otra aproximación la cual también se han reportado avances importantes consiste en ortogonalizar las señales de entrada. Chang [5.20] mostró que la velocidad de convergencia del algoritmo LMS puede ser incrementada en forma significativa reduciendo la dispersión de los valores propios de la matriz de autocorrelación de la señal de entrada, esto es reduciendo la diferencia existente entre el mayor y de menor valor del propio de la matriz de autocorrelación de la señal de entrada. Chang propuso un igualador con rápida convergencia inicial en el cual ciertas señales de entrada son aproximadamente “ortogonalizadas” usando un operador fijo [5.20]. Makhoul [5.21] modificó esta estructura reemplazando el operador fijo de la estructura de Chang [5.20] por una estructura adaptable tipo lattice. Sin embargo, la complejidad computacional de esas alternativas para incrementar la velocidad de convergencia del algoritmo LMS aun es relativamente alta para ciertas aplicaciones prácticas.

Algoritmos para filtrado adaptable en el dominio de la frecuencia, basados en la estructura de Chang [5.20], también han sido propuestos, en los cuales una transformación lineal, usualmente una transformación ortogonal, tal como la transformada discreta de Fourier

(DFT), o la transformada discreta coseno (DCT) son usadas para, aproximadamente, ortogonalizar la señal de entrada [5.22]. Subsecuentemente, el coeficiente del  $i$ -ésimo componente de la transformada de la señal de entrada es adaptado usando un factor de convergencia constante dividido por el estimado de la potencia del  $i$ -ésimo componente de la transformación de la señal de entrada, calculada usando una ventana [5.22].

En paralelo con los esfuerzos encaminados a incrementar la velocidad de convergencia de los algoritmos adaptables basados en búsqueda del gradiente, se han propuesto diversas alternativas que buscan a reducir la complejidad computacional de algoritmos para filtrado adaptable de respuesta a impulso infinita (IIR) [5.7]-[5.13], así como algoritmos que emplean procesamiento en bloque [5.11], [5.17]-[5.19]. Los filtros adaptables IIR son potencialmente capaces de proporcionar funciones de transferencia con características similares a las de los filtros de respuesta a impulso finita (FIR) con un menor número de coeficientes [5.7]-[5.13]. Sin embargo pueden llegar a ser inestables durante la adaptación y su convergencia es en general más lenta que sus contraparte FIR, además de que en la mayoría de los casos convergen a un mínimo local de la superficie de error [5.7]-[5.13]. Por su parte los algoritmos para procesamiento en bloque proporcionan considerable reducción en la complejidad computacional comparada con los filtros que emplean el algoritmo LMS u otros similares, especialmente para aplicaciones que requieren filtros de orden relativamente alto, ya que permiten el uso de métodos muy eficientes basados en la FFT (Fast Fourier Transform) para el cálculo de la señal de salida del sistema, así como para la adaptación de los coeficientes del mismo [5.17]-[5.19]. Aunada a la reducción en la complejidad computacional, estos algoritmos también proporcionan un menor desajuste de la solución óptima, debido a que el gradiente empleado es un promedio de los gradientes instantáneos lo cual resulta en una mejor aproximación del gradiente real, siendo además incondicionalmente estables. Sin embargo, estos algoritmos modifican el vector de coeficientes del filtro al final de cada bloque en la misma forma que lo hacen el algoritmo LMS o sus variantes, mientras que su condición de estabilidad permanece la misma. Por lo tanto su velocidad de convergencia podría ser menor que la del algoritmo LMS, especialmente cuando los valores propios de la matriz de autocorrelación están altamente dispersos, esto es cuando la relación del máximo al mínimo valor propio es mucho mayor que 1 [5.2], [5.16]. Además el retardo de bloque introducido, el cual en la mayoría de los casos es igual al orden del filtro [5.2], [5.16]-[5.18]. Los problemas mencionados limitan el uso de los algoritmos de bloque en diversas aplicaciones prácticas [5.16].

Debido a su potencial para manejar en tiempo real aplicaciones que requieren filtros adaptables de alto orden, se ha llevado a cabo una intensa investigación para resolver algunos de los problemas mencionados, habiendo sido publicados durante los últimos años varias propuestas que mejoran las características del algoritmo original, conocido como el FLMS (Fast Least Mean Square). Así en [5.26]-[5.28] se propone el algoritmo FBAF (Frequency Bin Adaptive Filter) y en [5.30] se propone el algoritmo MDF (Multidelay Block Frequency Domain Adaptive Filter). Ambos, el FBAF y el MDF, son esquemas para filtrado adaptable en el dominio de la frecuencia, básicamente con la misma estructura, los cuales reducen el retardo de bloque introducido por el algoritmo FLMS [5.17]. Ambos incrementan además la velocidad de convergencia del algoritmo FLMS [5.17] al adaptar los coeficientes del filtros con mayor frecuencia que éste, aunque ninguno de ellos toma en cuenta las estadísticas de la señal de entrada para la selección del factor de convergencia.

Otro problema extremadamente importante es el incremento de la velocidad de convergencia de los algoritmos para procesamiento en bloque en el dominio de la frecuencia. Con esta finalidad, en las referencias [5.24] y [5.25] se propone el JALG (Jumping Algorithm) en el cual se deriva un conjunto de factores de convergencia cercanamente óptimos a partir de una transformación ortogonal. Este algoritmo incrementa considerablemente la velocidad de convergencia del algoritmo FLMS [5.17]. Sin embargo su retardo de bloque es igual al orden del filtro e introduce algunos requerimientos adicionales para su realización [5.34].

En este capítulo se presenta una revisión del filtrado adaptable en bloque y se propone un algoritmo para filtrado adaptable en el dominio de la frecuencia con corto retardo de bloque, el cual permite simultáneamente reducir el retardo de bloque introducido por los algoritmos FLMS [5.17], [5.18], [5.29] y JALG [5.24] y [5.25], e incrementar considerablemente la velocidad de convergencia de los algoritmos FBAF [5.26]-[5.28] y MDF [5.30]. El algoritmo propuesto reduce el retardo de bloque calculando la salida del sistema por medio de convoluciones cortas en el dominio de frecuencia en forma similar a la propuesta por [5.26]-[5.28] y [5.30], incrementando en forma considerable la velocidad de convergencia del sistema usando un algoritmo basado en la búsqueda del gradiente con un factor de convergencia cercanamente óptimo. Resultados obtenidos por simulación muestran que el algoritmo propuesto presenta velocidades de convergencia mayores que las proporcionadas por el JALG [5.24], [5.25] y velocidades de convergencia mucho mayores que las del algoritmo FLMS [5.17], [5.18], [5.29] con un menor retardo de bloque. El algoritmo propuesto proporciona además velocidades de convergencia mayores que las de los algoritmos FBAF [5.26]-[5.28] y MDF [5.30] con el mismo retardo de bloque y complejidad computacional similar.

## 5.2. Algoritmo LMS en Bloque (FLMS)

Esta sección presenta una revisión del algoritmo FLMS el cual servirá de base para el desarrollo del algoritmo propuesto, así como para revisar las principales características de otros conocidos algoritmos de bloque en el dominio de la frecuencia.

Considere un filtro adaptable FIR cuya señal de salida está dada por

$$y(n) = \mathbf{X}^T(n)\mathbf{W}(n) \tag{5.1}$$

donde

$$\mathbf{W}(n) = [w_0(n), w_1(n), w_2(n), \dots, w_{N-1}(n)]^T \tag{5.2}$$

es el vector de coeficientes y

$$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \dots, x(n-N+1)]^T \quad (5.3)$$

es el vector de entrada.

Asuma que la señal de referencia se encuentra dada por

$$d(n) = \mathbf{H}^T \mathbf{X}(n) + r(n) \quad (5.4)$$

donde  $\mathbf{H}$  es el vector de coeficientes del sistema a ser identificado y  $r(n)$  es una señal que no puede ser identificada por  $y(n)$ , ya sea por tratarse de una señal no correlacionada con la entrada,  $x(n)$ , o porque el orden del filtro adaptable no es lo suficientemente grande. Seguidamente, asuma que el vector de coeficientes  $\mathbf{W}(k)$  será modificado durante el bloque  $k+1$  usando un algoritmo basado en la búsqueda del gradiente, así el vector de coeficientes  $\mathbf{W}(k)$  estará dado por

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \beta \nabla(k) \quad (5.5)$$

donde  $\nabla(k)$  es el gradiente estimado durante el  $k$ ésimo bloque el cual esta dado por

$$\nabla(k) = \sum_{j=0}^{N-1} e(kN+j) \mathbf{X}(kN+j) \quad (5.6)$$

donde el error de salida  $e(kN+j)$  estará dado por

$$e(kN+j) = d(kN+j) - \mathbf{W}^T(k) \mathbf{X}(kN+j) \quad (5.7)$$

Usando la ecuación (5.5) el vector de coeficientes es modificado solamente una vez cada  $N$  períodos de muestreo. Esto es, una vez cada bloque donde la longitud del bloque es igual a  $N$ .

Debido a que los coeficientes del filtro permanecen constantes durante  $N$  períodos de muestreo, la convolución necesaria para obtener la salida del sistema, dada por (5.1), podría realizarse usando la transformada rápida de Fourier (FFT). Así usando el método de convolución rápida con 50% de traslape, al vector de coeficientes se le agregan  $N$  ceros y se calcula una FFT de  $2N$  puntos. Así, si denotamos  $\mathbf{C}(k)$  como el vector de coeficientes en el dominio de la frecuencia, éste estará dado por

$$\mathbf{C}(k) = \underset{\text{N zeroes}}{\text{FFT}[\mathbf{W}(k), 0, 0, 0, \dots, 0]}^T \quad (5.8)$$

Seguidamente definimos una matriz  $\mathbf{B}(k)$  como

$$\mathbf{B}(k) = \text{diag}\{\text{FFT}[x(kN - N), x(kN - N + 1), \dots, x(kN - 1), x(kN), \dots, x(kN + N - 1)]\} \quad (5.9)$$

Así la salida del filtro  $y(kN)$ ,  $y(kN+1)$ , ..... , estará dada como

$$y(kN + j) = \text{Ultimos N terminos de } \text{FFT}^{-1}[\mathbf{B}(k)\mathbf{C}(k)], \quad j = 0, 1, \dots, N-1 \quad (5.10)$$

La estimación del gradiente se puede llevar a cabo en forma similar reconociendo que la ecuación (5.6) denota la correlación entre el error de salida y el vector de entrada. Así, procediendo en forma similar a la usada para calcular  $\nabla(k)$ , definimos

$$\mathbf{E}(k) = \underset{\text{N zeros}}{\text{FFT}[0, 0, 0, \dots, 0, e(kN), e(kN + 1), \dots, e(kN + N - 1)]}^T \quad (5.11)$$

Así, usando (5.9) el gradiente estará dado por

$$\nabla(k) = \text{Pimeros N terminos de } \text{FFT}^{-1}[\mathbf{B}^*(k)\mathbf{E}(k)] \quad (5.12)$$

donde  $\mathbf{B}^*(k)$  denota el complejo conjugado de la matriz  $\mathbf{B}(k)$ .

### 5.2.1. Condición de Convergencia

Para obtener las condiciones de convergencia del algoritmo FLMS considere la expresión del gradiente dada por (5.6) y sustituya en ésta (5.7). Haciendo lo anterior obtenemos

$$\nabla(k) = \sum_{j=0}^{N-1} \mathbf{X}(kN + j)[d(kN + j) - \mathbf{X}^T(kN + j)\mathbf{W}(k)] \quad (5.13)$$

$$\nabla(k) = \sum_{j=0}^{N-1} d(kN + j)\mathbf{X}(kN + j) - \sum_{j=0}^{N-1} \mathbf{X}(kN + j)\mathbf{X}^T(kN + j)\mathbf{W}(k) \quad (5.14)$$

Asumiendo que el proceso es estacionario y que  $N$  es suficientemente grande, el gradiente estará aproximadamente dado por

$$\nabla(k) = N\mathbf{P} - NR\mathbf{W}(k) \quad (5.15)$$

donde

$$\mathbf{P} = \frac{1}{N} \sum_{j=0}^{N-1} d(kN + j)\mathbf{X}(kN + j) \quad (5.16)$$

y

$$\mathbf{R} = \frac{1}{N} \sum_{j=0}^{N-1} \mathbf{X}(kN + j)\mathbf{X}^T(kN + j) \quad (5.17)$$

son el vector de correlación cruzada y matriz de autocorrelación, respectivamente. Seguidamente, sustituyendo (5.15) en (5.5) obtenemos

$$\mathbf{W}(k + 1) = \mathbf{W}(k) + \beta N[\mathbf{P} - \mathbf{R}\mathbf{W}(k)] \quad (5.18)$$

la cual se puede reescribir como

$$\mathbf{W}(k + 1) = \mathbf{W}(k) + \beta N\mathbf{R}^{-1}[\mathbf{P} - \mathbf{R}\mathbf{W}(k)] \quad (5.19)$$

$$\mathbf{W}(k + 1) = \mathbf{W}(k) + \beta N\mathbf{R}[\mathbf{R}^{-1}\mathbf{P} - \mathbf{W}(k)] \quad (5.20)$$

$$\mathbf{W}(k + 1) = \mathbf{W}(k) + \beta N\mathbf{R}[\mathbf{W}_{op} - \mathbf{W}(k)] \quad (5.21)$$

donde

$$\mathbf{W}_{op} = \mathbf{R}^{-1}\mathbf{P} \quad (5.22)$$

es la solución óptima. Sustrayendo la ecuación (5.22) en ambos lados de la ecuación (5.21) obtenemos

$$\zeta(k + 1) = [\mathbf{I} - \beta N\mathbf{R}]\zeta(k) \quad (5.23)$$

donde

$$\zeta(k) = \mathbf{W}(k) - \mathbf{W}_{op} \quad (5.24)$$

ya que  $\mathbf{R}$  es la matriz de autocorrelación, la cual, en la mayoría de las aplicaciones prácticas, es una matriz positiva definida o semidefinida, esta puede ser descompuesta usando una transformación ortogonal como sigue

$$\mathbf{R} = \mathbf{K}^T \mathbf{Q} \mathbf{K} \quad (5.25)$$

donde  $\mathbf{K}$  es la matriz de transformación y

$$\mathbf{Q} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_N] \quad (5.26)$$

Así, sustituyendo (5.25) en (5.23) obtenemos

$$\zeta(k+1) = [\mathbf{I} - \beta \mathbf{N} \mathbf{K}^T \mathbf{Q} \mathbf{K}] \zeta(k) \quad (5.27)$$

$$\mathbf{K} \zeta(k+1) = \mathbf{K} [\mathbf{I} - \beta \mathbf{N} \mathbf{K}^T \mathbf{Q} \mathbf{K}] \zeta(k) \quad (5.28)$$

$$\mathbf{K} \zeta(k+1) = \mathbf{K} [\mathbf{K}^T \mathbf{K} - \beta \mathbf{N} \mathbf{K}^T \mathbf{Q} \mathbf{K}] \zeta(k) \quad (5.29)$$

$$\mathbf{K} \zeta(k+1) = \mathbf{K} \mathbf{K}^T [\mathbf{I} - \beta \mathbf{N} \mathbf{Q}] \mathbf{K} \zeta(k) \quad (5.30)$$

$$\mathbf{K} \zeta(k+1) = [\mathbf{I} - \beta \mathbf{N} \mathbf{Q}] \mathbf{K} \zeta(k) \quad (5.31)$$

$$\mathbf{V}(k+1) = [\mathbf{I} - \beta \mathbf{N} \mathbf{Q}] \mathbf{V}(k) \quad (5.32)$$

donde

$$\mathbf{V}(k) = \mathbf{K}\zeta(k) \quad (5.33)$$

Iterando (5.32) y reescribiendo el resultado en forma de matriz obtenemos

$$\begin{bmatrix} v_0(k+1) \\ v_1(k+1) \\ \vdots \\ v_{N-1}(k+1) \end{bmatrix} = \begin{bmatrix} (1-\beta N\lambda_1)^{k+1} & 0 & 0 & 0 \\ 0 & (1-\beta N\lambda_2)^{k+1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & (1-\beta N\lambda_N)^{k+1} \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix} \quad (5.34)$$

Así, el algoritmo convergerá, en promedio, a la solución óptima si

$$|1 - \beta N\lambda_i| < 1 \quad (5.35)$$

para toda  $i$ . Esto es

$$-1 < 1 - \beta N\lambda_{\max} < 1$$

$$-1 < 1 - \beta N\lambda_{\max}$$

$$-2 < -\beta N\lambda_{\max}$$

$$\frac{2}{N\lambda_{\max}} > \beta$$

$$1 - \beta N\lambda_{\max} < 1$$

$$\beta N\lambda_{\max} > 0$$

Así el algoritmo convergerá, en promedio, a la solución óptima si

$$0 < \beta < \frac{2}{N\lambda_{\max}} \quad (5.36)$$

y usando el hecho que

$$\text{TR}(\mathbf{R}) = N\overline{x^2(n)} > \lambda_{\max} \quad (5.37)$$

obtenemos

$$0 < \beta < \frac{2}{N^2\overline{x^2(n)}} \quad (5.38)$$

Esta condición de estabilidad, aunque mas restringida que la anterior es mucho mas usada, dada la dificultad que implica el estimar los valores propios, y a que el valor cuadrático medio de la señal de entrada, o sea su potencia, puede ser fácilmente estimada.

De la ecuación (5.34) se desprende que si tomamos el factor de convergencia  $\beta=(\lambda_r)^{-1}$  el factor  $v_r(n)$  se anulará en una iteración. Esto es adaptando  $\mathbf{W}(n)$  en la dirección del gradiente con un factor de convergencia igual a  $\beta=(\lambda_r)^{-1}$ , el vector  $\mathbf{W}(n)$  arribará a una posición en la superficie de error en la cual se anula una de las componentes ortogonales del vector de error. Así, adaptando consecutivamente el vector de coeficientes con factores de convergencia  $\{(\lambda_1)^{-1}, (\lambda_2)^{-1}, (\lambda_3)^{-1}, \dots, (\lambda_N)^{-1}\}$ , todas las componentes se anularán en  $N$  iteraciones lo cual permitirá que el algoritmo converja en  $N$  bloques. Este es el fundamento del algoritmo JALG (Jumping algorithm) [5.24]-[5.25], en el cual los valores propios de la matriz de autocorrelación es estiman usando una transformación ortogonal, usualmente la transformada discreta coseno, o la transformada discreta de Fourier [5.22]-[5.25]. El inverso de los valores propios es entonces usado, en forma secuencial, como factor de convergencia. Cada uno de ellos es usado, uno a la vez, durante un número determinado de iteraciones, empezando con el valor propio mayor [5.24], [5.25]. Este algoritmo incrementa considerablemente la velocidad de convergencia del algoritmo. El principal problema que presenta este algoritmo es el retardo de bloque el cual es igual al orden del filtro [5.24]-[5.25].

### 5.2.2. Número de Multiplicaciones por Período de Muestreo

Por cada bloque de  $N$  puntos el algoritmo FLMS requiere 5 FFT de  $2N$  datos reales cada una. Una FFT de  $2N$  puntos reales puede ser realizada usando una FFT de  $N$  datos y  $N$  multiplicaciones complejas. Una FFT de  $N$  puntos requiere aproximadamente  $(N/2)\log_2(N)-N$  multiplicaciones complejas. Por lo tanto, las 5 FFT de  $2N$  puntos en cada bloque pueden ser realizadas con aproximadamente  $(5N/2)\log_2(N)-5N$  multiplicaciones complejas. Adicionalmente para calcular  $\mathbf{B}(k)\mathbf{C}(k)$  y  $\mathbf{B}^*(k)\mathbf{E}(k)$ ,  $2N$  multiplicaciones complejas son requeridas y  $2N+1$  multiplicaciones reales para adaptar. Así asumiendo que una multiplicación compleja es equivalente a 4 multiplicaciones reales, el algoritmo FLMS requiere aproximadamente  $10\log_2(N)+8$  multiplicaciones reales por período de muestreo [5.2].

Por su parte el algoritmo JALG [5.24], [5.25] requiere, además una FFT de  $N$  puntos,  $N$  multiplicaciones complejas y  $2N+1$  multiplicaciones reales para estimación del factor de convergencia. Así el algoritmo JALG [5.24]-[5.25] requiere  $11\log_2(N) + 12$  multiplicaciones reales por período de muestreo.

### 5.3. Filtro Adaptable con Corto Retardo de Bloque (Propuesto)

Como se mencionó en las secciones anteriores, dos de los problemas principales que presentan los algoritmos de bloque basados en la búsqueda de gradiente, son su velocidad de convergencia y el retardo de bloque. Soluciones a estos dos problemas han sido propuestas separadamente, habiendo aparecido en la literatura propuestas que ya sea, incrementan la velocidad de convergencia del algoritmo FLMS [5.23]-[5.25] o reducen su retardo de bloque [5.26]-[5.28], sin incrementar considerablemente su complejidad computacional. Sin embargo ninguna de estas propuestas atacan de manera simultánea los dos problemas.

Esta sección presenta un algoritmo para filtrado adaptable el dominio de la frecuencia para procesamiento en bloque con corto retardo de bloque (SDFADF) el cual incrementa en forma significativa la velocidad de convergencia de los algoritmos FLMS [5.17], [5.18], FBAF [5.26]-[5.28] y MDF [5.30], reduciendo en forma simultánea el retardo de bloque introducido por algoritmos tales como el FLMS [5.17], [5.18] y JALG [5.24], [5.25]. El algoritmo propuesto reduce el retardo de bloque empleando convoluciones y correlaciones con corto tiempo de retardo tales como las empleadas por [5.26]-[5.28], [5.30], e incrementa la velocidad de convergencia del sistema empleando un algoritmo basado en la búsqueda del gradiente con un factor de convergencia cercanamente óptimo. Resultados obtenidos por simulación muestran que el algoritmo propuesto proporciona velocidades de convergencia mayores que las del JALG [5.24], [5.25] y mucho mayores que las del FLMS [5.17], [5.18], con un retardo de bloque mucho menor. Este también proporciona velocidades de convergencia mayores que el FBAF [5.26]-[5.28] y el MDF [5.30] con un retardo de bloque y complejidad computacional similares.

### 5.3.1 Estructura Propuesta.

Considere la señal de salida, de un filtro de respuesta a impulso finita la cual está dada por

$$\hat{y}(kL + j) = \mathbf{W}^T(k)\mathbf{X}(k) \quad (5.39)$$

donde  $j=0,1,\dots,L-1$ , y  $k$  es una constante positiva

$$\mathbf{W}(k) = [\mathbf{W}_0(k), \mathbf{W}_1(k), \dots, \mathbf{W}_{M-1}(k)]^T \quad (5.40)$$

$$\mathbf{W}_m(k) = [w_{mL}(k), w_{mL+1}(k), \dots, w_{mL+L-1}(k)]^T \quad (5.41)$$

Es el vector de coeficientes del filtro adaptable, y

$$\mathbf{X}(k) = [\mathbf{X}_0(k), \mathbf{X}_1(k), \dots, \mathbf{X}_{M-1}(k)]^T \quad (5.42)$$

$$\mathbf{X}_m(k) = [x((k - m)L + j), x((k - m)L + j - 1), \dots, x((k - m)L + j - L + 1)]^T \quad (5.43)$$

es el vector de entrada. Asumiendo que el vector de coeficientes,  $\mathbf{W}(k)$ , permanece constante, al menos durante  $L$  períodos de muestreo, la salida del sistema,  $y(kL+j)$ , se puede calcular usando el método de la convolución rápida con 50% de traslape. Así usando la propiedad de linealidad de la transformada de Fourier, después de algunas manipulaciones, de las ecuaciones (5.39)-(5.43) obtenemos [5.26]-[5.28]

$$\hat{y}(kL + j) = \text{Últimos } L \text{ terminos de } \text{FFT}^{-1} \left[ \sum_{m=0}^{M-1} \mathbf{B}_m(k) \mathbf{C}_m(k) \right], \quad j = 0, 1, \dots, M - 1 \quad (5.44)$$

Donde  $\text{FFT}^{-1}$  denota la transformada inversa de Fourier,

$$\mathbf{B}_m(k) = \text{diag} \left\{ \text{FFT} \left[ \mathbf{X}_{m-1}(k), \mathbf{X}_m(k) \right] \right\} \quad (5.45)$$

$$\mathbf{C}_m(k) = \text{FFT} \left[ \mathbf{W}_m(k), 0, 0, \dots, 0, 0 \right] \quad (5.46)$$

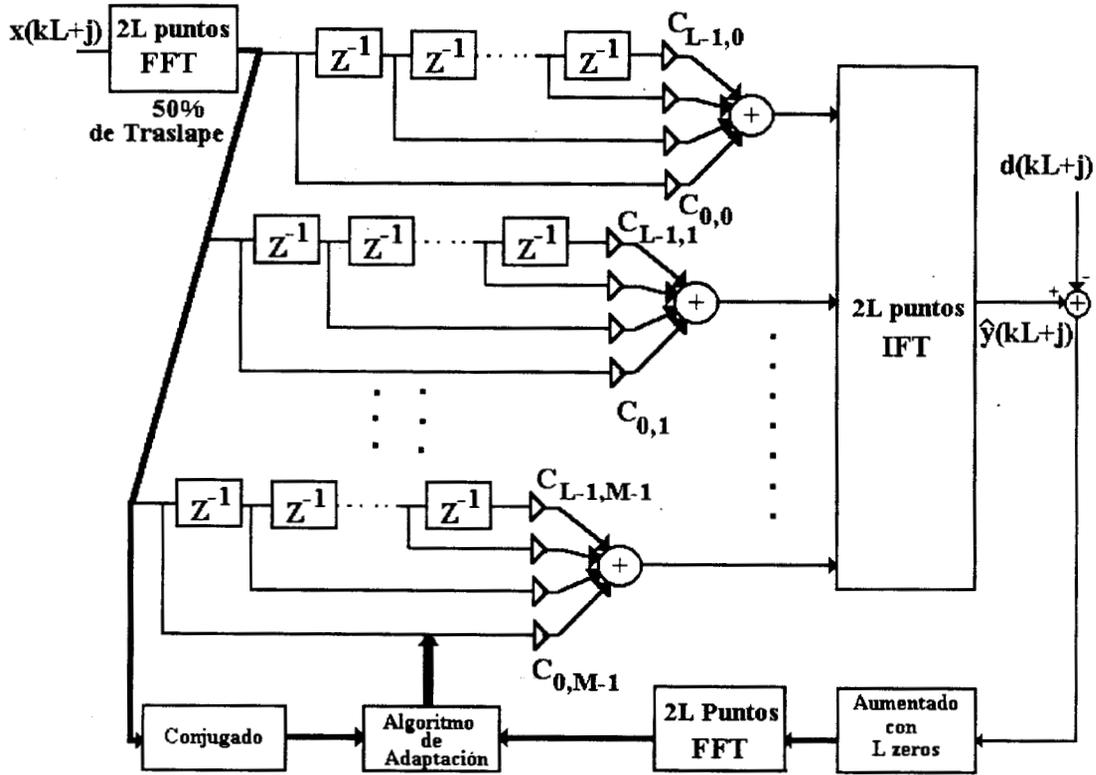


Figura 5.1. Estructura propuesta para filtrado en el dominio de frecuencia con corto retardo de bloque.

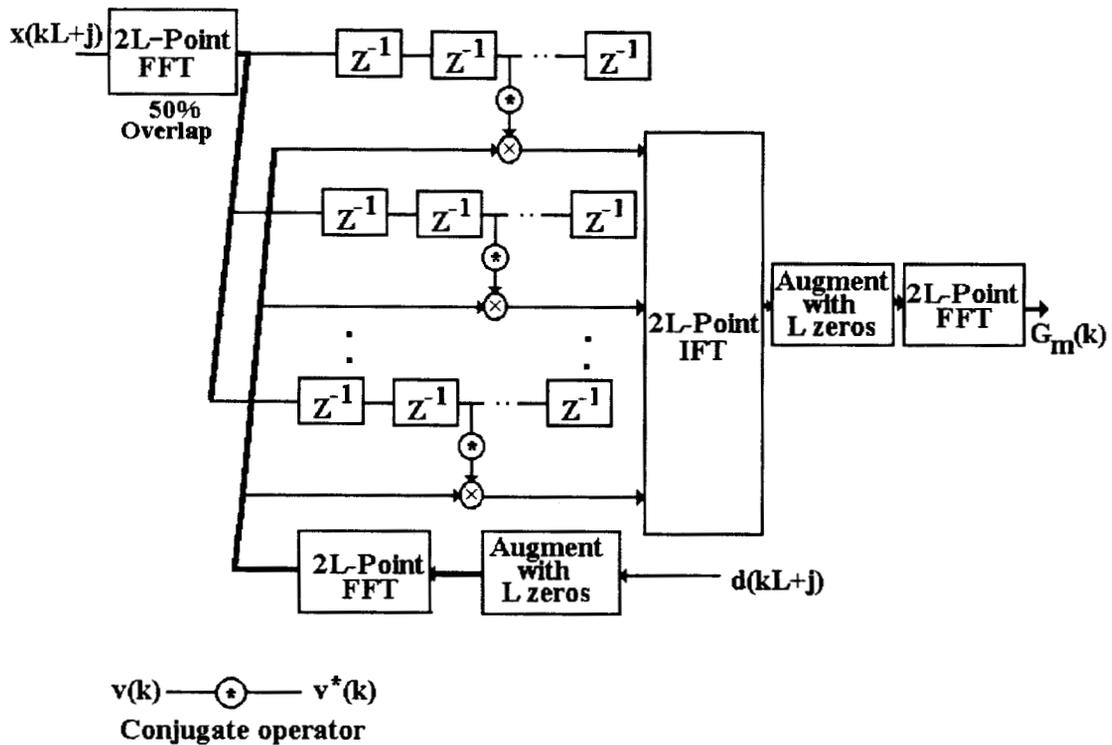


Figura 5.2. Sistema usado para la estimación del gradiente en la estructura propuesta.

y  $k$  es el índice del bloque. Las ecuaciones (5.44)-(5.46) llevan a la estructura para filtrado adaptable mostrada en la figura 5.1.

### 5.3.2 Algoritmo de Adaptación

Cuando un algoritmo de adaptación basado en la búsqueda del gradiente es empleado, el vector de coeficientes del filtro adaptable durante el  $k$ -ésimo bloque estará dado por

$$\mathbf{W}_m(k) = \mathbf{W}_m(k-1) + \beta \nabla_m(k-1) \quad (5.47)$$

donde

$$\nabla_m(k) = \frac{1}{L} \sum_{j=0}^{L-1} e(kL+j) \mathbf{X}_m(kL+j) \quad (5.48)$$

es el gradiente estimado durante el  $k$ ésimo bloque,  $e(kL+j)$  es el error de salida y  $\mathbf{X}_m(kL+j)$ , es el vector de entrada. La ecuación (5.48) denota la correlación cruzada entre el vector de entrada,  $\mathbf{X}_m(kL+j)$ , y el error de salida,  $e(kL+j)$ , el cual puede ser calculado usando el método conocido como “overlap-save” con 50% de traslape, como se muestra a continuación

$$\nabla_m(k) = \text{Primeros } L \text{ terminos de } FFT^{-1}[\mathbf{B}_m^*(k) \mathbf{E}(k)] \quad (5.49)$$

donde

$$\mathbf{E}(k) = [FFT[0,0,0\dots 0, e(kL), e(kL+1), e(kL+2), \dots, e(kL+L-1)]]^T \quad (5.50)$$

y  $\mathbf{B}_m^*(k)$  denota el complejo conjugado de la matriz  $\mathbf{B}_m(k)$  dada por. (5.45).

El vector de coeficientes expresado en el dominio de la frecuencia,  $\mathbf{C}_m(k)$  puede ser adaptado también en el dominio de la frecuencia, en lugar de adaptar el vector de coeficientes en el dominio del tiempo,  $\mathbf{W}_m(k)$  y posteriormente transformar este al dominio de la frecuencia para realizar las operaciones de filtrado. Así, si el vector de coeficientes en el dominio de la frecuencia,  $\mathbf{C}_m(k)$ , es adaptado usando un algoritmo basado en la búsqueda del gradiente,  $\mathbf{C}_m(k+1)$  estará dado por:

$$\mathbf{C}_m(k+1) = \mathbf{C}_m(k) + \beta \mathbf{G}_m(k), \quad m = 0, 1, 2, \dots, M-1 \quad (5.51)$$

donde

$$\mathbf{G}_m(k) = [\text{FFT}[\nabla_m(k), 0, 0, \dots, 0]]^T \quad (5.52)$$

La figura 5.2 muestra el diagrama del sistema usado para estimar  $\mathbf{G}_m(k)$

### 5.3.3 Factor de Convergencia

Con el fin de derivar el factor de convergencia del algoritmo propuesto, sustituya las ecuaciones (5.7), (5.43) y (5.48) en (5.47). Haciendo esto y asumiendo que la señal de entrada es un proceso ergódico estacionario en sentido amplio, de acuerdo a lo mostrado anteriormente, para longitudes de bloque suficientemente largas tenemos

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \beta L [\mathbf{P} - \mathbf{R}\mathbf{W}(k-1)] \quad (5.53)$$

donde

$$\mathbf{P} = \frac{1}{L} \sum_{j=0}^{L-1} d(kL+j) \mathbf{X}^T(kL+j) \quad (5.54)$$

es la correlación cruzada entre el vector de entrada y la señal de referencia, y

$$\mathbf{R} = \frac{1}{L} \sum_{j=0}^{L-1} \mathbf{X}(kL+j) \mathbf{X}^T(kL+j) \quad (5.55)$$

Tomando el valor esperado de (5.53) y sustrayendo en ambos lados la solución óptima dada por la ecuación de Wiener Hopf, después de algunas manipulaciones obtenemos [5.2], [5.4], [5.18] que el vector de coeficientes ortogonalizado,  $\mathbf{V}(k)$ , está dado por

$$\mathbf{V}(k) = (\mathbf{I} - \beta L \mathbf{Q})^k \mathbf{V}(0) \quad (5.56)$$

donde

$$\mathbf{V}(k) = \mathbf{T}(\mathbf{W}(k) - \mathbf{W}_{op}) \quad (5.57)$$

$$\mathbf{Q} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_N] \quad (5.58)$$

$\mathbf{T}$  es una transformación ortogonal, y  $\lambda_r$  es el r-ésimo valor propio de matriz  $\mathbf{R}$ . Así re escribiendo (5.56) en forma de matriz

$$\begin{bmatrix} v_0(k+1) \\ v_1(k+1) \\ \vdots \\ v_{N-1}(k+1) \end{bmatrix} = \begin{bmatrix} (1-\beta L\lambda_1)^{k+1} & 0 & 0 & \dots & 0 \\ 0 & (1-\beta L\lambda_2)^{k+1} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & (1-\beta L\lambda_N)^{k+1} \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix} \quad (5.59)$$

se tiene que  $\mathbf{V}(k)$  convergirá exponencialmente a cero y entonces el vector de coeficientes del filtro,  $\mathbf{W}(k)$  convergirá a la solución óptima sí:

$$0 < \beta < \frac{2}{L\lambda_{\max}} \quad (5.60)$$

La ecuación (5.60) determina el máximo valor de  $\beta$ , pero de mayor interés es el factor de convergencia que proporciona la velocidad de convergencia máxima de los algoritmos basados en la búsqueda del gradiente.

Para un factor de convergencia fijo, la velocidad de convergencia es dominada por modo convergente mas lento, el cual corresponde al mínimo valor propio de la matriz de autocorrelación, mientras que el modo más rápido corresponde al máximo valor propio. Así, ya que las curvas correspondientes a cualquier otro valor propio,  $(1-L\beta\lambda_i)$ , caen dentro de las curvas,  $(1-L\beta\lambda_{\min})$  y  $(1-L\beta\lambda_{\max})$ , el valor de  $\beta$  que proporciona la mayor velocidad

de convergencia, cuando un factor de convergencia fijo es usado, se puede obtener haciendo. [5.33].

$$|1 - L\beta\lambda_{\max}| = |1 - L\beta\lambda_{\min}| \quad (5.61)$$

La selección de cualquier otro valor de  $\beta$  resultará en una menor velocidad de convergencia del modo correspondiente ya sea al máximo valor propio o al mínimo [5.33]. Entonces de la ecuación (5.61) se tiene que el valor de  $\beta$  que proporciona la mayor velocidad de convergencia,  $\beta_f$ , estará dado por

$$\beta_f = \frac{2}{L(\lambda_{\max} + \lambda_{\min})} \quad (5.62)$$

Para esta selección de  $\beta$  los modos correspondientes a ambos, el máximo y el mínimo valor propio convergen a la misma velocidad. [5.33]

Estimar los valores propios de la matriz de autocorrelación requiere considerable esfuerzo computacional, debido a que la transformación óptima  $T$  requerida para calcular los valores propios de la matriz de autocorrelación depende de la señal de entrada y, salvo en algunos casos particulares, no existe un algoritmo rápido para calcular esta. Entonces el uso de ésta en muchas aplicaciones en tiempo real no es práctico [5.22], [5.24], [5.25] excepto para algunas señales particulares. Así para reducir el costo computacional debido al cálculo de los valores propios, la misma transformada discreta de Fourier (DFT) de la señal de entrada, la cual se emplea para el cálculo de la salida del filtro y el gradiente estimado, será empleada para la estimación de máximo y mínimo valor propio de la matriz de autocorrelación de la señal de entrada, requeridos para la estimación del factor de convergencia del algoritmo propuesto. Así usando la transformada discreta de Fourier (DFT) y asumiendo ergodicidad, los valores propios de la matriz de autocorrelación de la señal de entrada pueden ser estimados como [5.24], [5.25].

$$L\hat{Q} = \frac{L}{k} \sum_{i=1}^k |\mathbf{CX}(i)| |\mathbf{CX}(i)|^* \quad (5.63)$$

donde  $\mathbf{CX}(i)$  denota la transformada de Fourier de la señal de entrada  $x(kL+j)$  y  $*$  denota la operación complejo conjugado. Esto significa que el máximo y el mínimo valor propio requeridos para la estimación de factor de convergencia del algoritmo propuesto, corresponde al máximo y mínimo global, respectivamente, de la densidad de potencia espectral de la señal de entrada. Entonces, con el fin de usar las transformadas de Fourier ya calculadas para la estimación de los valores propios requeridos, podemos usar el método de los periodogramas promediados modificados [5.39] con 50% de traslape para estimar la

densidad de potencia espectral de la señal de entrada durante el k-ésimo bloque, y a partir de este los valores propios requeridos para la estimación de  $\beta_f$ . Entonces de (5.45) y (5.63) se tiene que la densidad de potencia espectral puede ser estimada de la siguiente manera.

$$L\hat{Q}(k) = (1 - \gamma)\hat{Q}(k - 1) + \gamma \left[ \sum_{m=0}^{M-1} \mathbf{B}_m(i)\mathbf{B}_m^*(i) \right] \quad (5.64)$$

donde  $\gamma^{-1}$  es aproximadamente igual al número de bloques usado para estimar la densidad de potencia espectral de la señal de entrada. Finalmente, si nosotros definimos

$$L\hat{Q}(k) = (\zeta_1, \zeta_2, \dots, \zeta_N) \quad (5.65)$$

de la ecuación (5.59) se obtiene que

$$\beta_f = \frac{2}{\max(\zeta_1, \zeta_2, \dots, \zeta_N) + \min(\zeta_1, \zeta_2, \dots, \zeta_N)} \quad (5.66)$$

donde  $\max(\zeta_1, \zeta_2, \dots, \zeta_N)$  denota el máximo valor de  $(\zeta_1, \zeta_2, \dots, \zeta_N)$  el cual será el valor estimado de  $L\lambda_{\max}$ , y  $\min(\zeta_1, \zeta_2, \dots, \zeta_N)$  denota el mínimo valor de  $(\zeta_1, \zeta_2, \dots, \zeta_N)$  el cual, por su parte, será el valor estimado de  $L\lambda_{\min}$ . Finalmente el algoritmo de adaptación propuesto estará dado por la ecuación (5.51) con  $\beta$  reemplazado por  $\beta_f$  ( ecuación (5.66)). La estimación del factor de convergencia se muestra en la figura 5.3.

#### 5.3.4. Número de Multiplicaciones por Período de Muestreo

Por cada bloque de  $L$  puntos, el algoritmo para procesamiento en bloque propuesto requiere

$$(4M + 6)\text{Log}_2 L + 8M \quad (5.67)$$

multiplicaciones reales por periodo de muestreo para filtrado y modificación de los coeficientes del filtro [5.26]-[5.28]. Adicionalmente éste requiere  $2M+2$  multiplicaciones reales por periodo de muestreo para llevar a cabo la estimación del factor de convergencia. Así el algoritmo propuesto requiere aproximadamente

$$(4M + 6)\text{Log}_2 L + 10M + 2 \quad (5.68)$$

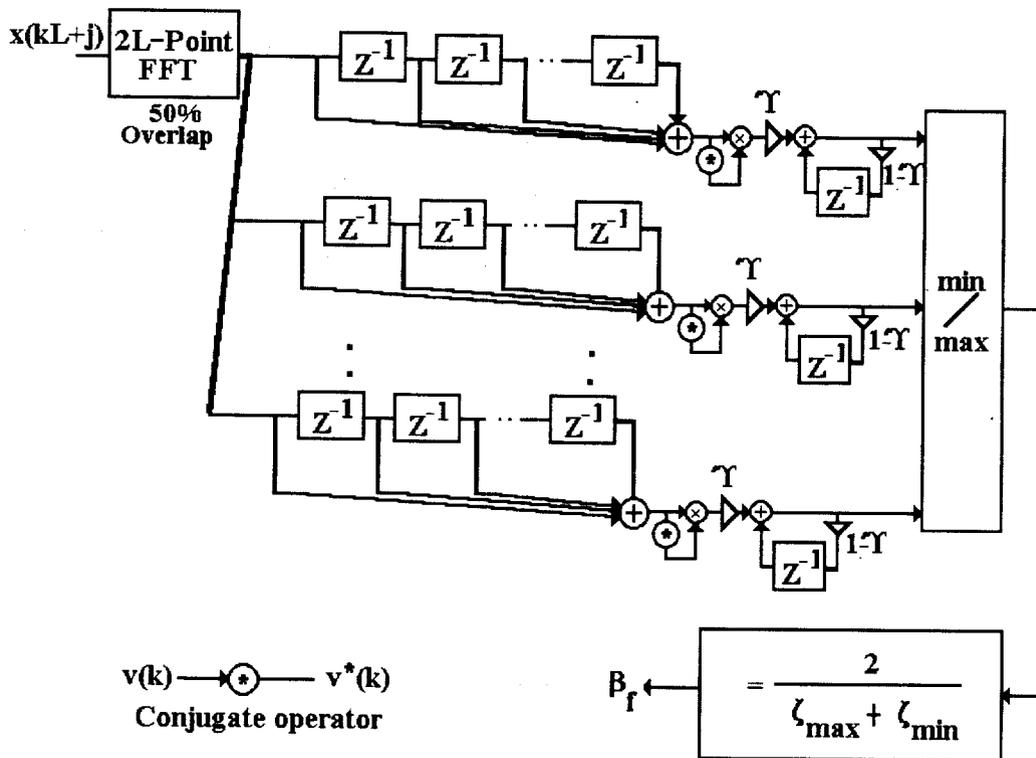


Figura 5.3. Estimación del factor de convergencia en la estructura propuesta.

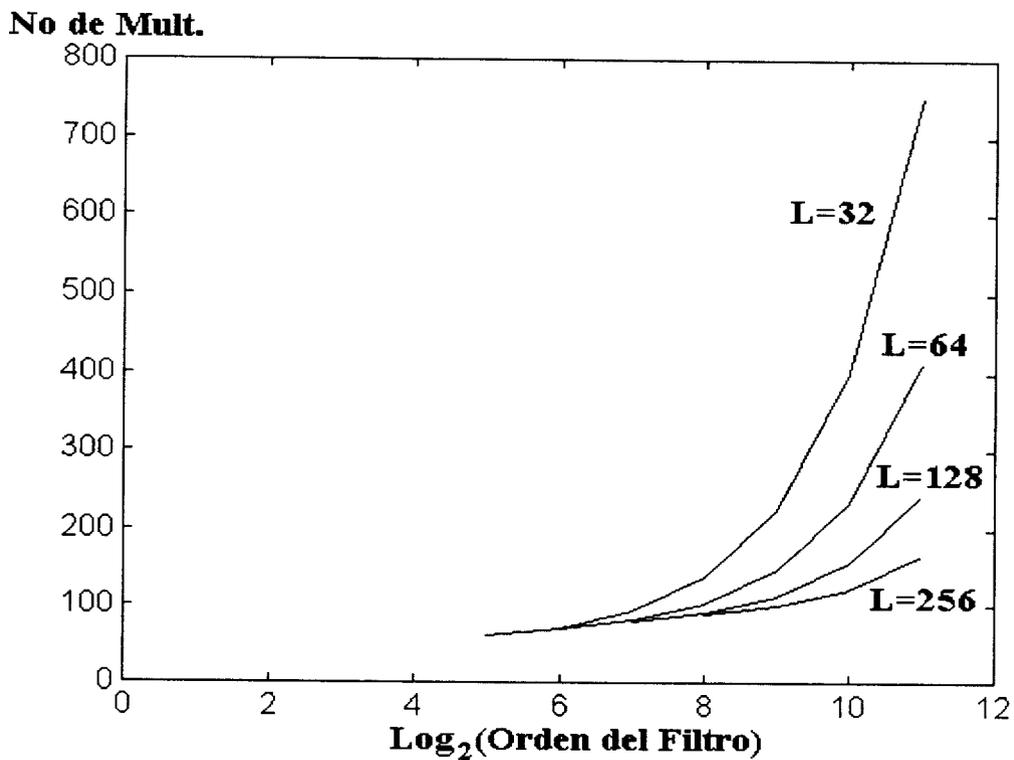


Figura 5.4. Complejidad computacional en relación al retardo de bloque, para filtros de varios órdenes.

multiplicaciones reales por período de muestreo, donde M es número de bloques. La figura 5.4 muestra el número de multiplicaciones para el mismo orden del filtro, y diferentes retardos de bloque L.

#### 5.4 Resultados Obtenidos por Simulación.

Los resultados obtenidos por simulación fueron generados usando una configuración de identificación de sistemas, aunque se espera que el algoritmo propuesto se comporte de manera satisfactoria en cualquier otra aplicación de filtrado adaptable. El criterio usado fue para evaluar las características de convergencia del algoritmo para filtrado adaptable propuesto es el error cuadrático medio normalizado, definido como

$$\text{MSE} = 10 \log_{10} \frac{E[(y(kL + j) - \hat{y}(kL + j))^2]}{E[y^2(kL + j)]} \quad (5.69)$$

donde  $y(kL+j)$  denota la salida del sistema desconocido. Esto es la señal de referencia en ausencia de ruido aditivo, y  $\hat{y}(kL+j)$  es la salida de filtro adaptable.

Las figuras 5.5 y 5.6 muestran las características de convergencia del algoritmo propuesto con cuatro diferentes retardos de bloque (N/M) cuando es requerida a identificar un sistema desconocido FIR de orden 128. La señal de entrada fue una secuencia de ruido blanco. La relación señal a ruido entre la señal de salida del sistema desconocido y el ruido aditivo es igual a 35 dB. La figura 5.7 muestra las características de convergencia del algoritmo propuesto con tres diferentes retardos de bloque (N/M) cuando es requerida a identificar un sistema desconocido FIR de orden 128. Aquí la señal de entrada fue un proceso autoregresivo de orden 10 con una función de autocorrelación similar a la de una señal de voz actual, y la relación señal a ruido entre la señal de salida del sistema desconocido y el ruido aditivo es igual a 35 dB. En las figuras 5.6 y 5.7 se muestran también la convergencia del algoritmo FLMS para su comparación. En los dos casos la potencia de la señal de entrada fue normalizada a uno. Las figuras 5.8 y 5.9 muestran el error cuadrático medio (MSE) obtenido, con tres diferentes retardos de bloque, cuando el algoritmo propuesto es requerido a identificar un sistema desconocido de orden 128. La señal de entrada fue una señal de voz actual con una relación señal a ruido de 40 dB. La figura 5.10 muestra el MSE obtenido, con dos diferentes retardos de bloque, cuando la estructura propuesta es requerida a identificar un sistema desconocido de orden 128. La señal de entrada fue una señal de voz actual con una relación señal a ruido de 40 dB. La potencia promedio de la señal de entrada se normalizó igual a uno. Nuevamente el MSE obtenido usando el algoritmo FLMS se muestra para su comparación. Esos resultados muestran que la velocidad de convergencia incrementa conforme el tamaño del bloque L disminuye. Esto

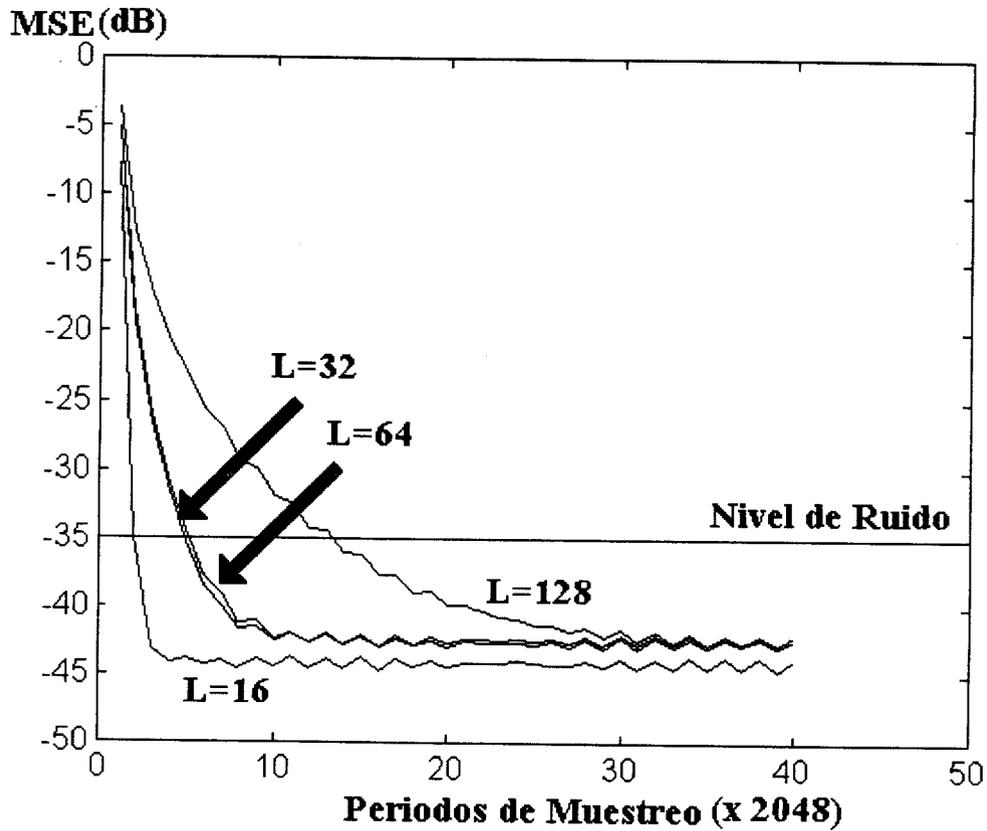


Figura 5.5. Convergencia del algoritmo propuesto con 4 diferentes retardos de bloque. La señal de entrada fue una secuencia de ruido blanco con potencia igual a uno.

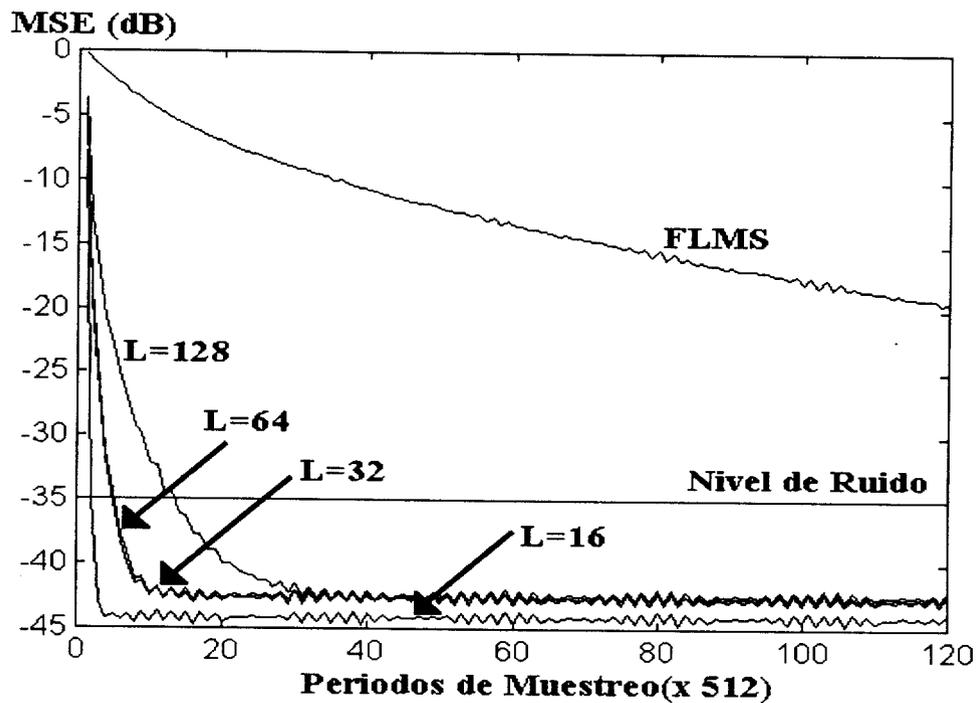


Figura 5.6. Convergencia del algoritmo propuesto con 4 diferentes retardos de bloque. La señal de entrada fue una secuencia de ruido blanco con potencia igual a uno. La convergencia del algoritmo convencional FLMS se muestra para comparación.

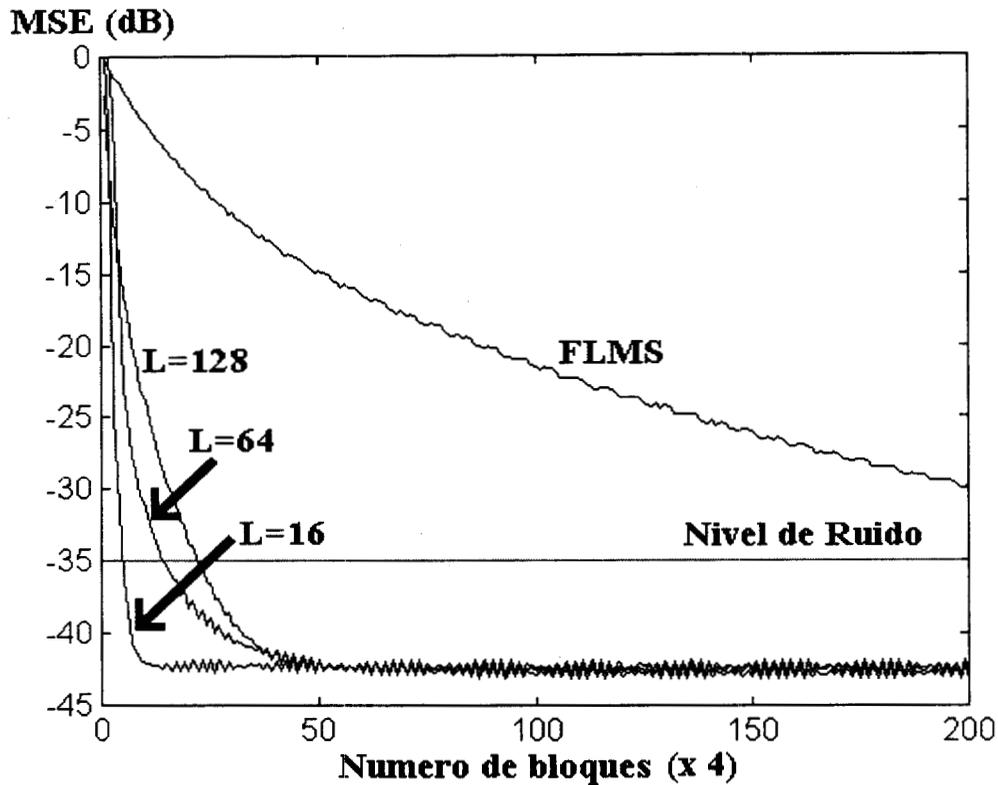


Figura 5.7. Convergencia del algoritmo propuesto con 4 diferentes retardos de bloque, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue una secuencia de ruido blanco con potencia igual a uno. La convergencia del algoritmo convencional FLMS se muestra para comparación.

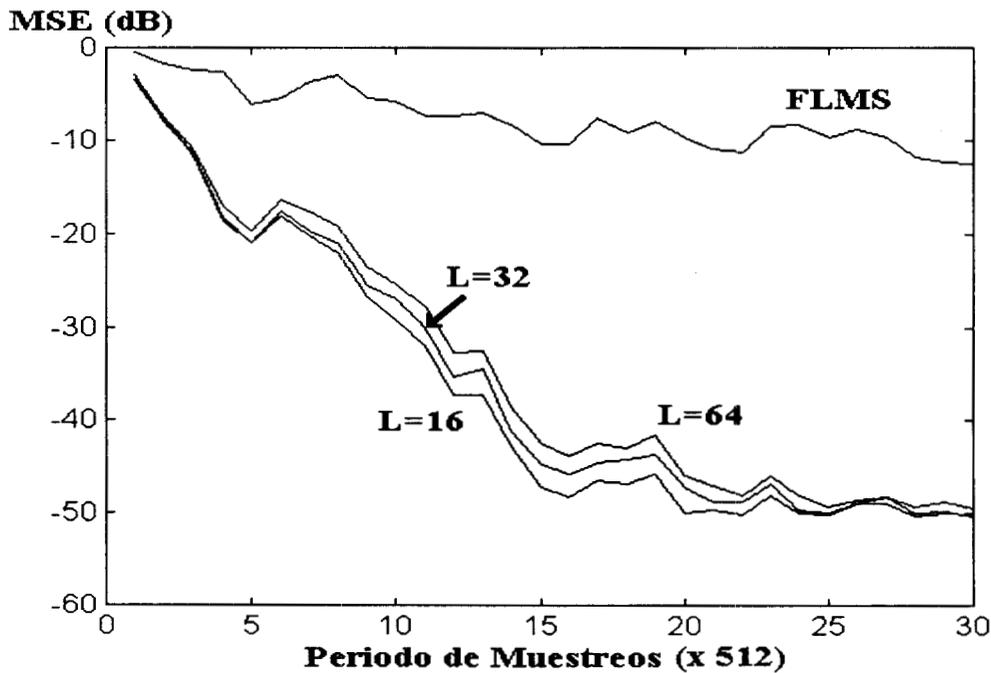


Figura 5.8. Convergencia del algoritmo propuesto con 3 diferentes retardos de bloque, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue una señal de voz actual con potencia promedio igual a uno. La convergencia del algoritmo convencional FLMS se muestra para comparación.

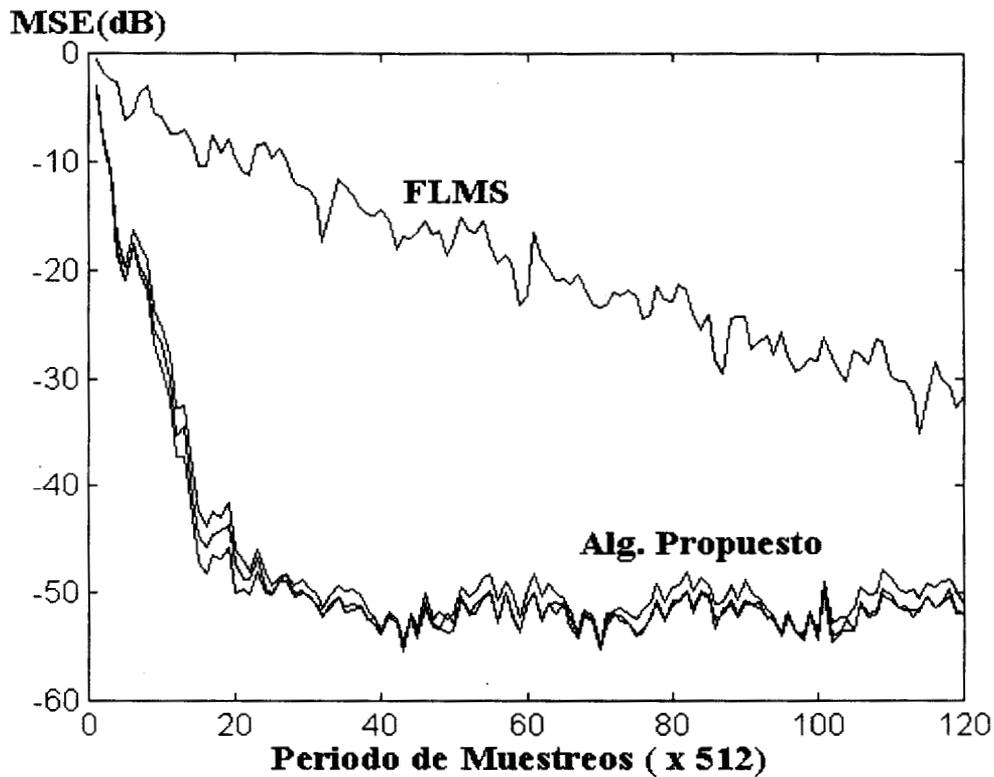


Figura 5.9. Convergencia del algoritmo propuesto con 3 diferentes retardos de bloque, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue una señal de voz actual con potencia promedio igual a uno. La convergencia del algoritmo convencional FLMS se muestra para comparación.

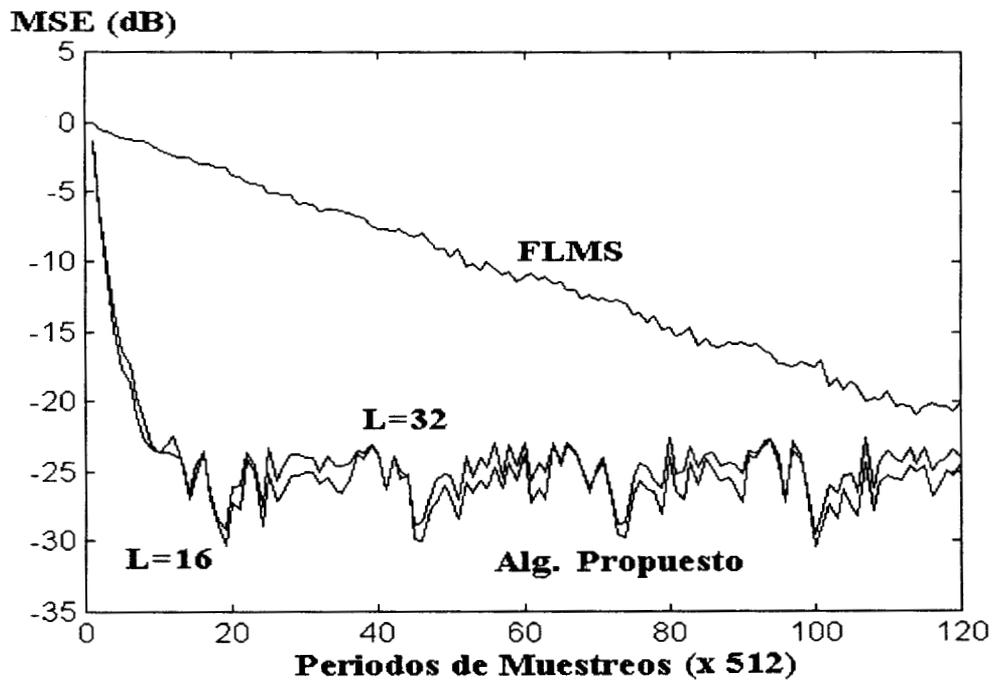


Figura 5.10. Convergencia del algoritmo propuesto con 2 diferentes retardos de bloque, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue una señal de voz actual con potencia promedio igual a uno. La convergencia del algoritmo convencional FLMS se muestra para comparación.

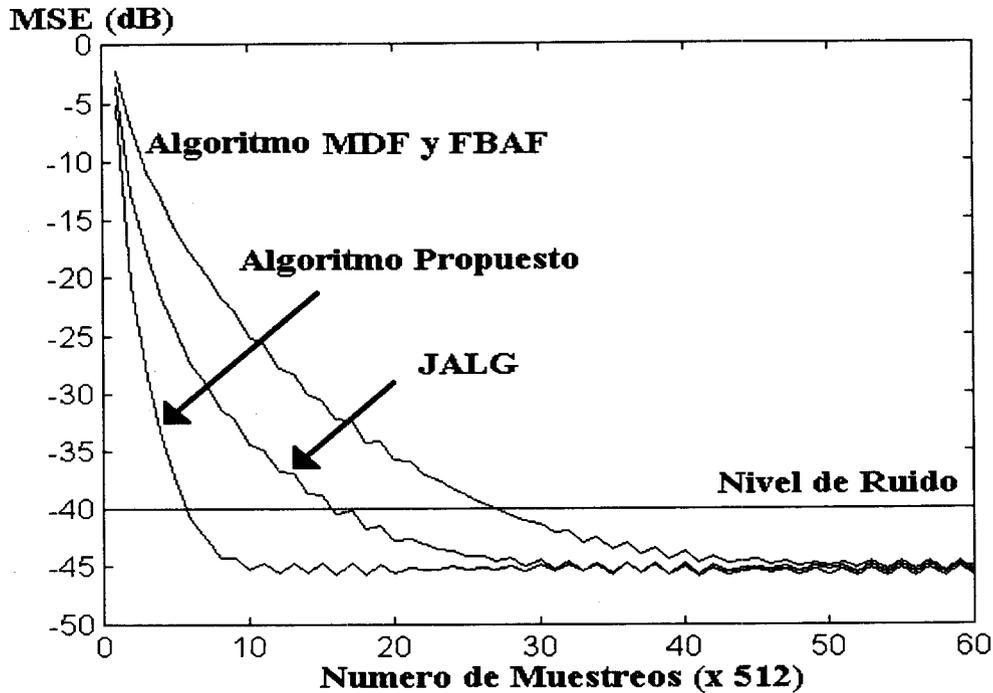


Figura 5.11. Convergencia del algoritmo propuesto con 32 muestras de retardo, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue una secuencia de ruido blanco con potencia igual a uno. La convergencia del algoritmo FBAF y MDF con el mismo retardo de bloque y el algoritmo JALG se muestra para comparación.

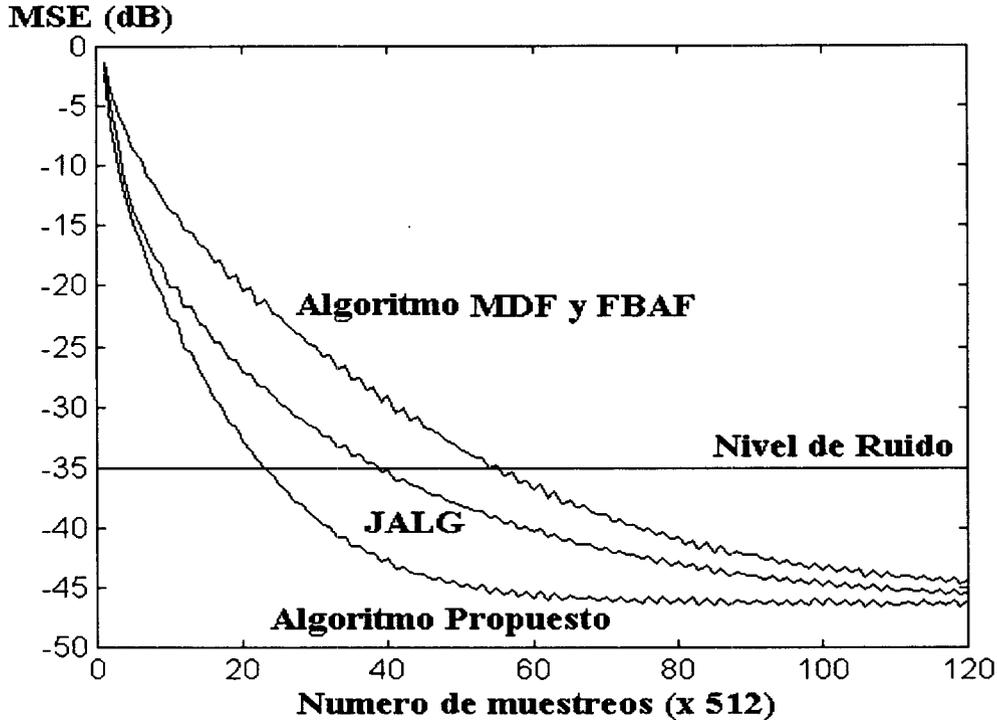


Figura 5.12. Convergencia del algoritmo propuesto con 32 muestras de retardos de bloque, cuando es requerido para identificar un sistema de orden 128. La señal de entrada fue un proceso AR con potencia igual a uno. La convergencia del algoritmo FBAF y del MDF con 32 muestras de retardo, y el algoritmo JALG con 128 se muestra para comparación.

se debe al hecho de que cuando el tamaño del bloque disminuye el vector de coeficientes del filtro se adapta con mayor frecuencia. La figura 5.11 muestra el error cuadrático medio, MSE, cuando el algoritmo propuesto, el MDF, el FBAF, y el JALG son requeridos a identificar un sistema desconocido de orden 128. Aquí  $M$  fue igual a 4, y  $L=32$  tanto para el algoritmo propuesto como para los algoritmos MDF y FBAF. La señal de entrada fue ruido blanco y la relación señal a ruido entre la señal de salida del sistema desconocido y el ruido aditivo fue aproximadamente 40 dB. La figura 5.12 muestra el MSE cuando el algoritmo propuesto, y los algoritmos MDF, FBAF y JALG son requeridos a identificar un sistema desconocido de orden 128. Aquí  $M=4$  y  $L=32$  para el algoritmo propuesto, y los algoritmos MDF y FBAF. La señal de entrada fue un proceso autoregresivo de orden 15 con una densidad de potencia espectral muy similar a la de una señal de voz actual. La relación señal a ruido entre la señal de salida del filtro y el ruido aditivo fue igual a 35 dB. En ambos casos la potencia de la señal de entrada fue igual a uno. Los resultados obtenidos por simulación muestran que el algoritmo propuesto proporciona velocidades de convergencia mayores que las proporcionadas por los algoritmos MDF y FBAF con el mismo retardo de bloque y similar complejidad computacional, y velocidades de convergencia mayores que el algoritmo JALG con un retardo de bloque menor, aunque en este caso la complejidad computacional es mayor.

## 5.5 Conclusiones

En este capítulo se propuso un algoritmo para procesamiento adaptable en bloque en el dominio de la frecuencia. Para obtener un menor retardo de bloque la salida del filtro fue calculada como la suma de  $M$  convoluciones parciales [5.26]-[5.28] y [5.30]. Cada una de esas convoluciones fue calculada usando el método conocido como "Overlap-Save" con 50% de traslape para reducir complejidad computacional. Un factor de convergencia cercanamente óptimo para el algoritmo basado en la búsqueda del gradiente es estimado usando el método de los periodogramas modificados promediados [5.39] con 50% de traslape. Este hecho permite estimar el factor de convergencia usando los mismos datos obtenidos de las transformaciones efectuadas para el cálculo de la salida del sistema.

Las simulaciones computacionales muestran que el esquema propuesto proporciona mucha mayor velocidad de convergencia que el bien conocido algoritmo FLMS, descrito en secciones anteriores con un retardo de bloque mucho menor, aunque su complejidad computacional es un tanto mayor, especialmente cuando el algoritmo propuesto opera con bloque relativamente cortos. Este también proporciona una velocidad de convergencia mayor que los algoritmos MDF [5.30] y FBAF [5.26]-[5.28] con el mismo retardo de bloque y similar complejidad computacional. El algoritmo propuesto también proporciona una velocidad de convergencia mayor que el algoritmo JALG [5.24], [5.25] cuando  $M>1$ , debido a que el vector de coeficientes es adaptado mas frecuentemente. Sin embargo en esta situación la complejidad del algoritmo propuesto es un tanto mayor. Resultados obtenidos por simulación muestran que también que la velocidad de convergencia del algoritmo propuesto se incrementa cuando el retardo de bloque disminuye, debido a que en

esta situación el vector de coeficientes es adaptado mas frecuentemente. Sin embargo esta reducción en el retardo de bloque e incremento en la velocidad de convergencia resulta en una mayor complejidad computacional. Así hay un compromiso entre la velocidad de convergencia, el retardo de bloque y la complejidad computacional del algoritmo propuesto.

De lo anterior se desprende que los algoritmos propuestos en los capítulos 4 y 5 resuelven dos problemas importantes en filtrado adaptable basado en la búsqueda del gradiente. La sensibilidad al ruido aditivo y la reducción en la complejidad computacional que permita la realización de filtros adaptables de alto orden en tiempo real. Sin embargo existen otras aplicaciones en las cuales además de una alta inmunidad al ruido aditivo se requieren altas velocidades de convergencia para manejar sistemas variables con el tiempo. Para éste tipo de aplicaciones, el capítulo 6 propone una estructura para filtrado adaptable del tipo RLS, la cual no presenta problemas de estabilidad numérica, pudiendo al mismo tiempo ser realizada tanto en forma digital como en forma analógica.

## CAPITULO 6

### **Algoritmos Adaptables para Filtrado y Predicción Usando una Red de Hopfield Modificada**

En los capítulos anteriores se propusieron soluciones a dos problemas fundamentales del filtrado adaptable, basado en la búsqueda del gradiente, como lo son: El adecuado manejo de las variaciones en la relación señal a ruido existente en diversas aplicaciones, en forma tal que se evite la degradación en el funcionamiento del sistema, y el manejo en tiempo real de filtros de alto orden. Como se muestra en los capítulos 4 y 5, los algoritmos propuestos se comportan mejor que los anteriormente existentes cuando operan en las condiciones para las que fueron concebidos. Sin embargo, existen aplicaciones que requieren sistemas con características diferentes a las de los algoritmos desarrollados anteriormente, tales como: Mayor velocidad de convergencia y alta inmunidad al ruido aditivo, menor consumo de potencia y tamaño, etc. Esto sugiere la necesidad de desarrollar algoritmos con estas características, siendo los filtros adaptables analógicos excelentes candidatos para este fin.

Durante los últimos años, los filtros adaptables analógicos han sido objeto de una activa investigación debido a su habilidad para manejar en tiempo real señales de muy altas frecuencias, siendo su tamaño y consumo de potencia mucho menor que sus contrapartes digitales. Durante este tiempo, varias estructuras analógicas para filtrado adaptable han sido propuestas en la literatura, casi todas ellas empleando versiones analógicas del bien conocido algoritmo LMS (Least Mean Square). Sin embargo las versiones analógicas del algoritmo LMS presentan las mismas limitaciones que sus contrapartes digitales cuando operan con señales cuya relación señal a ruido (SNR) es baja. Este hecho sugiere la necesidad de desarrollar versiones analógicas de algoritmo RLS el cual, como se sabe, tiene muy baja sensibilidad al ruido aditivo lo cual lo hace muy atractivo para operar con señales cuya SNR es baja. Sin embargo una relación directa del algoritmo RLS en forma analógica sería difícil ya que esto involucraría la inversión de una matriz.

Para resolver este problema se propone una estructura para filtrado adaptable analógico en el cual la adaptación del vector de coeficientes se lleva a cabo en tiempo continuo usando una red modificada de Hopfield, con una función de activación lineal. La estructura propuesta no presenta problemas de estabilidad, ya que no requiere ninguna inversión de matriz explícita para realizar el algoritmo RLS en tiempo continuo. Ésta fue evaluada por medio de simulaciones computacionales, operando tanto como identificador de sistemas que como predictor lineal. Los resultados obtenidos muestran que el sistema propuesto operando como identificador de sistemas en condiciones estacionarias, presenta una velocidad de convergencia inicial similar a la proporcionada por el algoritmo RLS

convencional con un menor desajuste que el presentado por el algoritmo RLS convencional debido al sobre muestreo introducido en la realización analógica, así como mejores características de convergencia, cuando el sistema propuesto opera en condiciones en las cuales las señales de entrada y referencia son no estacionarias, especialmente cuando las estadísticas de las señales involucradas varían rápidamente con el tiempo. Los resultados obtenidos por simulación muestran además que cuando el sistema opera con una configuración de predictor lineal, éste presenta un funcionamiento muy similar al del algoritmo RLS convencional, operando ambos bajo las mismas condiciones. De los resultados anteriores se concluye que la estructura propuesta presenta un funcionamiento altamente satisfactorio cuando opera en ambientes tanto estacionarios como no estacionarios, y debido a su realización analógica, parece ser una buena alternativa a los filtros adaptables digitales, en aplicaciones cuyas señales de entrada y referencia tienen componentes de alta frecuencia.

## **6.1. Introducción**

Los filtros adaptables FIR han sido parte fundamental en la solución de diversos problemas prácticos, tales como cancelación de eco [6.1], [6.9], [6.10], cancelación de ruido [6.1], aclaramiento de líneas [6.1], [6.29], codificación de voz y de audio [6.1], [6.29], deconvolución predictiva en exploración sísmica [6.1], e igualación adaptable para canales de comunicación digital degradados por interferencia intersímbolos, [6.1]-[6.3], [6.11], [6.20], [6.29], etc. Su buen funcionamiento en la mayoría de las aplicaciones anteriores es el resultado de una investigación muy activa que se ha realizado en este campo durante los últimos 20 años. Este hecho, junto con el avance de tecnología digital ha permitido el desarrollo de diversos algoritmos adaptables muy eficientes, así como sofisticados, para aplicaciones de filtrado adaptable digital, la mayoría de las cuales emplean estructuras transversales de respuesta a impulso finita, FIR, con algoritmos de adaptación ya sea del tipo LMS o RLS [6.1]-[6.8]. Sin embargo en los algoritmos del tipo LMS, la velocidad de convergencia es en la mayoría de los casos baja, presentando además una alta sensibilidad al ruido aditivo lo cual limita su uso en la solución de diversos problemas prácticos [6.1]. Por otro lado, el algoritmo de RLS tiene una alta velocidad de convergencia inicial y muy baja sensibilidad al ruido aditivo [6.1]. Sin embargo éste presenta una alta complejidad computacional, lo cual limita su aplicación cuando el orden de los filtros es relativamente alto, o cuando se tiene un poder de computo limitado, además, las versiones de este algoritmo con baja complejidad computacional, tal como el algoritmo "Fast Kalman", presentan en general problemas de estabilidad numérica [6.1], [6.11].

Además de los problemas mencionados anteriormente, los filtros digitales transversales FIR presentan algunos otros problemas cuando estos son requeridos a manejar, en tiempo real, frecuencias mayores que aquellas en el rango de audio, o cuando se requieren filtros de orden relativamente alto, debido a la limitación impuesta por la velocidad de operación de los procesadores digitales. Este problema se presenta aunque se usen en muchos casos sistemas con procesadores múltiples. Los problemas anteriores han sido abordados de diversas maneras, habiendo sido propuestos diferentes tipos de estructuras y algoritmos de

adaptación, tales como estructuras de filtros adaptables IIR [6.12]-[6.14], algoritmos adaptables en el dominio de la frecuencia para filtrado en bloque [6.6]-[6.8], algoritmos para filtrado adaptable en sub bandas [6.9].[6.10], y filtros adaptables analógicos [6.15]-[6.20]. Todos de ellos con características deseables, sin embargo todos ellos presentan también ciertos problemas que aun deben resolverse.

Recientemente el interés en los filtros adaptables analógicos han aumentado, debido a que ellos pueden manejar, fácilmente, señales de muy alta frecuencia. Éstos son de tamaño muy pequeño, y su consumo de potencia es muy bajo comparando con sus contra partes digitales [6.15]-[6.20], aunque en general su precisión es menor.

La mayoría de las estructuras de filtros adaptables analógicos propuestas hasta la fecha, usan la versión de tiempo continuo del algoritmo de LMS para adaptar el vector de coeficientes del filtro [6.15], [6.16], [6.18], y por lo mismo presentan problemas similares que sus contra partes digitales, cuando operan con señales cuyas relaciones señal a ruido son bajas [6.1]. Este hecho sugiere la necesidad de desarrollar estructuras de filtros adaptables analógicos del tipo RLS.

Por otra parte, las redes neuronales de Hopfield [6.21]-[6.26], las cuales se pueden realizar en forma analógica, se han usado con éxito para resolver varios problemas de optimización. Este hecho sugiere que la red Neuronal de Hopfield, o alguna modificación de esta puede ser usada para realizar un algoritmo adaptable analógico del tipo RLS, sin necesidad de una explícita inversión de matriz, evitando así potenciales problemas de inestabilidad numérica.

En este capítulo se propone una estructura de filtro adaptable analógico, en la cual, la estimación de los valores óptimos de los coeficientes se efectúa de manera analógica usando una red completamente conectada, la cual es una versión modificada de la red de Hopfield . La estructura propuesta es evaluada, por medio de simulación computacional, operando tanto en una configuración de identificador de sistemas, como en una configuración de predictor lineal. Los resultados obtenidos de simulaciones muestran las características deseables de la estructura propuesta.

## 6.2. Estructura Propuesta

Considere el filtro adaptable analógico mostrado en la figura 6.1, cuya señal de salida  $y(t)$  esta dada por

$$y(t) = \sum_{k=0}^{N-1} w_k x_k(t) \quad (6.1)$$

donde  $x_k(t)$  es la señal de salida del sistema que tiene una función de transferencia dada por

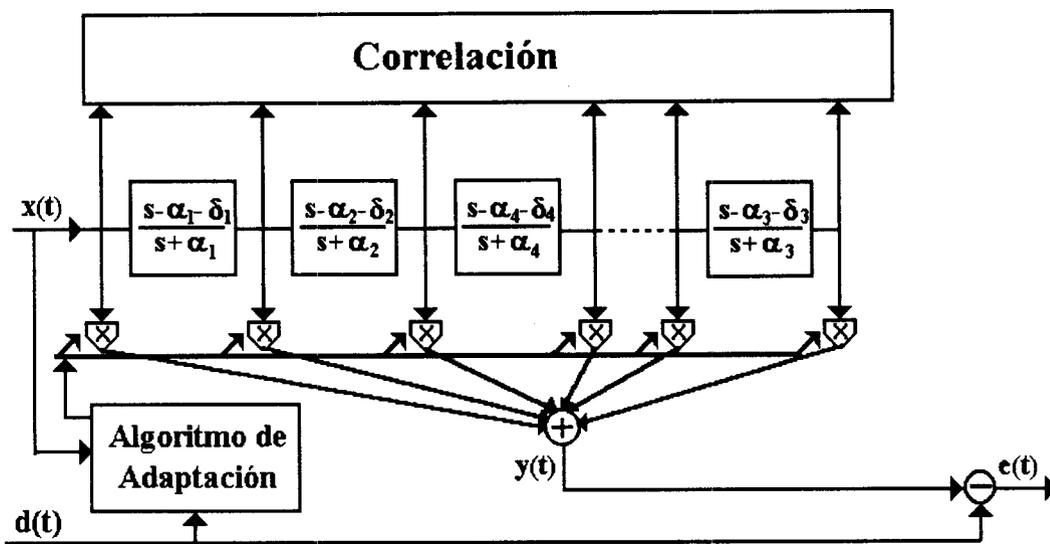


Figura 6.1. Estructura propuesta.

$$H_k(s) = \prod_{i=1}^k \frac{C_i [s - \alpha_i - \delta_i]}{s + \alpha_i} \quad (6.2)$$

aquí  $\delta_i$  es cierta desviación de la condición de pasa todo ideal,  $C_i$  es una constante positiva y  $w_k$  es el k-esimo coeficiente del filtro adaptable, el cual se estima de manera que el valor cuadrático medio de la señal de error sea mínimo, donde

$$e(t) = d(t) - y(t) \quad (6.3)$$

es la señal de error, y  $d(t)$  es señal de referencia. Entonces de las ecuaciones (6.1) y (6.3), el valor cuadrático medio de la señal de error esta dado por

$$E \left[ \left( d(t) - \sum_{k=0}^{N-1} w_k x_k(t) \right)^2 \right] \quad (6.4)$$

Finalmente, usando el principio de ortogonalidad en estimación de mínimos cuadrados, después de algunas manipulaciones se obtiene que, el vector de coeficientes óptimos esta dado por [6.1],[6.2]

$$W = \Phi_{xx}^{-1} \Phi_{xd} \quad (6.5)$$

donde  $W$  es el vector de coeficientes óptimo, cuyo k-esimo elemento es  $w_k$ ,  $\Phi_{xd}$  es el vector de correlación entre la señal de salida de la sección de pasa todo y la señal de referencia, cuyo j-esimo elemento esta dado por

$$\Phi_{xd}(j) = E[d(t)x_j(t)] \quad (6.6)$$

y  $\Phi_{xx}$  es la matriz de correlación de las señales de salida de la sección de pasa todo, cuyo (j,k)-esimo elemento esta dado por

$$\Phi_{xx}(j,k) = E[x_j(t)x_k(t)] \quad (6.7)$$

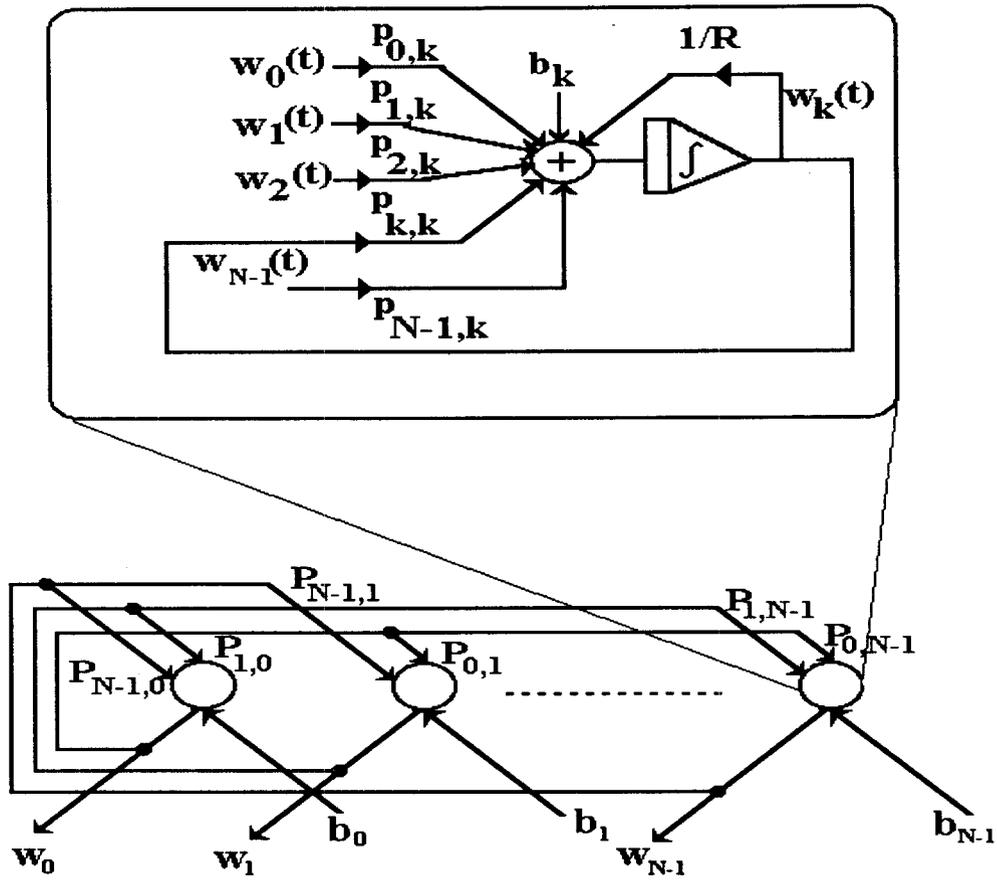


Figura 6.2(a). Red de Hopfield modificada con función de activación lineal.

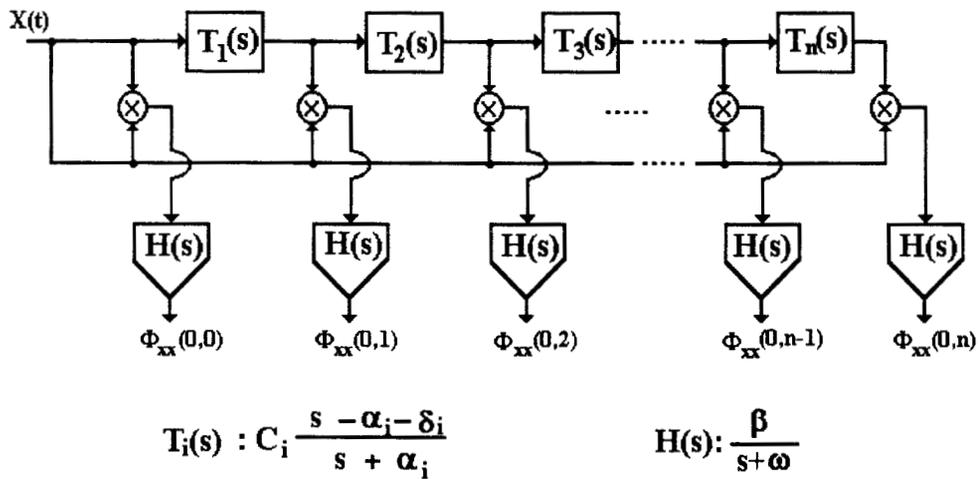


Figura 6. 2(b). Estimación de la secuencia de autocorrelación de la señal de entrada  $x(t)$ .

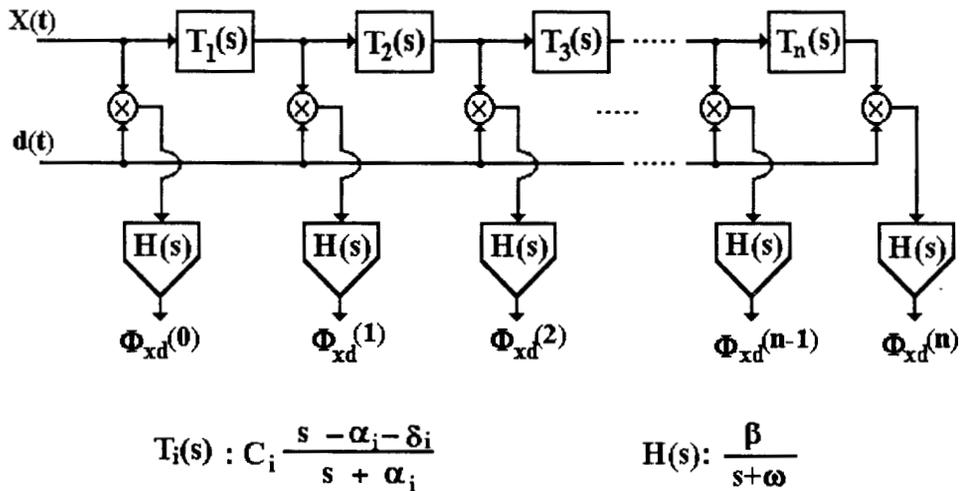


Figura 6. 2(c). Estimación de la secuencia de correlación cruzada entre la señal de entrada  $x(t)$  y la señal de referencia  $d(t)$ .

Para obtener un algoritmo adaptable para estimación del vector de coeficientes del sistema, considere la respuesta de la red Hopfield en tiempo continuo con una función de activación lineal como se muestra en la figura 6.2a. La señal de salida de una red modificada de Hopfield con estas características estará dada por [6.17], [6.21]-[6.27].

$$c \frac{dw_k(t)}{dt} = -\frac{1}{R} w_k(t) + \sum_{m=0}^{N-1} p_{m,k} w_m(t) + b_k, \quad k = 1, 2, \dots, N-1 \quad (6.8)$$

donde  $w_k(t)$  es la salida del k-esimo nodo, R y c son constantes reales positivas  $p_{m,k}$  es el peso asignado a la conexión que va desde el m-esimo nodo hasta k-esimo nodo y  $b_k$  es un factor real.

Tomando la transformada de Laplace de la ecuación (6.8), obtenemos

$$SW(S) + \frac{1}{RC} W(S) - \frac{1}{C} PW(S) = \frac{1}{C} B \quad (6.9)$$

Seguidamente, usando el teorema del Valor Final, después de un intervalo de tiempo suficientemente largo, la salida de red de Hopfield modificada estará dada por [6.17]

$$W(\infty) = [I - RP]^{-1} RB \quad (6.10)$$

Finalmente de las ecuaciones (6.5) y (6.10) se tiene que, la señal de salida de la red de Hopfield modificada convergirá a la solución óptima dada por la ecuación de Wiener-Hopf (6.5) si

$$\Phi_{xx} = (I - RP) \quad (6.11)$$

y

$$\Phi_{xd} = RB \quad (6.12)$$

Así, de las ecuaciones (6.10)-(6.12) se obtiene que, después de que se ha obtenido la convergencia, la red de Hopfield modificada proporciona la solución óptima de la ecuación

de Wiener-Hopf (6.5), y por lo tanto esta se puede usar para la estimación del vector de coeficientes óptimo de un filtro adaptable analógico, si los pesos de conexión que va desde el m-esimo nodo hasta k-esimo nodo esta dado por

$$P_{m,k} = [1 - \phi_{xx}(m,k)]/R \quad \text{for } m = k \quad (6.13)$$

$$P_{m,k} = -\Phi_{xx}(m,k)/R \quad \text{for } m \neq k \quad (6.14)$$

y

$$b_k = \Phi_{xd}(k)/R \quad (6.15)$$

donde  $\Phi_{xx}(m,k)$  y  $\Phi_{xd}(k)$  son los valores de las secuencias de correlación, estimados a partir de los datos de entrada como se muestra en las figuras 6.2b y 6.2c.

Con el fin de comparar las características de convergencia del algoritmo propuesto con las del algoritmo RLS convencional, supongamos una implementación analógica del sistema y consideremos que vector de salida de la red de Hopfield modificada esta dado por las ecuaciones (6.5) y (6.10). Estas ecuaciones proporcionan el vector optimo de coeficientes, sin embargo la matriz de correlación  $\Phi_{xx}(k)$  y el vector de correlación cruzada  $\Phi_{xd}(k)$  no son conocidos de antemano en la mayoría de los casos prácticos, por lo tanto estos se tienen que estimar a partir de los datos de entrada. La manera mas común para estimar  $\Phi_{xx}(k)$  y  $\Phi_{xd}(k)$  es mediante el uso de un filtro pasa bajas de primer orden, como se muestra en las siguientes ecuaciones [6.1], [6.2], [6.30]

$$\hat{\Phi}_{xx}(t) = \gamma \hat{\Phi}_{xx}(t-T) + (1-\gamma) \mathbf{X}(t) \mathbf{X}^T(t) \quad (6.16)$$

$$\hat{\Phi}_{xd}(t) = \gamma \hat{\Phi}_{xd}(t-T) + (1-\gamma) d(t) \mathbf{X}(t) \quad (6.17)$$

donde  $T/(1-\gamma)$  es el valor estimado de la memoria de sistema, (medido en segundos), y T es el periodo de muestreo. Iterando las ecuaciones (6.16) y (6.17), después de algunas manipulaciones se obtiene [6.1], [6.29]

$$\hat{\Phi}_{xx}(t) = (1-\gamma) [\gamma \hat{\Phi}_{xx}(t-T) + \mathbf{X}(t) \mathbf{X}^T(t)] \quad (6.18)$$

$$\hat{\Phi}_{xd}(t) = (1-\gamma) [\gamma \hat{\Phi}_{xd}(t-T) + d(t) \mathbf{X}(t)] \quad (6.19)$$

Finalmente, sustituyendo las ecuaciones (6.18) y (6.19) en las ecuaciones (6.11) y (6.12) a partir de la ecuación (6.10), obtenemos

$$\mathbf{W}(t) = [\gamma \hat{\Phi}_{xx}(t-T) + \mathbf{X}(t)\mathbf{X}^T(t)]^{-1} [\gamma \hat{\Phi}_{xd}(t-T) + d(t)\mathbf{X}(t)] \quad (6.20)$$

El algoritmo de mínimos cuadrados recursivo (RLS) estándar es una realización recursiva de la ecuación (6.20), por lo cual se puede esperar que el algoritmo propuesto y el algoritmo estándar RLS tengan las mismas características de convergencia cuando ambos operan bajo las mismas condiciones. Por lo tanto, el desajuste en la solución óptima producido por ambos algoritmos adaptables debido al vector de ruido en los coeficientes del sistema

$$\varepsilon_1 = \mathbf{w}(t) - E[\mathbf{w}(t)] \quad (6.21)$$

esta dado aproximadamente por [6.1]

$$M = \frac{(1-\gamma)N}{1+\gamma} \quad (6.22)$$

donde  $N$  es el número de coeficientes del filtro. Aquí, suponiendo que la memoria del sistema en segundos se mantiene constante, si el periodo de muestreo se reduce por un constante  $K$  o sea que la señal de entrada y la señal de referencia se sobre muestrean por un factor  $K$ , cosa que puede ser hecha fácilmente en realizaciones analógicas debido a que la velocidad de procesamiento de los sistemas analógicos es mucho mayor que la velocidad de los digitales, el nuevo factor de olvido estará dado por  $1-(1-\gamma)/K$ . Esto resulta, de ecuación (6.22), en un menor desajuste. Así, debido a que el filtro adaptable analógico propuesto es potencialmente capaz de operar a frecuencias mucho mayores que a la que operan los filtros adaptables digitales se puede esperar un menor desajuste en el vector de coeficientes del sistema propuesto, en comparación con el de su contra parte digital, después de obtenida la convergencia.

### 6.3. Complejidad Computacional

La estructura propuesta esta diseñada para realización analógica usando datos de entrada analógicos y una forma de realización continua. Por lo tanto estrictamente hablando, no es posible hablar acerca de complejidad computacional porque en las realizaciones analógicas,

cada multiplicación es realizada en paralelo por un multiplicador analógico diferente. Esto significa que un filtro analógico con N coeficientes tendrá cuando menos N multiplicadores para filtrado. Además de las multiplicaciones analógicas, las demás operaciones que se requieren para modificar los coeficientes del sistema se realizan en paralelo. Así, aunque la estructura propuesta se ha intentado para una realización analógica, esta también podría realizarse en forma digital. Así este caso, la estructura propuesta requiere, aproximadamente, 2N multiplicaciones para estimación del vector de correlación  $\Phi_{xd}(k)$  y 2N+1 multiplicaciones para la estimación de la correlación  $\Phi_{xx}(k)$  y N multiplicaciones para filtrado. Además de estos, se requiere  $N^2+2N$  multiplicaciones para adaptación de los pesos usando red de Hopfield modificada. Así, suponiendo que la estructura propuesta es realizada en forma digital, su complejidad total es aproximadamente  $N^2+7N+1$  multiplicaciones por cada periodo de muestreo. Esta complejidad es equivalente a la complejidad computacional del algoritmo RLS convencional.

#### 6.4. Simulaciones Computacionales

Los filtros adaptables se han usado exitosamente en la solución de numerosos problemas prácticos, tales como cancelación de ruido y cancelación de eco, en los cuales filtros adaptables operan como un identificador de sistema, así como codificación predictiva lineal y aclaramiento de líneas [6.1], etc., en los cuales se usan también filtros adaptables operando con una configuración de predicción lineal. Para evaluar el uso potencial de estructura propuesta en la solución de varios tipos de los problemas prácticos, el funcionamiento de ésta se evaluó usando simulaciones computacionales, en las cuales el sistema propuesto opera ya sea como identificador de sistema y predicción lineal. En todos estos casos, la realización analógica con sobre-muestreo en la línea de retardo se simuló por medio de la transformada bilineal, y el comportamiento analógico del sistema fue simulado como se muestra en figura 6.3. La ecuación diferencial de la red de Hopfield modificada ha calculado usando el método de Runge-Kutta de cuarto orden, y la secuencia de correlación se requiere durante proceso de adaptación se ha estimado usando filtros pasa baja de primer orden.

$$\Phi_{xd}(k, t) = \gamma \Phi_{xd}(k, t - T) + (1 - \gamma) x_k(t) d(t) \quad (6.23)$$

$$\Phi_{xx}(j, k, t) = \gamma \Phi_{xx}(j, k, t - T) + (1 - \gamma) x_j(t) x_k(t) \quad (6.24)$$

donde  $0 < \gamma < 1$  es el factor de olvido, el cual está relacionado con número de datos usados para estimar las componentes de la matriz de autocorrelación  $\Phi_{xx}(j, k, t)$  y el vector de correlación  $\Phi_{xd}(k, t)$ ,  $x_k(t)$  es el dato de entrada que se multiplica con k-esimo coeficiente  $w_k(t)$  del filtro adaptable, y  $d(t)$  es la respuesta deseada del filtro o señal de referencia.

Una medida que se puede emplear para evaluar el funcionamiento del filtro adaptable propuesto es error cuadrático medio normalizado, el cual se puede definir como sigue

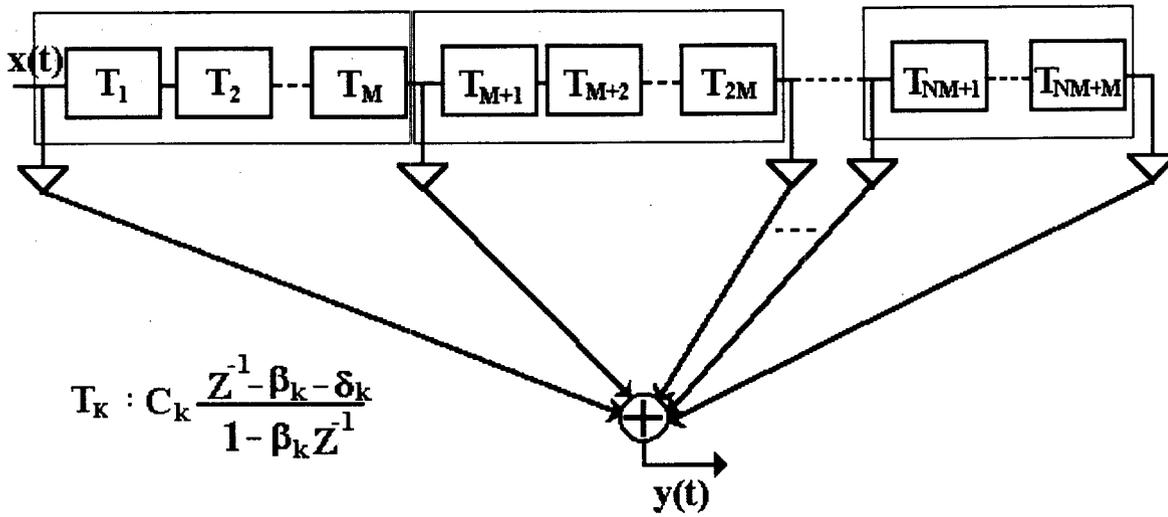


Figura 6.3. Estructura de tiempo discreto usada para simular el funcionamiento de una estructura de tiempo continuo.

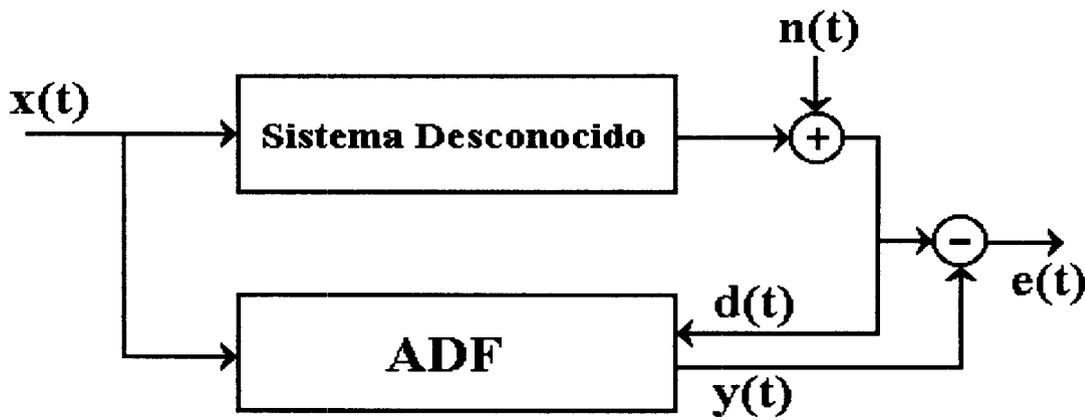


Figura 6.4. Filtro adaptable operado con una configuración de identificación de sistemas.

$$MSE = 10 \text{Log}_{10} \left[ \frac{\sum_{k=0}^M e^2(kT)}{\sum_{k=0}^M d^2(kT)} \right] \quad (6.25)$$

donde  $e(kT)$  y  $d(kT)$  son el error de salida y la señal de referencia , respectivamente.

El funcionamiento de la estructura propuesta se ha comparado con la estructura RLS convencional en todos los casos.

### Caso 1. Identificador de sistema

Una de las aplicaciones más exitosas del filtro adaptable es el identificador de sistemas (Fig. 6.4), en el cual  $x(t)$  es la señal de entrada de ambos sistemas, el desconocido y la estructura de filtro adaptable, y  $d(t)$  es la señal de salida deseada o señal de referencia. Las figuras 6.5 y 6.6 muestran la convergencia de la estructura propuesta cuando se requiere identificar un sistema desconocido de orden 14, el cual es invariante en el tiempo. Ambas figuras muestran dicha convergencia bajo diferentes condiciones de relación señal a ruido. En la figura 6.5, la relación señal a ruido es igual a 0 dB, 6 dB y 12 dB, mientras que en la figura 6.6 la relación señal a ruido es igual a 18dB, 20dB y 26dB, respectivamente. La secuencia de entrada, en todos los casos es una secuencia de ruido blanco. Las figuras 6.7 y 6.8 muestran la convergencia de la estructura propuesta, cuando opera como identificador de sistemas, como se mencionó anteriormente con señales de voz actual. En la figura 6.7 se muestra el error cuadrático medio(MSE) cuando la relación señal a ruido es de 0dB, 6dB y 12dB, mientras que la figura 6.8 muestra el comportamiento del sistema cuando la relación señal a ruido es de 18dB, 20dB y 26dB. En ambos casos la señal de entrada es una señal de voz actual.

La figura 6.9 muestra comparación del funcionamiento de convergencia entre la estructura propuesta y la estructura RLS convencional, aquí la señal de entrada fue una señal de voz actual con una relación señal a ruido (SNR) de 26dB, mientras en la figura 6.10, la señal de entrada fue una secuencia de ruido blanco con una relación señal a ruido de 20dB. Las figuras 6.5 – 6.10 muestran que la estructura propuesta y el algoritmo RLS convencional tienen una velocidad de convergencia inicial similar, aunque el algoritmo propuesto proporciona un menor MSE, y tiene menor desajuste que el algoritmo convencional, debido a que el factor de sobre muestreo  $K$  es 10, lo cual se puede hacer fácilmente en una realización analógica. Las figuras 6.9 y 6.10 muestran que el funcionamiento de convergencia de la estructura RLS convencional es muy sensible a variaciones en el factor de olvido, el cual decide la cantidad de memoria para evaluación de correlación. Cuando el sistema es invariante en tiempo el valor de este factor de olvido debe ser cercano a 1, para obtener un mejor funcionamiento. En todos los casos  $M$  de la ecuación (6.25) fue igual a 49.

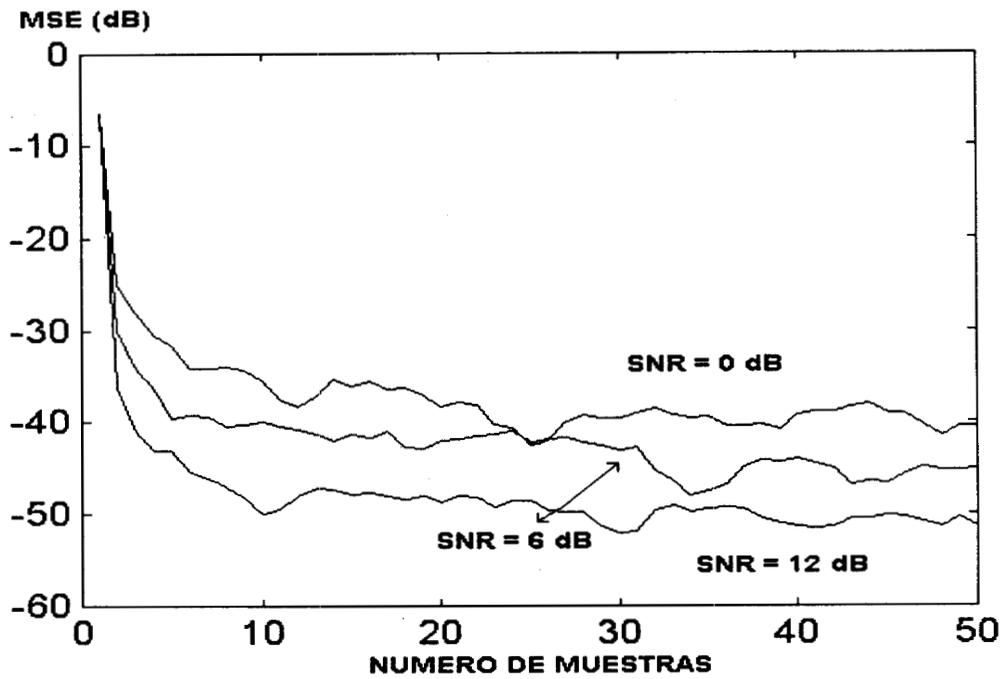


Figura 6.5. Convergencia e la estructura propuesta cuando es requerida a identificar un sistema desconocido de orden 14.

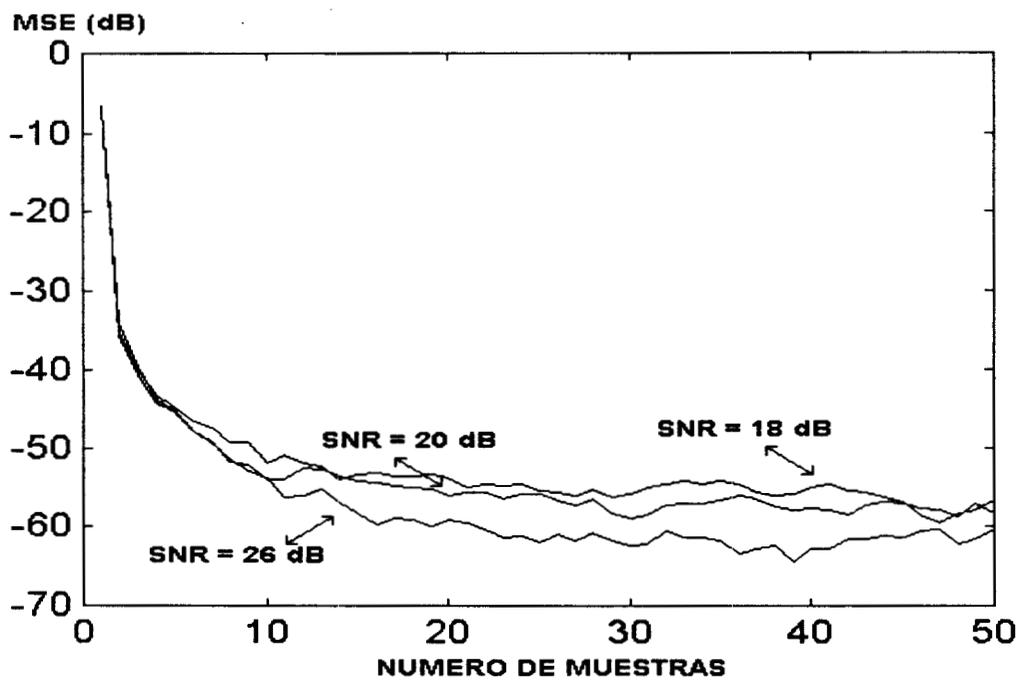


Figura 6.6. Convergencia e la estructura propuesta cuando es requerida a identificar un sistema desconocido de orden 14.

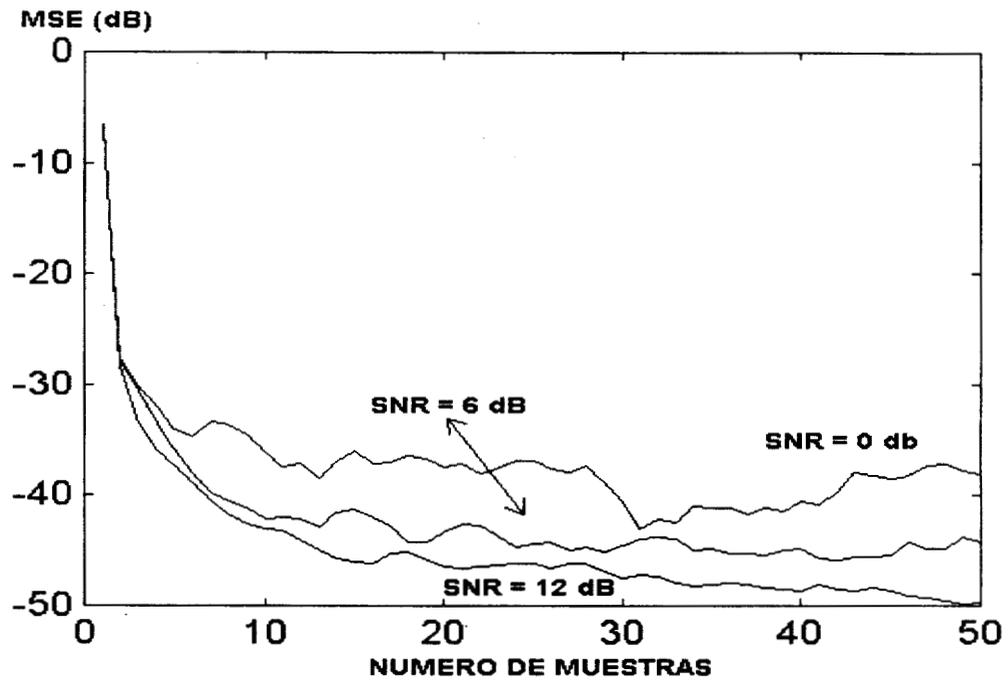


Figura 6.7. Convergencia de la estructura propuesta cuando esta es requerida a identificar un sistema desconocido de orden 14. La señal de entrada es una señal de vos actual.

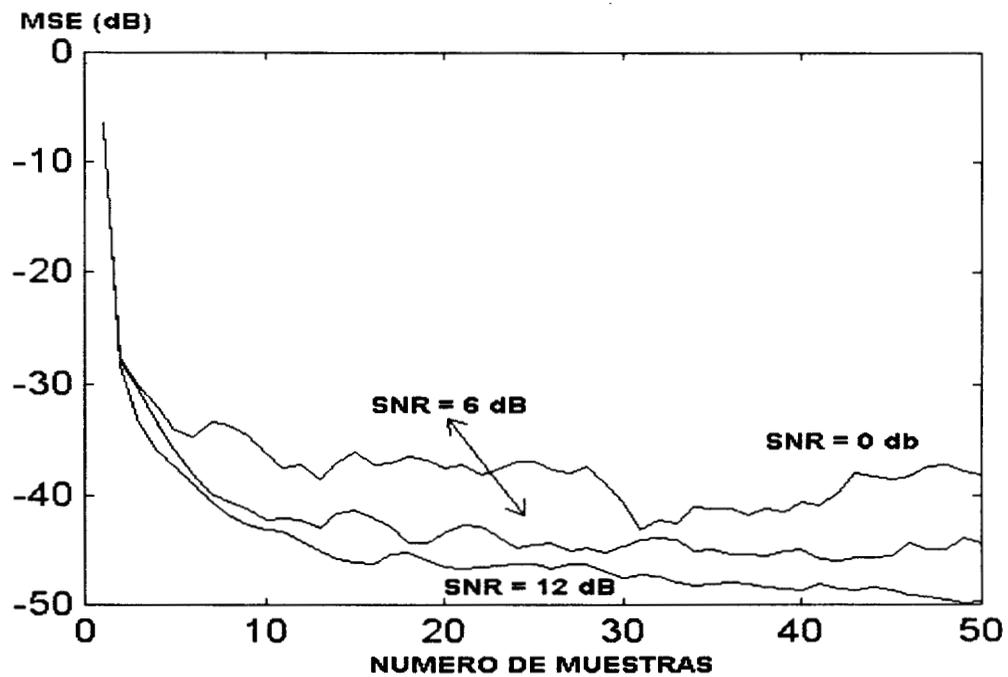


Figura 6.8. Convergencia de la estructura propuesta cuando esta es requerida a identificar un sistema desconocido de orden 14. La señal de entrada es una señal de vos actual.

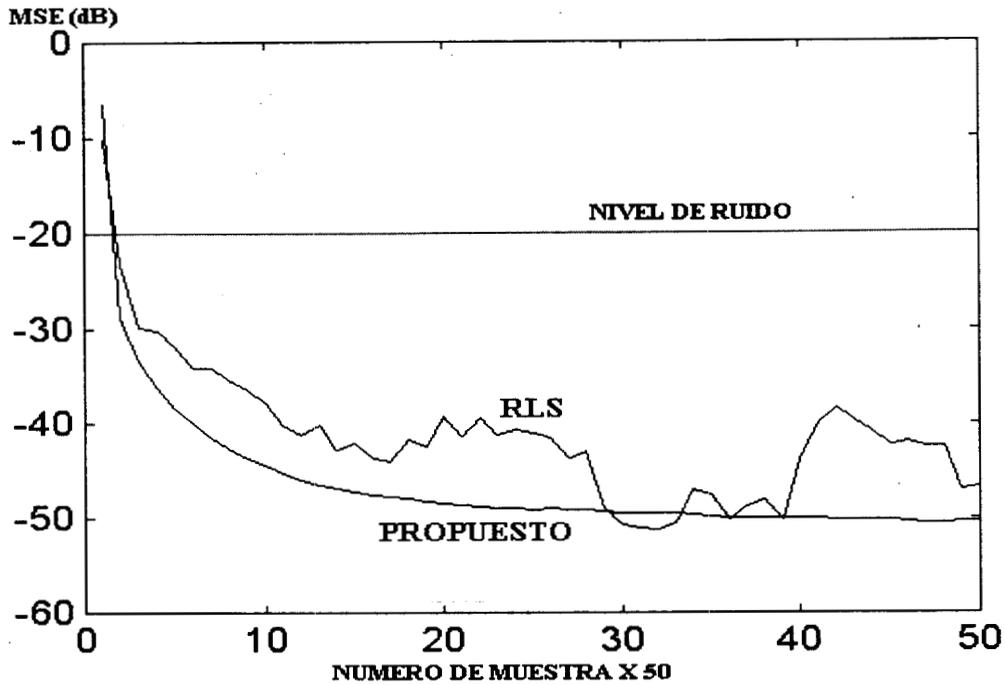


Figura 6.9. Características de convergencia del algoritmo propuesto operando en una configuración de identificación de sistemas. El orden del filtro es 14 y la señal de entrada una secuencia de ruido blanco. La convergencia obtenida usando el algoritmo RLS convencional se muestra para su comparación.

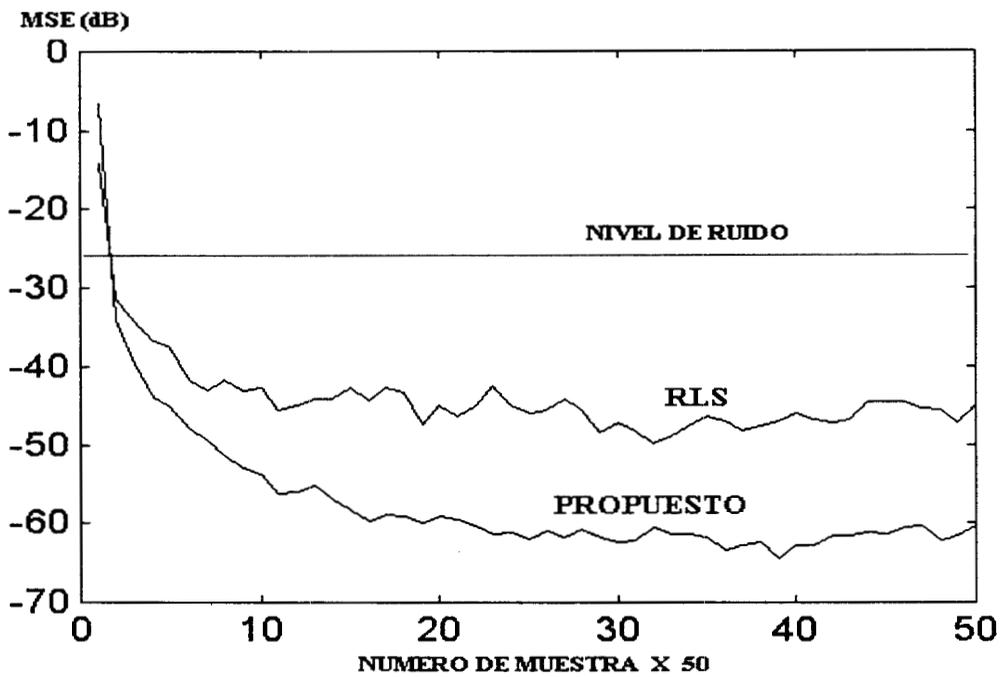


Figura 6.10. Características de convergencia del algoritmo propuesto operando en una configuración de identificación de sistemas. El orden del filtro es 14 y la señal de entrada una señal de voz actual. La convergencia obtenida usando el algoritmo RLS convencional se muestra para su comparación.

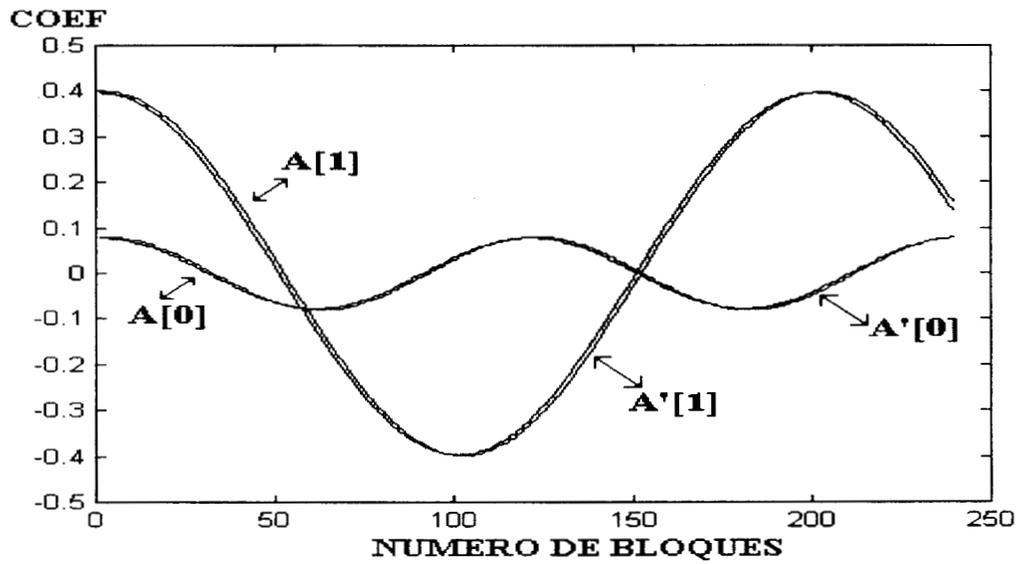


Figura 6.11. Trayectoria de los coeficientes del sistema propuesto  $A'[i]$  cuando éste es requerido a identificar un sistema de orden 14 que varía lentamente con el tiempo  $A[i]$ . La señal de entrada es una señal de vos actual.

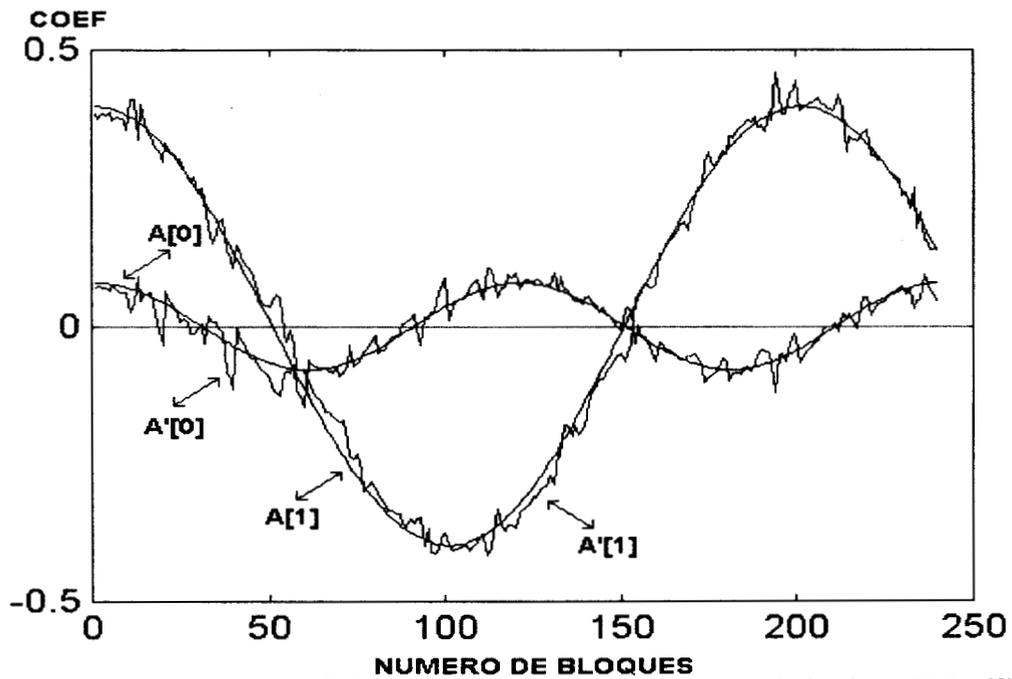


Figura 6.12. Trayectoria de los coeficientes de un filtro transversal usando el algoritmo RLS  $A'[i]$  cuando éste es requerido a identificar un sistema de orden 14 que varía lentamente con el tiempo  $A[i]$ . La señal de entrada es una señal de vos actual.

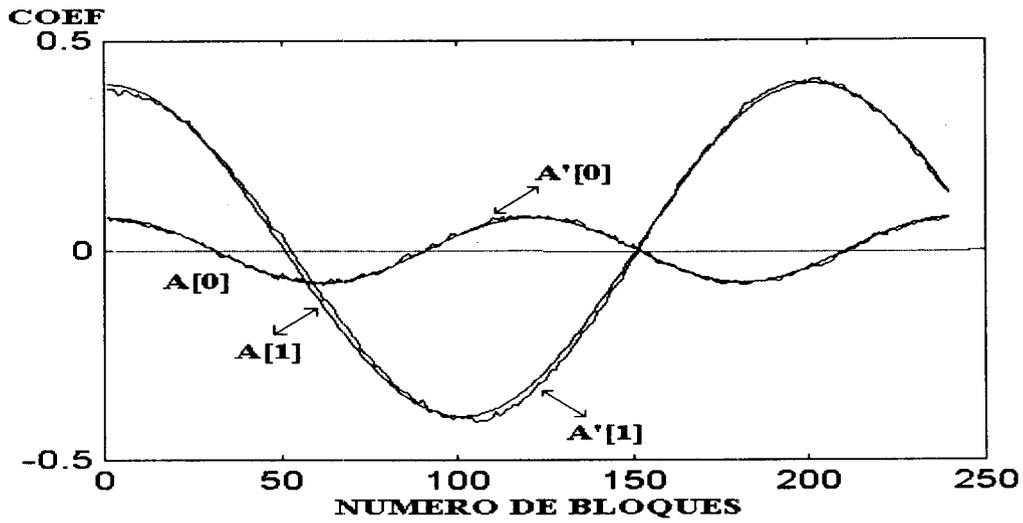


Figura 6.13. Trayectoria de los coeficientes de un filtro transversal usando el algoritmo RLS  $A'[i]$  cuando éste es requerido a identificar un sistema de orden 14 que varía lentamente con el tiempo  $A[i]$ . La señal de entrada es una señal de ruido blanco.

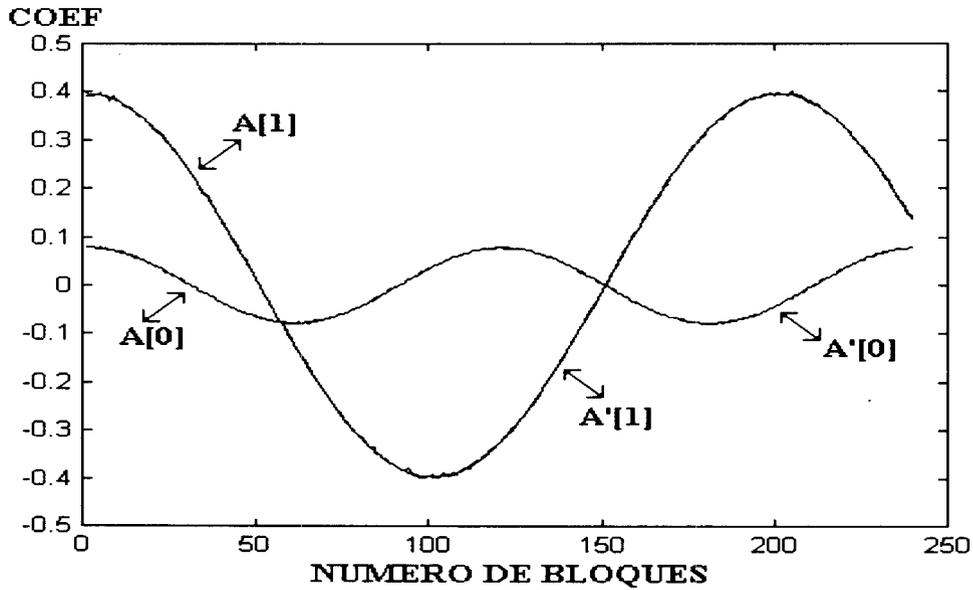
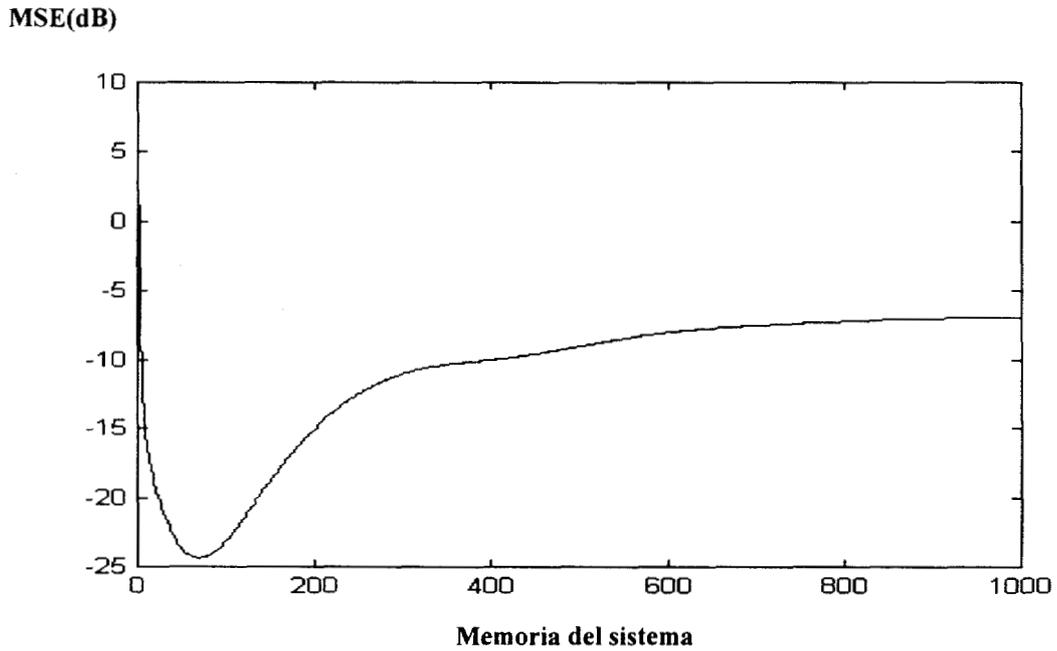
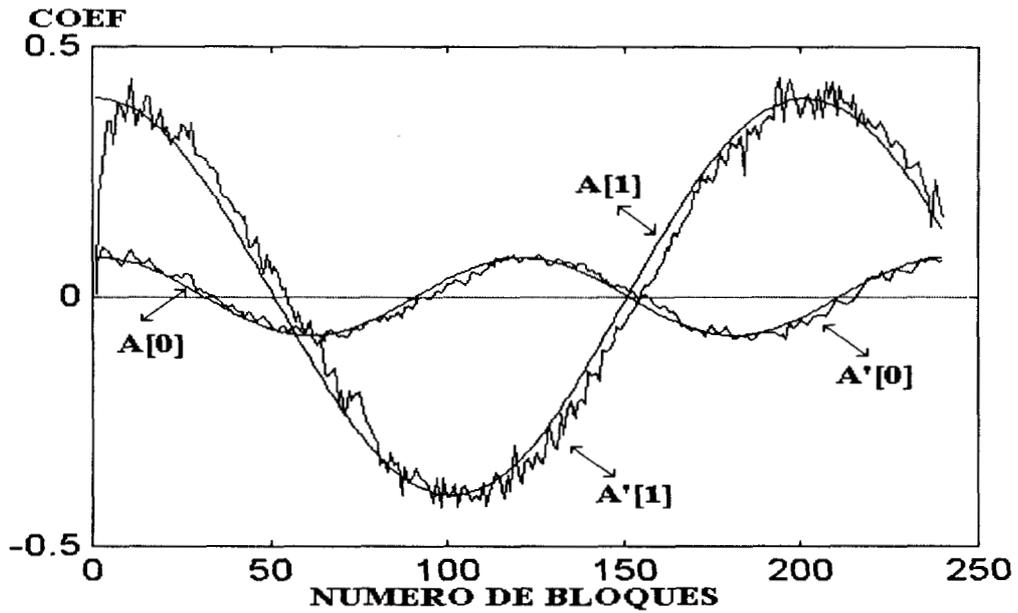


Figura 6.14. . Trayectoria de los coeficientes del sistema propuesto  $A'[i]$  cuando éste es requerido a identificar un sistema de orden 14 que varía lentamente con el tiempo  $A[i]$ . La señal de entrada es una señal de ruido blanco.



**Figura 6.15.** Error cuadrático medio, en función a la memoria del sistema obtenido cuando un filtro transversal adaptado usando el algoritmo RLS es requerido a identificar un sistema desconocido variable con el tiempo. La señal de entrada fue una secuencia de ruido blanco.



**Figura 6.16.** Trayectoria de los coeficientes del sistema propuesto  $A'[i]$  cuando éste es requerido a identificar un sistema de orden 14 que varía rápidamente con el tiempo  $A[i]$ . La señal de entrada es una señal de ruido blanco.

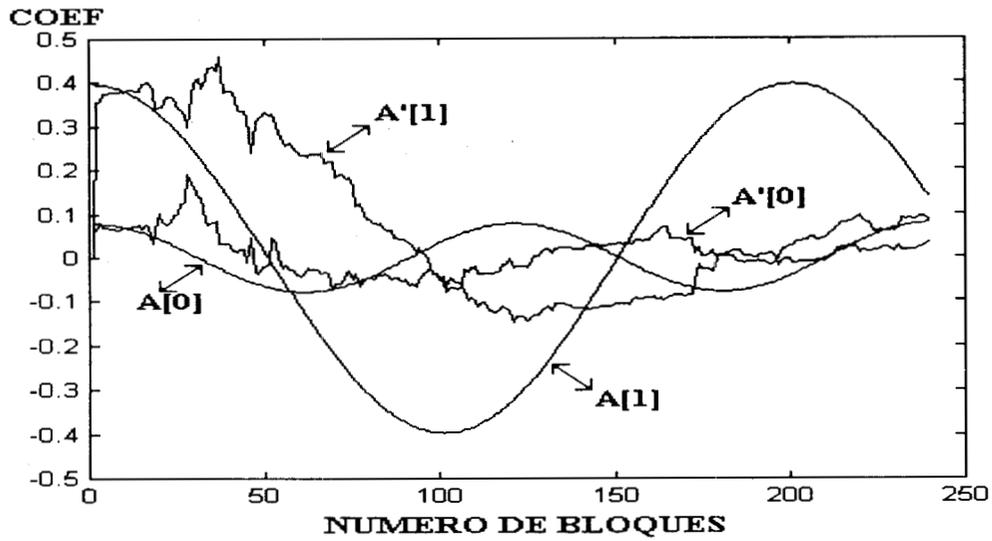


Figura 6.17. Trayectoria de los coeficientes de un filtro transversal usando el algoritmo RLS,  $A'[i]$ , cuando éste es requerido a identificar un sistema de orden 14 que varía rápidamente con el tiempo  $A[i]$ . La señal de entrada es una señal de ruido blanco y el factor de olvido  $\lambda=0.99$ .

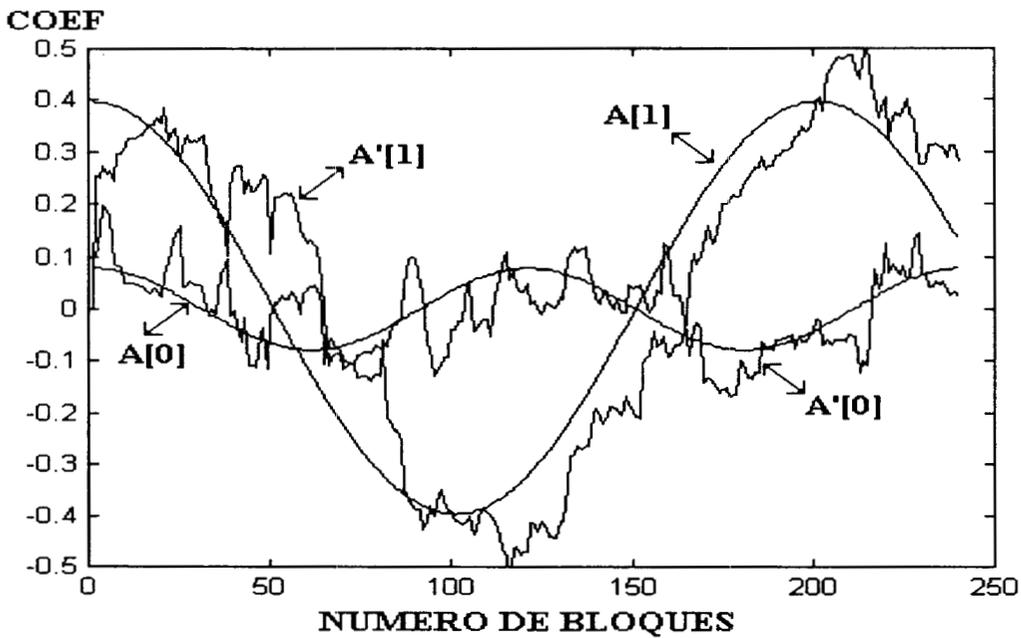


Figura 6.18. Trayectoria de los coeficientes de un filtro transversal usando el algoritmo RLS,  $A'[i]$ , cuando éste es requerido a identificar un sistema de orden 14 que varía rápidamente con el tiempo  $A[i]$ . La señal de entrada es una señal de ruido blanco y el factor de olvido  $\lambda=0.9$ .

El factor de olvido empleado fue igual a 0.9999.

Figuras 6.11 – 6.14 muestran la trayectoria de los coeficientes de la estructura propuesta y filtro adaptable RLS en el tiempo discreto cuando son requeridos a identificar un sistema desconocido variando lentamente en el tiempo. Los coeficientes del sistema desconocido cambia en cada bloque como se muestra ecuaciones (6.26) y (6.27), donde  $k$  es el número de bloque y cada bloque tiene 100 muestras.

$$A_i[k] = A_i[0] * \cos(2\pi k / 120) \quad i \text{ es impar} \quad (6.26)$$

$$A_i[k] = A_i[0] * \cos(2\pi k / 200) \quad i \text{ es par} \quad (6.27)$$

La señal de entrada en las figuras 6.11 y 6.12 es una señal de voz actual, mientras que en las figuras 6.13 y 6.14 el sistema operara con una secuencia de ruido blanco. El factor de olvido usado en la simulación del algoritmo RLS es 0.99, el cual es el mejor valor para la situación en la que el sistema varía lentamente como se muestra figura 6.15.

Las figuras 6.16-6.18 muestran la misma trayectoria cuando los coeficientes varían en cada período de muestreo. En la figura 6.16 se muestra la trayectoria de los coeficientes de la estructura propuesta, mientras que las figuras 6.17 y 6.18 se muestran las trayectorias de los coeficientes del filtro transversal RLS convencional, con factores de olvido iguales a 0.99 y 0.9 respectivamente. En los tres casos la señal de entrada es una secuencia de ruido blanco.

Estas figuras muestran que el algoritmo propuesto tiene gran capacidad de seguimiento cuando el sistema desconocido varía rápidamente con el tiempo. Mientras que el algoritmo estándar de RLS muestra cierta dificultad para seguir este tipo de variación. En todos casos  $A[6.k]$  significa coeficiente actual y  $A'[6.k]$  es valor estimado.

## Caso 2. Predicción Lineal

La señal de entrada de un filtro adaptable operando con una configuración de predicción lineal (Fig. 6.19) es una versión retardada de la señal de referencia, esto es,  $x(t)=d(t-T)$ , donde  $d(t)$  es la señal de referencia.

La figura 6.20 muestra la función de autocorrelación de una señal de voz actual, la función de autocorrelación de la señal estimada usando ambos, la estructura propuesta y el filtro RLS convencional, junto con la función de autocorrelación de los errores de salida. Aquí (1) es la secuencia de autocorrelación de la señal de voz, (2) es la secuencia de

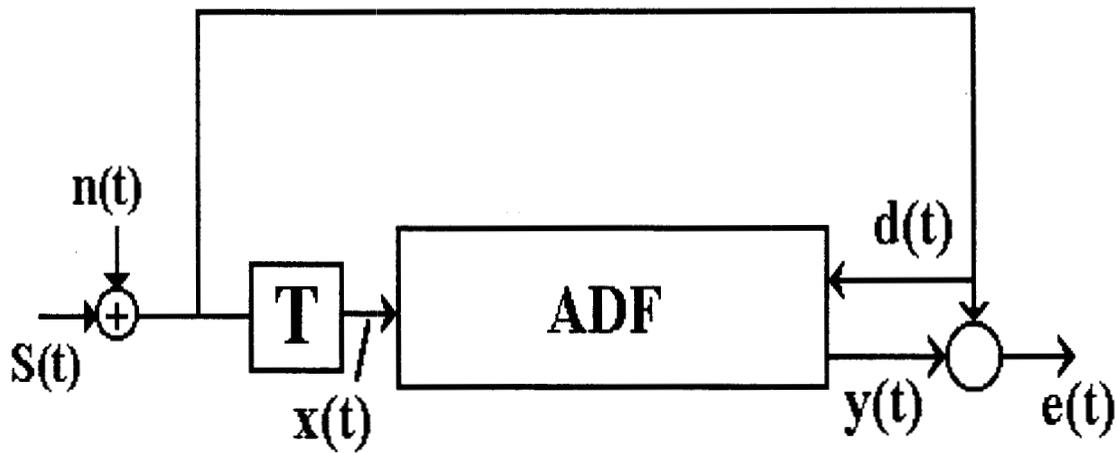
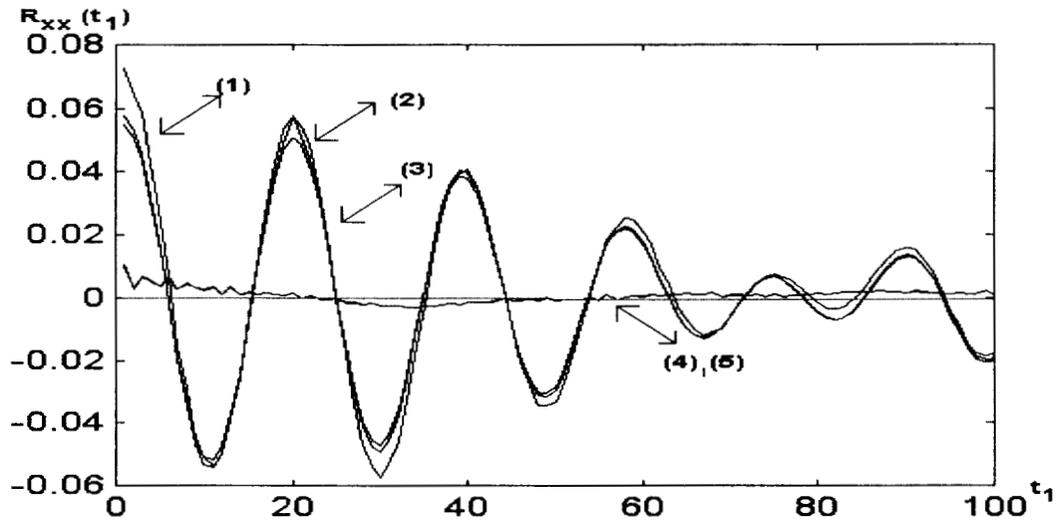
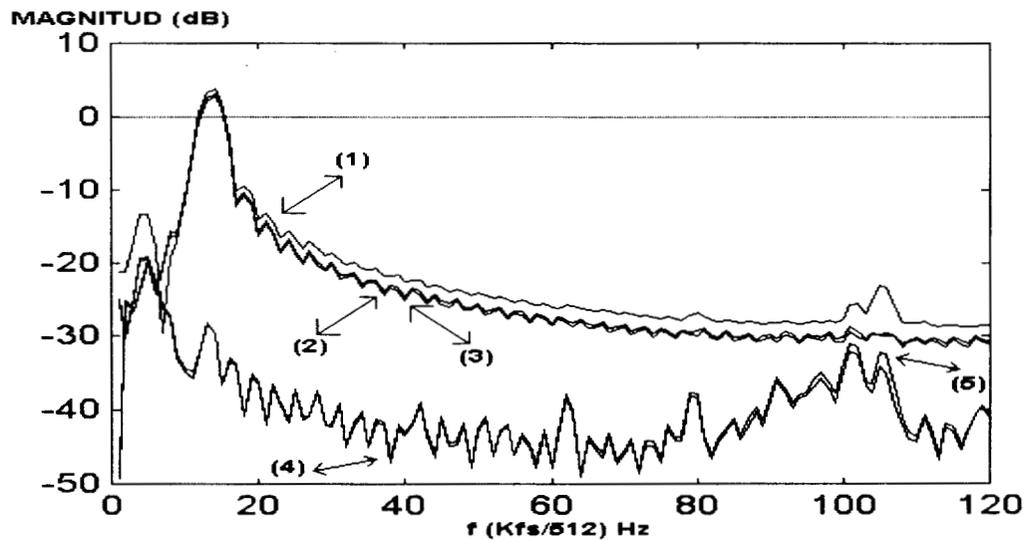


Figura 6.19. Filtro adaptable operando en configuración de predicción lineal.



**Figura 6.20.** (1) Secuencia de autocorrelación de una señal de vos actual. (2) Secuencia de autocorrelación de la señal estimadas usando la estructura propuesta operando como predictor lineal. (3) Secuencia de autocorrelación de la señal estimada usando un predictor lineal adaptado usando el algoritmo RLS. (4) y (5) errores de predicción obtenidos con las dos estructuras, la propuesta y la convencional.



**Figura 6.21.** (1) Densidad e potencia espectral de una señal de vos actual. (2) Densidad de potencia espectral de la señal estimadas usando la estructura propuesta operando como predictor lineal. (3) Densidad de potencia espectral de la señal estimada usando un predictor lineal adaptado usando el algoritmo RLS. (4) y (5) Densidad e potencia espectral de los errores de predicción obtenidos con las dos estructuras, la propuesta y la convencional.

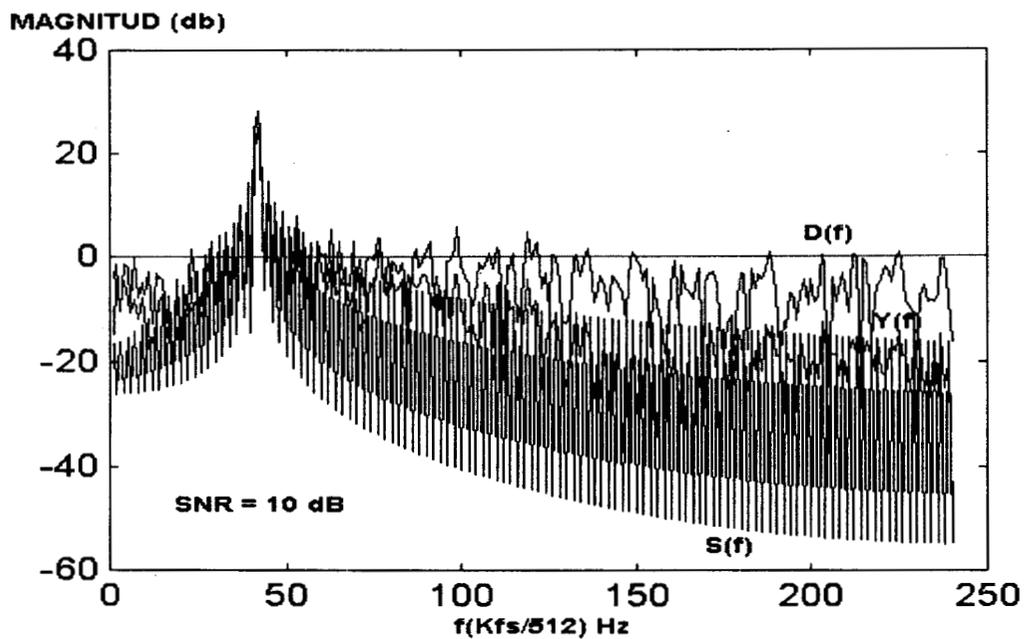


Figura 6.22. Funcionamiento del sistema propuesto cuando es requerido a operar como aclarador de líneas. La señal de entrada es una onda senoidal con una relación señal a ruido de 10 dB.

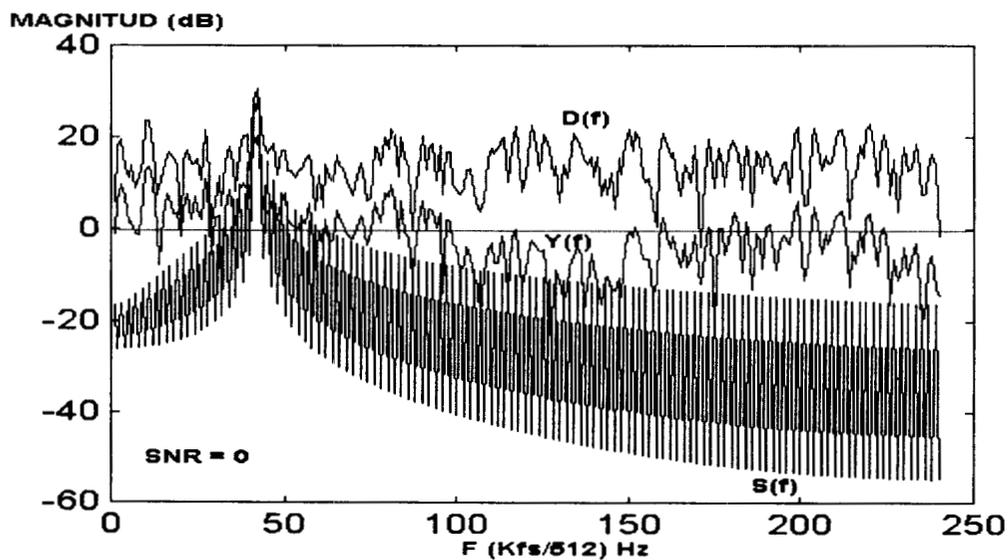


Figura 6.23. Funcionamiento del sistema propuesto cuando es requerido a operar como aclarador de líneas. La señal de entrada es una onda senoidal con una relación señal a ruido de 10 dB.

autocorrelación de la señal estimada usando la estructura propuesta, (3) es la secuencia de autocorrelación de la señal estimada usando el algoritmo RLS convencional, mientras que (4) y (5) son los errores de predicción usando la estructura propuesta y el predictor transversal adaptado usando el algoritmo RLS convencional, respectivamente.

Las densidades de potencia espectral de las señales de voz, tanto de las actuales como de las estimadas, se muestran en la figuras 6.21. En todos los casos el tiempo de retardo fue igual a un período de muestreo. La tabla 1 muestra el vector de coeficientes obtenido cuando ambos, el algoritmo propuesto y el RLS son usados para predecir un proceso auto regresivo de orden 5. El cual fue generado haciendo pasar una secuencia de ruido blanco a través de un filtro de quinto orden con solo polos. Los coeficientes del filtro usado para generar el proceso auto regresivo son mostrados también en la tabla 1 para comparación.

Las figuras 6.22 y 6.23 muestran la densidad de potencia espectral obtenida cuando el algoritmo propuesto opera en una configuración de aclarador de líneas (fig. 6.19), con un tiempo de retardo entre la señal de entrada y la referencia igual a 5 periodos de muestreo. La entrada fue una señal coseno contaminada aditivamente por una señal de ruido blanco, con una relación señal a ruido igual a 10 y 0 dB respectivamente. Las densidades de potencia espectral de  $s(t)$ ,  $d(t)$  y  $y(t)$ , (Figs. 6.22 y 6.23) fueron estimadas usando la Transformada Discreta de Fourier con 512 puntos. Esas figuras muestran que la estructura propuesta se comporta en forma altamente satisfactoria cuando ésta opera con una configuración de predicción lineal.

AR-process 5-order	Proposed			RLS		
	SNR=100dB	SNR=22dB	SNR= 12dB	SNR=100dB	SNR=22dB	SNR= 12dB
1.3136	1.313735	1.270557	0.827257	1.31435	1.206089	0.832661
-1.4401	-1.447184	-1.344815	-0.464168	-1.436659	-1.204447	-0.475778
1.0919	1.100099	1.270557	0.202012	1.082074	0.871011	-0.475778
-0.83527	-0.837452	-0.797337	-0.503518	-0.829362	-0.756292	-0.509769

**Tabla 1. Coeficientes del filtro adaptable obtenidos cuando el sistema propuesto y el convencional son usados con una configuración de identificación de sistemas para estimar u proceso AR de orden 5.**

## 6.5. Conclusiones

En este capitulo se propuso una estructura para filtrado adaptable analógico en la cual la adaptación del vector de coeficientes se lleva a cabo en tiempo continuo usando una red modificada de Hopfield, con una función de activación lineal. La estructura propuesta no

presenta problemas de estabilidad, ya que no requiere ninguna inversión de matriz explícita para realizar el algoritmo RLS en tiempo continuo.

La estructura propuesta fue evaluada por medio de simulaciones computacionales, en las cuales el sistema propuesto operó como identificador de sistemas y como predictor lineal. Los resultados obtenidos muestran que el sistema propuesto, operando como identificador de sistemas en condiciones estacionarias, presenta una velocidad de convergencia inicial similar a la proporcionada por el algoritmo RLS convencional. Aunque el sistema propuesto presenta un menor desajuste que el presentado por el algoritmo RLS convencional debido al sobre muestreo introducido en la realización analógica. Además, cuando el sistema propuesto opera en condiciones en las cuales las señales de entrada y referencia son no estacionarias, éste presenta características de convergencia superiores a aquellas del algoritmo RLS convencional, especialmente cuando las estadísticas de las señales involucradas varían rápidamente con el tiempo.

Las simulaciones computacionales muestran que cuando el sistema opera con una configuración de predictor lineal, presenta un funcionamiento muy similar al del algoritmo RLS convencional, operando ambos bajo las mismas condiciones. De los resultados anteriores se concluye que la estructura propuesta presenta un funcionamiento altamente satisfactorio cuando opera en ambientes tanto estacionarios como no estacionarios, y debido a su realización analógica, parece ser una buena alternativa a los filtros adaptables digitales, en aplicaciones con señales de entrada y referencia las cuales tienen componentes de alta frecuencia.

## **Red de Hopfield Modificada para Igualador de Canal de Comunicación**

El método más comúnmente usado para reducir la interferencia intersimbolos debido al desvanecimiento selectivo a ciertas frecuencias, la cual se encuentra presente en los canales de comunicaciones modernas, es la igualación adaptable. Sin embargo, a pesar de la efectividad de esta técnica muchos problemas permanecen aun sin solución, como el incremento de la frecuencia de muestreo, reducción de la complejidad computacional, reducción del tamaño y de los requerimientos de potencia de los sistema de igualadores usados. Para resolver estos problemas en este capítulo se propone una estructura de igualador de decisión retroalimentada (DFE) de tiempo continuo, basada en el algoritmo propuesto en el capítulo 6, en el cual la adaptación del vector de coeficientes se lleva a cabo mediante el uso de una red neuronal de Hopfield modificada de tiempo continuo, con valores de entrada complejos. Esto permite que la salida del DFE y la actualización del vector de coeficientes pueden ser computados en menor tiempo y con menor consumo de potencia que el requerido por sus contrapartes digitales. El funcionamiento de la estructura propuesta se compara con el funcionamiento de la estructura convencional adaptada usando el algoritmo digital RLS, usando simulación computacional.

## 7.1 Introducción

La información digital transmitida a través de los canales de comunicación físicos es, en la mayoría de las ocasiones, distorsionada debido a la interferencia de intersímbolos, (ISI), la cual es causada principalmente por la propagación multitrayectoria de las señales transmitidas, así como por las características no ideales de los canales de comunicación [7.1]-[7.20]. Para evitar este problema se ha llevado a cabo una intensiva investigación durante las últimas dos décadas. Como resultado de esto, varios métodos los cuales intentan resolver los problemas causados por la interferencia intersímbolos (ISI) en los sistemas de comunicación digital, han aparecido en la literatura durante los últimos años [7.1]-[7.8], [7.10], [7.14]-[7.20]. Entre los métodos propuestos, las estructuras de igualación adaptable son las más ampliamente usadas [7.1]-[7.8], [7.14], [7.16], [7.18]-[7.20], convirtiéndose muchas de ellas en parte fundamental de un gran número de sistemas de comunicación prácticos que operan con canales de comunicación quasiaestacionarios, casi todos ellos implementados en forma digital. Esto es porque el avance de la tecnología digital ha hecho sido posible la realización de algoritmos sofisticados para procesamiento digital de señales. Sin embargo, cuando algunas de estas estructuras son usadas para reducir la ISI en los canales de comunicación variables con el tiempo, tales como un canal de comunicación móvil, ellas presentan serias dificultades y un funcionamiento pobre en muchos de los casos, llevando a una degradación del funcionamiento de la mayoría de los sistemas de comunicación.

Los canales de comunicación móvil pueden variar rápidamente debido al movimiento del vehículo, influyendo grandemente en la magnitud de las variaciones las irregularidades del terreno, la densidad de edificios, etc. [7.9], [7.11], [7.13]. Así los canales de comunicación pueden asumirse como estacionarios solamente en fracciones de longitudes de onda, esto es, los canales de comunicación móvil se pueden asumir estacionarios solo sobre 4 a 5 m, en la banda de los 900 MHz. Esto ocasiona que en muchos casos la velocidad de variación del canal sea mas rápida que la velocidad de convergencia del igualador, resultando en un pobre funcionamiento del igualador adaptable. Esto a su vez ocasiona un funcionamiento deficiente de todo el sistema de comunicación móvil.

El tiempo de retardo entre la onda directa y la onda indirecta (o reflejada) es otro factor que también debe ser considerado. Por ejemplo las comunicaciones móviles dependen del medio en el cual se encuentre el vehículo en movimiento, típicamente el tiempo de retraso entre la onda directa y la onda indirecta (o reflejada) puede ser más de 10  $\mu$  seg. [7.9], [7.11], [7.13]. En adición a los problemas debidos a las variaciones con el tiempo de las características de los canales de comunicaciones móviles mencionadas anteriormente, desvanecimientos profundos y anchos en ciertas bandas de frecuencia causan que el igualador deba colocar altas ganancias en esas bandas de frecuencia con el fin de compensar el desvanecimiento de la señal, incrementado en esta forma el ruido aditivo. Otra importante limitación que debe ser considerada en las comunicaciones móviles son los requerimientos de tamaño y potencia de los sistemas usados. Por ejemplo en muchos casos las estructuras de los igualadores adaptables usadas en comunicaciones móviles requieren tener una baja complejidad computacional, simple realización en hardware y bajo consumo

de potencia, además de una buena habilidad de seguimiento para compensar las variaciones con el tiempo de las características del canal de comunicación.

Los igualadores adaptables más usados para reducir la interferencia intersímbolos, ISI, en canales de comunicación estacionarios o cercanamente estacionarios, son los igualadores de decisión retroalimentada (DFE) y el Estimador de Máxima Semejanza (MLSE) [7.1]-[7.5],[7.18]. En general el MLSE, el cual es implementado mediante el uso del algoritmo Viterbi, es el método más efectivo de detección para comunicaciones digitales distorsionadas por ISI y ruido aditivo cuando las estadísticas de los canales de comunicación son constantes o varían lentamente con el tiempo [7.1], [7.2], [7.18]. Sin embargo, el MLSE tiene una alta complejidad computacional la cual limita su aplicación en algunos sistemas prácticos. Además, cuando se usa en canales de comunicación que varían rápidamente con el tiempo, los errores acumulados en las funciones de probabilidad degradan considerablemente el funcionamiento del igualador MLSE [7.6]-[7.8].

El igualador de decisión retroalimentada (DFE) tiene una mucho menor complejidad computacional que el MLSE [7.2], [7.18] y, a pesar de que en ciertos casos no es capaz de obtener la solución óptima proporcionada por el MLSE, el funcionamiento del DFE es similar al del MLSE en muchas situaciones prácticas, con una mucho menor complejidad computacional [7.2], [7.18]. La parte principal del igualador de decisión retroalimentada, DFE, es un filtro adaptable cuyo vector de coeficientes es continuamente actualizado el algoritmo LMS "Least Mean Square" o RLS "Recursive Least Square" [7.1], [7.2], [7.18], cuyas características se presentan en el capítulo 3.

Para compensar la severa distorsión introducida por la rápida variación en el tiempo de los canales móviles, se requieren algoritmos de adaptación con altas velocidades de convergencia para actualizar los coeficientes del DFE. El algoritmo adaptable LMS es ampliamente utilizado en diversos sistemas prácticos, debido a su baja complejidad computacional y a que proporciona un buen funcionamiento cuando se emplea para igualar canales de comunicación estacionarios, o cuando las características del canal cambian lentamente con el tiempo, esto es las variaciones con el tiempo que presenta el canal son mucho más lentos que la velocidad de convergencia del algoritmo [7.2], [7.18]. Así, estas bajas velocidades de convergencia limitan grandemente la habilidad del sistema para igualar canales de comunicación que varían rápidamente con el tiempo tales como los que se presentan en algunos sistemas de comunicación móvil [7.2], [7.18]. Por otro lado, el algoritmo de adaptación RLS tiene velocidades de convergencia mucho mayores que las del algoritmo LMS y una mucho menor sensibilidad al ruido aditivo, lo que le permite seguir con mayor facilidad las variaciones con el tiempo que presentan los parámetros de los canales de comunicación móvil [7.1], [7.2], [7.18]. Debido a todo esto, aunque su complejidad computacional sea mucho mayor que la del algoritmo LMS, el algoritmo RLS con memoria de diferente longitud, es usualmente utilizado para actualizar el vector de coeficientes del DFE [7.2].

Diversos algoritmos han sido propuestos para reducir la complejidad computacional del RLS, como lo es el rápido algoritmo de Kalman [7.1]-[7.3] el cual reduce la complejidad computacional de el algoritmo adaptable de  $O(N^2)$  a  $O(N)$ . Esto representa una

considerable reducción de complejidad computacional. Sin embargo pese a que esta reducción es importante, estos algoritmos aun presentan, en ciertos casos, problemas de estabilidad numérica [7.1]-[7.3]. Así, la alta complejidad computacional del algoritmo RLS aunado a la inestabilidad numérica de las versiones del RLS con baja complejidad computacional, representan problemas de realización cuando son usados en sistemas de comunicación de móvil, tales como los sistemas de telefonía celular.

Recientemente han sido propuestos igualadores adaptables basados en redes neuronales artificiales, los cuales presentan un funcionamiento aceptable en cuando requeridos a igualar canales de comunicación variables con el tiempo [7.6]-[7.8],[7.14],[7.16]. Sin embargo su complejidad computacional es mucho mayor que la del convencional DFE usando el algoritmo adaptable RLS [7.6]-[7.8],[7.14],[7.16]. Este factor puede limitar su uso en algunas aplicaciones prácticas.

Por otro lado el interés en los sistemas adaptables analógicos, ha crecido considerablemente en los últimos años, debido a que ellos tienen el potencial de manejar frecuencias mucho mayores, son de mucho menor tamaño, y presentan un consumo de potencia mucho menor que sus contra partes digitales [7.19]-[7.24].

En este capítulo se propone un DFE de tiempo continuo en el cual el vector de coeficientes es adaptado mediante el uso de una red neuronal de Hopfield modificada de tiempo continuo [7.19],[7.20],[7.25]-[7.28]. Debido a su realización analógica, la estructura propuesta tiene potencialmente un tamaño muy pequeño, bajo consumo de potencia, mayor velocidad de convergencia y mayor habilidad para igualar canales de comunicación variables con el tiempo que sus contra partes digitales.

Los resultados obtenidos por simulación computacional muestran que la estructura propuesta presenta un funcionamiento similar al DFE convencional adaptado usando el algoritmo RLS cuando ambos son requeridos a igualar canales de comunicación estacionarios o canales que varían lentamente con el tiempo. Sin embargo la estructura propuesta supera ampliamente al DFE convencional cuando ambos son requeridos a igualar canales de comunicación cuyas características varían rápidamente con el tiempo.

## 7.2. Modelo del canal de Comunicación Móvil

La figura 7.1 muestra el modelo general de un sistema de comunicación [7.4], [7.5]. Aquí la señal transmitida,  $Z_T(t)$ , es dada por

$$Z_T(t) = \sum a_n f_T(t - nT) \quad (7.1)$$

donde  $\{a_n\}$  representa los datos complejos transmitidos valuados en el intervalo de símbolo,  $T$  es el espaciado entre símbolos, y  $f_T(t)$  representa la señal de banda base a

ser transmitida [7.4], [7.5]. Seguidamente la señal  $Z_T(t)$  modula en amplitud a una portadora de frecuencia  $f_c$  para producir la señal transmitida  $S_T(t)$  la cual estará dada por :

$$S_T(t) = \text{Re}[Z_T(t)e^{j2\pi f_c t}] \quad (7.2)$$

Subsecuentemente,  $S_T(t)$  se transmite a través de un canal lineal con respuesta a impulso  $C(t)$ , y dado que la señal transmitida es contaminada por ruido aditivo gaussiano  $n(t)$ , la señal recibida  $u(t)$  estará dada por [7.4], [7.5]

$$u(t) = S_T(t) * C(t) + n(t) \quad (7.3)$$

Dado que el modelo del canal no incluye offset de frecuencia ni pequeñas desviaciones de fase como se muestra en la figura 7.1 [7.4], [7.5], la señal demodulada  $S_R(t)$  y su representación en frecuencia  $S_R(\omega)$  están dadas por

$$S_R(t) = S_T(t) * C(t)e^{-j2\pi f_c t} + n(t)e^{-j2\pi f_c t} \quad (7.4)$$

$$S_R(\omega) = \frac{1}{2} Z_T^r(\omega)C(\omega + \omega_c) + \frac{1}{2} Z_T^r(\omega + 2\omega_c)C(\omega + \omega_c) - \frac{1}{2j} Z_T^i(\omega)C(\omega + \omega_c) + \frac{1}{2j} Z_T^i(\omega + 2\omega_c)C(\omega + \omega_c) + N(\omega + \omega_c) \quad (7.5)$$

Seguidamente, la señal demodulada  $S_R(t)$  es pasada a través de un filtro pasa bajas con la respuesta de impulso  $f_R(t)$  para producir la señal  $Z_R(\omega)$  cuya representación de frecuencia dominante esta dada por [7.4]

$$Z_R(\omega) = S_R(\omega)f_R(\omega) \quad (7.6)$$

$$Z_R(\omega) = [\frac{1}{2} Z_T^r(\omega)C(\omega + \omega_c) - \frac{1}{2j} Z_T^i(\omega)C(\omega + \omega_c) + N(\omega + \omega_c)]f_R(\omega) \quad (7.7)$$

Tomando la transformada inversa de Fourier de (7.7) obtenemos la señal recibida,  $Z_R(t)$ , la cual esta dada por ( 7.8)

$$Z_R(t) = \frac{1}{2} Z_T^r(t) * C(t) e^{-j2\pi f_c t} * f_R(t) - \frac{1}{2j} Z_T^i(t) * C(t) e^{-j2\pi f_c t} * f_R(t) + n(t) * f_R(t) \quad (7.8)$$

Definiendo

$$Z_T(t) = a(t) * f_T(t) \quad (7.9)$$

$$Z_T^r(t) = a^r(t) * f_T(t) \quad (7.10)$$

$$Z_T^i(t) = a^i(t) * f_T(t) \quad (7.11)$$

obtenemos

$$Z_R(t) = \frac{1}{2} [a^r(t) * f_T(t) * C(t) e^{-j2\pi f_c t} * f_R(t) + ja^i(t) * f_T(t) * C(t) e^{-j2\pi f_c t} * f_R(t)] + [n(t) * f_R(t)] \quad (7.12)$$

Haciendo

$$h(t) = \frac{1}{2} f_T(t) * f_R(t) * C(t) e^{-j2\pi f_c t} \quad (7.13)$$

$Z_R(t)$  puede ser representado por

$$Z_R(t) = (a^r(t) + ja^i(t)) * h(t) + n(t) * f_R(t) \quad (7.14)$$

Finalmente, ya que

$$a^r(t) + ja^i(t) = \sum (a_n^r + ja_n^i) \delta(t - nT) \quad (7.15)$$

La ecuación (7.14) se puede expresar como

$$Z_R(t) = \sum_{n=0}^{N-1} (a_n^r + ja_n^i)h(t - nT) + w(t) \quad (7.16)$$

donde:

$$w(t) = n(t) * f_R(t) \quad (7.17)$$

Una evaluación realista de cualquier método de igualación depende fuertemente del modelo de canal de comunicación usado a este fin [7.9]-[7.11]. Debido a la intensa investigación que se ha llevado a cabo para modelar adecuadamente los canales de comunicación móviles, se ha encontrado que la señal recibida comúnmente consiste de varias componentes con trayectorias múltiples las cuales pueden pensarse como ondas independientes cuyas fases, amplitudes, retardos y ángulos de incidencia, son variables aleatorias [7.9]. Así el canal móvil de comunicación puede presumirse como un proceso aleatorio que es el resultado de la superposición de dos procesos aleatorios estacionarios. Uno de ellos está relacionado a las fluctuaciones en la señal debidas a las características físicas de medio, por ejemplo en un área urbana es las fluctuaciones están relacionadas con la densidad y altura promedio de edificios o al ancho de las calles. Este procesos suponerse estacionario sobre varios cientos de metros en la banda de frecuencias de 900 MHz [7.9]. El segundo es un proceso aleatorio de corta duración, el cual es relativo al movimiento de la estación móvil. Este se debe principalmente al movimiento del equipo de comunicación y es responsable de la fluctuación del canal de propagación dentro de algunas fracciones de longitud de onda. El proceso aleatorio a corto plazo puede asumirse con estadísticas del tipo de Rayleigh y ser estacionario sobre 4-5 m en la banda de frecuencia de 900 MHz [7.9]. Por otro lado un canal de comunicación móvil en una zona suburbana o rural, puede ser simulado agregando una componente determinística a la respuesta a impulso del canal de comunicación móvil de área urbana descrita en [7.9] y [7.14]. Tal componente determinística está relacionada a la trayectoria principal mediante un factor  $\rho$ , que indica la razón de la componente determinística y la aleatoria [7.9]. Así para valores de  $\rho$  cercano a 0 dB, la contribución aleatoria predomina [7.9], teniendo en tales casos el proceso que caracteriza el canal de comunicación una distribución Rayleigh. Sin embargo un aumento moderado de  $\rho$ , esto es un incremento de  $\rho$  de entre 5 a 10 dB, ocasiona cambios notables en las propiedades estadísticas y dinámicas de la respuesta a impulso del canal [7.9]. Con base en estas interpretaciones cualitativas de algunos resultados experimentales, puede hacerse la siguiente clasificación [7.9], [7.14].

(a) Centro urbano con alta densidad de edificios (>30%),

$\rho = -\infty$  (dB). El canal presenta solamente componentes multitrayectoria

(b) Area urbana con moderada densidad de edificios (entre 20% - 30 %)

$0 \leq \rho \leq 4$  dB.

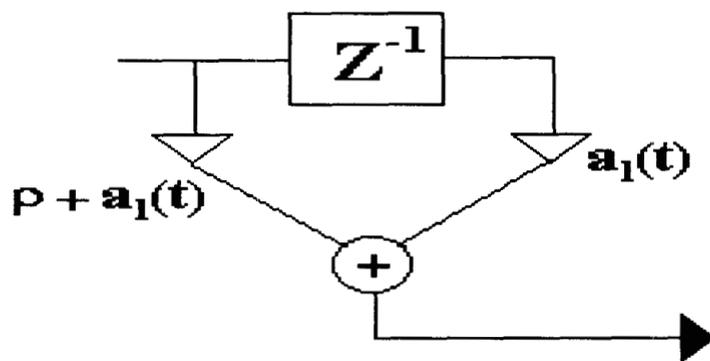


Figura 7.2. Modelo discreto del canal de comunicación con multitrayectoria.

(c) Area urbana con baja densidad de edificios (10% - 20%)

$$4 \leq \rho \leq 6 \text{ dB.}$$

(d) Area suburbana

$$6 \leq \rho \leq 10 \text{ dB.}$$

(e) Area rural abierta

$$\rho > 10 \text{ dB.}$$

La figura 7.2 muestra el modelo de canal que tiene multitrayectoria.

### 7.3. La Estructura Propuesta

Considere el igualador de decisión retroalimentada de tiempo continuo (DFE) mostrado en la figura 7.3(a) cuya señal de salida  $y(t)$  está dada por:

$$y(t) = \sum_{k=0}^{N-1} W_k X_k(t) \quad (7.18)$$

donde  $X_k(t)$  es la señal de salida de un sistema cuya función de transferencia esta dada por

$$H_k(s) = \prod_{i=1}^k \frac{s - \alpha_i}{s + \alpha_i} \quad (7.19)$$

y  $W_k$  es el  $k$ -ésimo coeficiente que se estima tal que la energía del error de estimación  $e(n)$  se reduzca a un mínimo, donde

$$e(t) = d(t) - y(t) \quad (7.20)$$

es el error de salida y  $d(t)$  es la señal de referencia, la cual es un conjunto de símbolos conocidos durante el período de entrenamiento y los símbolos detectados durante el período de operación.

De las ecuaciones (7.18) y (7.20) se obtiene que la energía del error de salida es dada por:

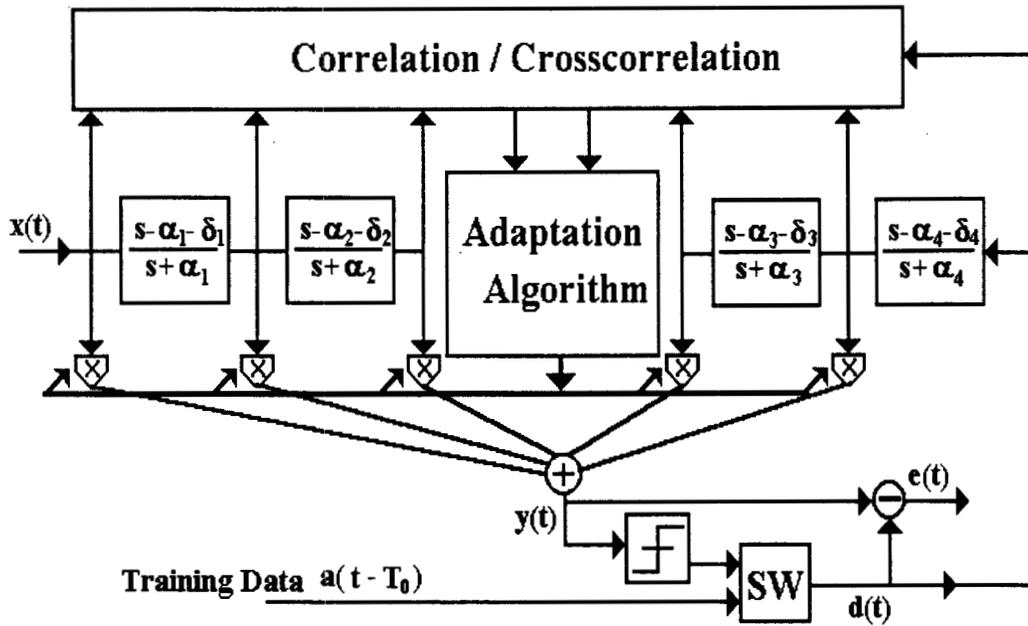


Figura 7.3(a). Estructura propuesta para igualación adaptable analógica de canales de comunicación.

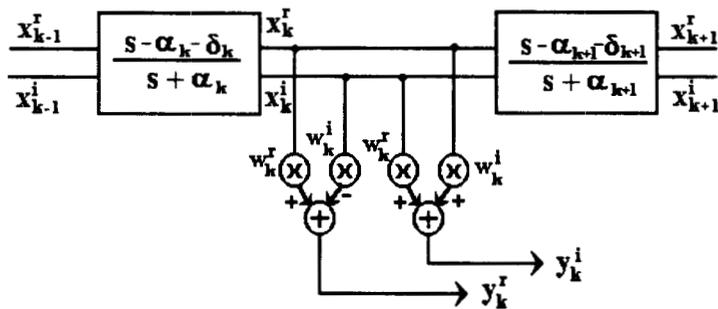


Figura 7.3(b). Línea de retardo con señales y multiplicaciones complejas.

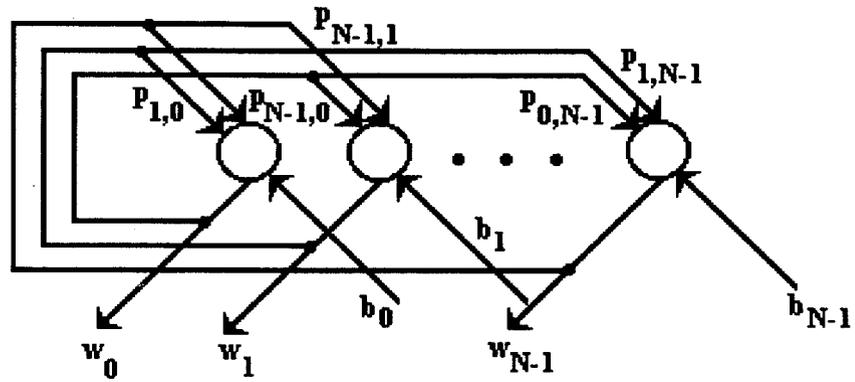


Figura 7.3(c). Red de Hopfield modificada.

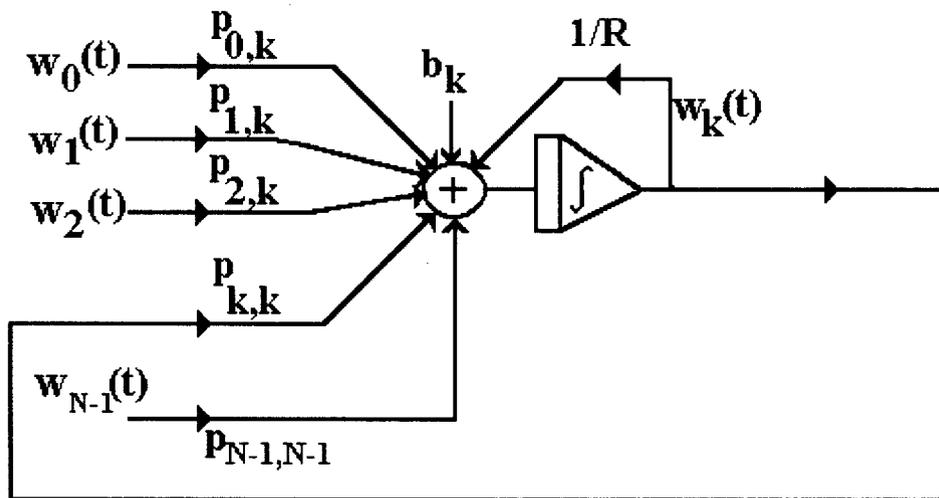


Figura 7.3(d).  $k$ -ésimo nodo de la red de Hopfield modificada con función de activación lineal.

$$\int_0^T d(\tau) - \sum_{k=0}^{N-1} W_k X_k(t) \quad (7.21)$$

Seguidamente, usando el principio de ortogonalidad en mínimos cuadrados, después de algunas manipulaciones se obtiene que el vector óptimo de coeficientes de la estructura propuesta esta dada por [7.2]

$$\mathbf{W} = \Phi_{xx}^{-1} \Phi_{xd} \quad (7.22)$$

donde  $\mathbf{W}$  es el vector de coeficientes cuyo  $k$ -ésimo elemento es  $W_k$ ,  $\Phi_{xd}$  es el vector de correlación entre las señales de salida de las secciones pasa-todo y la señal de referencia cuyo  $j$ -ésimo elemento es dado por

$$\Phi_{xd}(j) = \int_0^T d(\tau) X_j^*(\tau) d\tau \quad (7.23)$$

donde  $*$  denota la operación complejo conjugado, y  $\Phi_{xx}$  es la matriz de correlación entre las secciones de salida pasa-todo donde el  $(j,k)$ -ésimo elemento está dado por

$$\Phi_{xx}(j, k) = \int_0^T X_j(\tau) X_k^*(\tau) d\tau \quad (7.24)$$

Con el fin de derivar un algoritmo adaptable para la estimación en línea del vector de coeficientes  $\mathbf{W}$ , considere una red neuronal de Hopfield modificada de tiempo continuo como la mostrada en las figuras 7.3c-7.3d cuya señal de salida esta dada por [7.25], [7.28]:

$$c \frac{dw_k(t)}{dt} = -\frac{1}{R} w_k(t) + \sum_{m=0}^{N-1} P_{m,k} w_m(t) + b_k, \quad k = 1, 2, 3, \dots, n-1 \quad (7.25)$$

donde  $w_k(t)$  es la señal de salida del  $k$ -ésimo nodo,  $R$  y  $c$  son constantes positivas reales,  $P_{m,k}$  es el peso de la trayectoria que va del  $m$ -ésimo nodo al  $k$ -ésimo nodo y  $b_k$  es una constante real que se conoce como sesgo o "bias".

Tomando la transformada de Laplace de (7.25) tenemos que

$$SW(S) + \frac{1}{RC} W(S) + \frac{1}{C} PW(S) = \frac{1}{C} B$$

Seguidamente, usando el Teorema del valor Final se obtiene que, después de un intervalo de tiempo suficientemente largo, la salida de la red neuronal Hopfield modificada estará dada por [7.25]

$$W(\infty) = [I - RP]^{-1} RB \quad (7.27)$$

Considere seguidamente el vector de coeficientes óptimo del filtro adaptable el cual está dado por la ecuación (7.22), y asuma que:

$$P = \frac{(I - \Phi_{xx})}{R} \quad (7.28)$$

y

$$B = \frac{\Phi_{xd}}{R} \quad (7.29)$$

Entonces de las ecuaciones (7.27)-(7.29) se sigue que, después de la convergencia ha sido obtenida, la red neuronal de Hopfield modificada, proporciona la solución óptima de la ecuación de Wiener-Hopf y entonces esta se puede usar para la estimación del vector de coeficientes óptimo de un filtro adaptable de tiempo continuo [7.21].

#### 7.4. Simulaciones Computacionales

Los resultados de simulación presentados en esta sección intentan mostrar la pertinencia y funcionamiento del sistema propuesto. Con este fin, se usarán las ideas descritas en la sección 7.2 con el fin de simular los canales de radio móvil con tres trayectorias de propagación, donde el comportamiento en tiempo del canal de comunicación está caracterizado por la mitad de la dispersión Doppler. Esto es, el valor de la variación del canal multitrayectoria estará dado por [7.15]:

$$f_D = f_c \frac{V}{C} \quad (7.30)$$

donde  $f_c$  es la frecuencia de portadora,  $V$  es la velocidad móvil y  $C$  es la velocidad de la luz. Aquí el equivalente discreto del canal (7.16), (7.17) estará formado por una

combinación lineal de 3 secuencias básicas (una trayectoria directa y dos reflejadas) como se muestra en la figura 7.4. Esto es [7.14]:

$$h(k,r) = \theta_r f_r(k), \quad r = 0,1,2 \quad (7.31)$$

donde

$$f_1(k) = 1 \quad (7.32)$$

$$f_0(k) = e^{j2\pi k m_1} \quad (7.33)$$

y

$$f_2(k) = e^{j2\pi k m_2} \quad (7.34)$$

donde para un tiempo rápido de variación del canal  $m_1$ , y  $m_2$  se asumen iguales a 120 y 200 respectivamente, y  $\theta_0 = 0.5$ ,  $\theta_1 = 1$ ,  $\theta_2 = 0.5$ . Estos números son bastante realistas para una frecuencia de portadora de 900 MHz, a una velocidad de transmisión de 20Kb/s y una velocidad del vehículo igual a 100 Km/h. La entrada fue una serie de símbolos 4-QAM que se pasaron a través del canal y se contaminado con ruido aditivo de distribución Gaussiano (additive white Gaussian noise, AWGN). Ningún código de corrección de errores fue usado. La estructura de la señal transmitida consiste de trenes de 2 datos de prueba y 6 datos de información.

Para evaluar el funcionamiento de la estructura propuesta, (figura 7.3), por medio de simulación, se toma en cuenta que ésta se trata de un sistema de tiempo continuo. Así el valor de la frecuencia de operación del DFE propuesto se asumió ser mucho mayor que la velocidad de símbolo y entonces el retardo entre dos coeficientes consecutivos del DFE propuesto deberá ser mayor que el período de muestreo. Con esto en mente, la etapa de filtrado de la estructura DFE de tiempo continuo propuesta se transformó en una estructura de tiempo discreto por medio de la transformación bilineal, obteniéndose el filtro de tiempo discreto mostrado en la figura 7.5. En la cual, con el fin de llevar a cabo la simulación computacional correspondiente, y debido a las limitaciones de tiempo, el periodo de muestreo de la estructura propuesta se asumió 10 veces menor que la duración de símbolo, esto es, la frecuencia de operación del sistema es 10 veces la frecuencia de símbolo, aunque en una situación real este número sea teóricamente infinito. Así, el número de secciones pasa-todo de primer orden insertadas entre las secciones dos coeficientes consecutivos del DFE fue de 10. Para simular las características no ideales de la línea de retardo, se agregó cierta desviación a la posición del cero en las secciones del filtro pasa-todo de primer orden, así como también una pérdida de aproximadamente 0.01 dB entre dos coeficientes consecutivos del filtro como se muestra en la figura 7.5. Estas

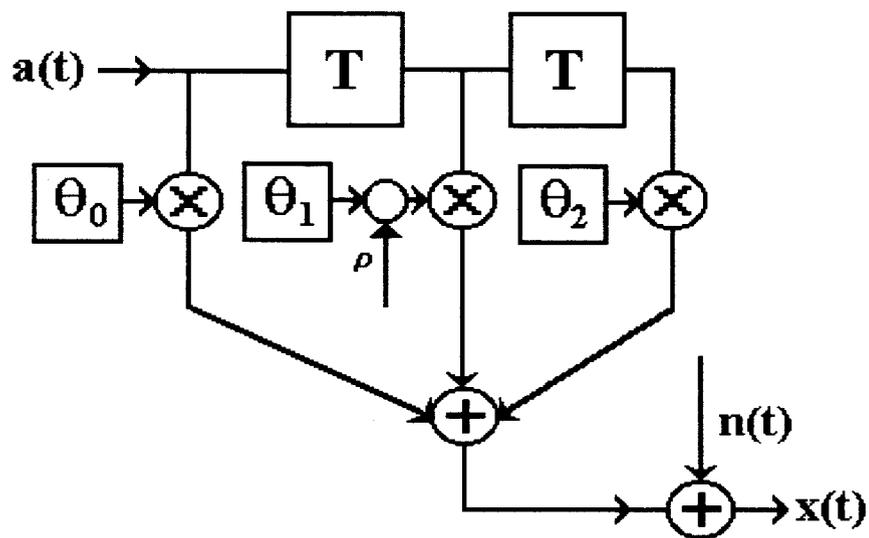


Figura 7.4. Equivalente discreto del canal de comunicación con coeficientes variables con el tiempo.

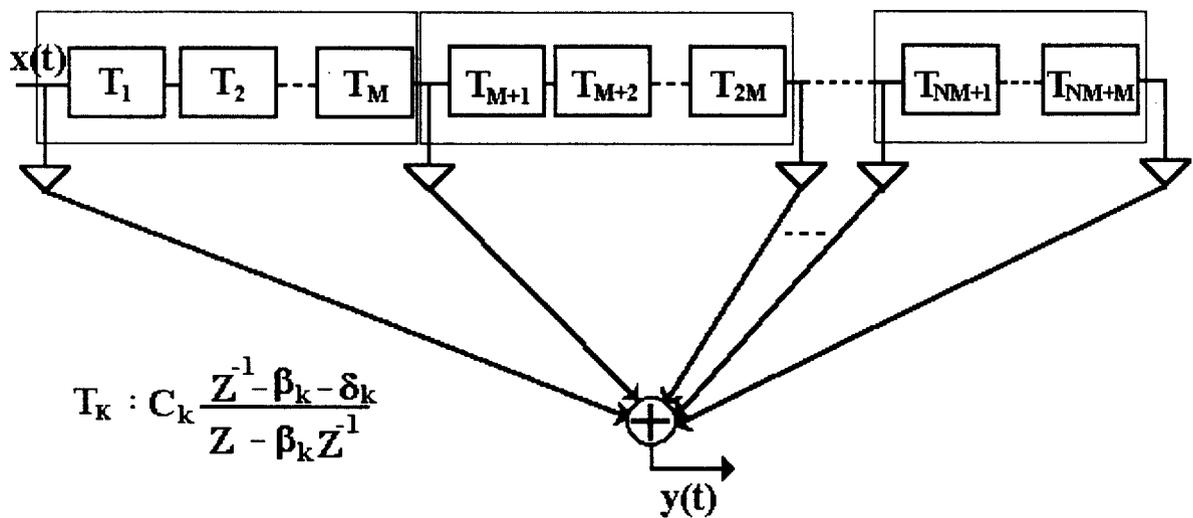


Figura 7.5. Filtro discreto empleado para simular el comportamiento de un sistema analógico.

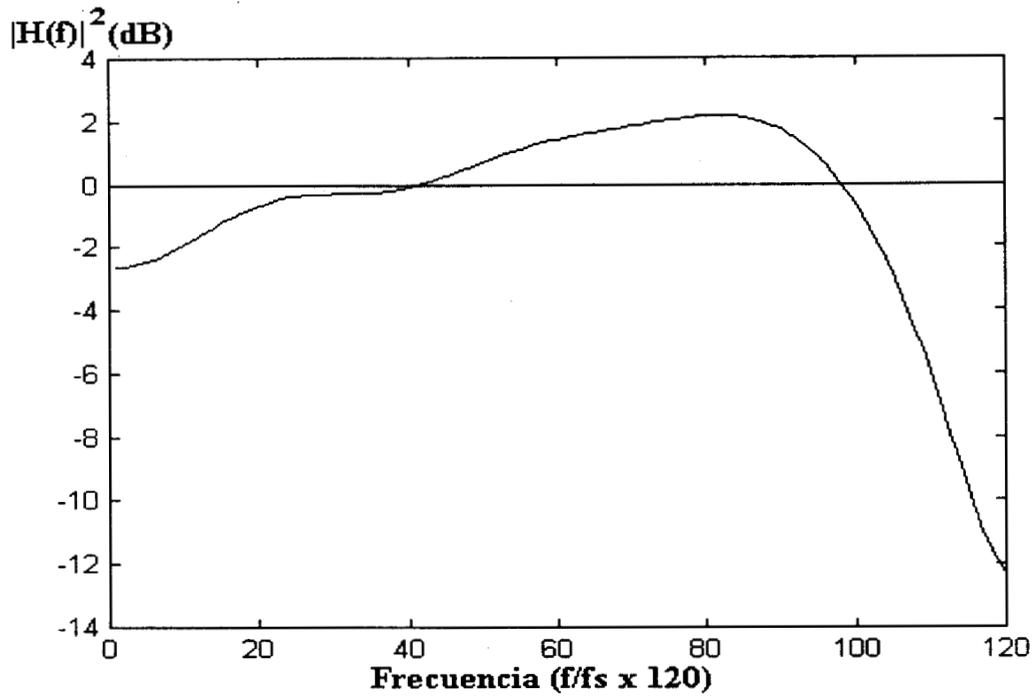


Figura 7.6. Magnitud de la respuesta en frecuencia de un canal de telefónico de alta calidad.

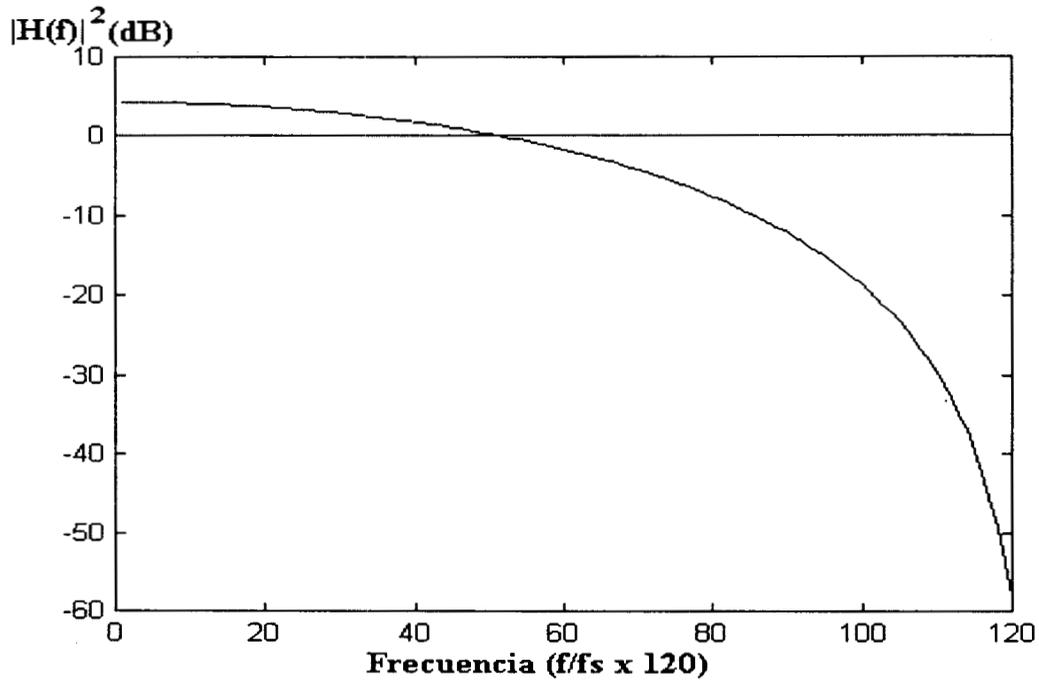


Figura 7.7. Magnitud de la respuesta en frecuencia de un canal de comunicación con severa distorsión espectral, típicamente encontrado en comunicación móvil.

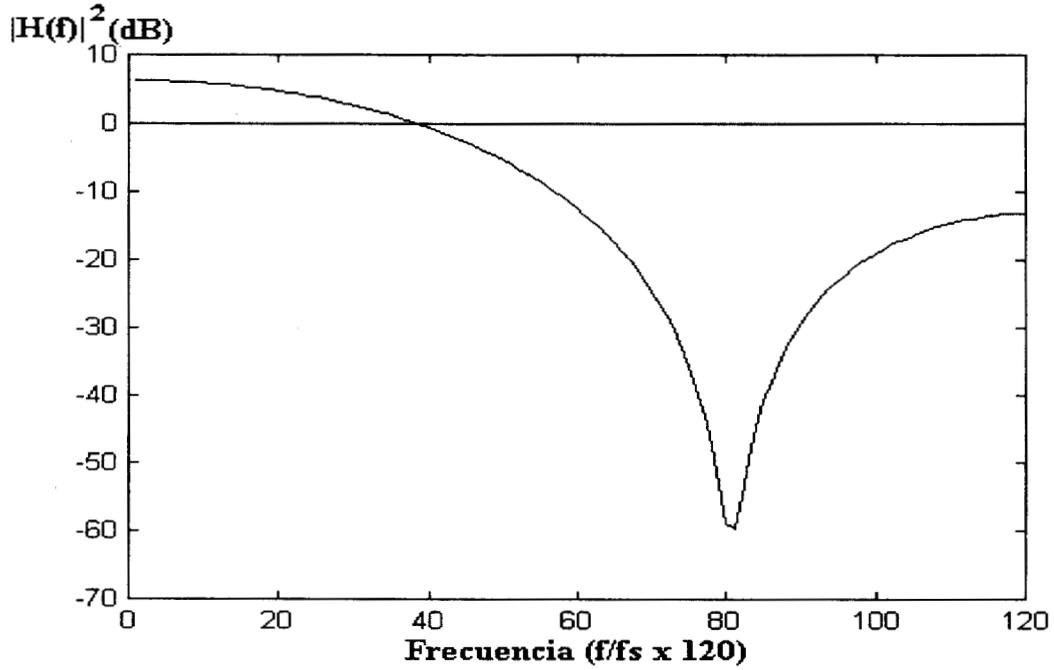


Figura 7.8. Magnitud de la respuesta en frecuencia de un canal de comunicación con severos huecos espectral, frecuentemente encontrados en comunicación móvil.

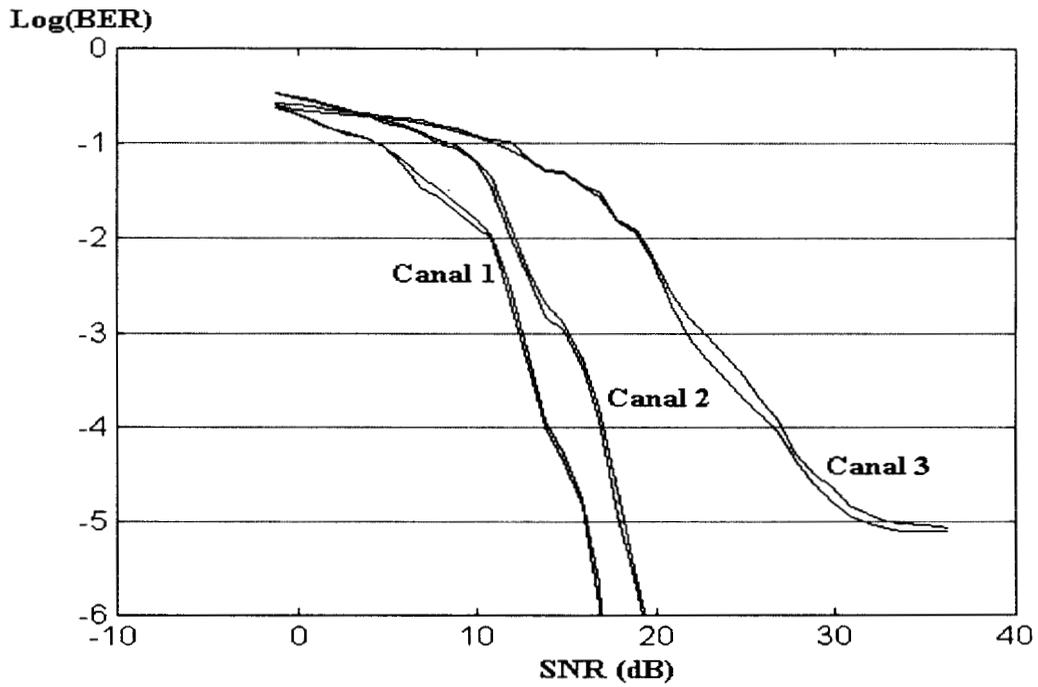


Figura 7.9. Numero promedio de errores en la recepción obtenidos usando la estructura propuesta y un igualador DFE adaptado usando el algoritmo RLS, cuando ambos son requeridos a igualar los tres canales invariantes en el tiempo mostrados en las figuras 7.6 –7.8.

pérdidas son las mismas que introducen frecuentemente los seguidores de voltaje, frecuentemente usados en líneas de retardo analógicas [7.21], [7.24]. La red neuronal de Hopfield fue simulada resolviendo su ecuación diferencial usando el método de Runge-Kutta de cuarto orden. Aquí los integradores requeridos para estimar las funciones autocorrelación y la correlación cruzada fueron reemplazados por filtros pasa bajas. El ruido aditivo fue una secuencia de ruido blanco.

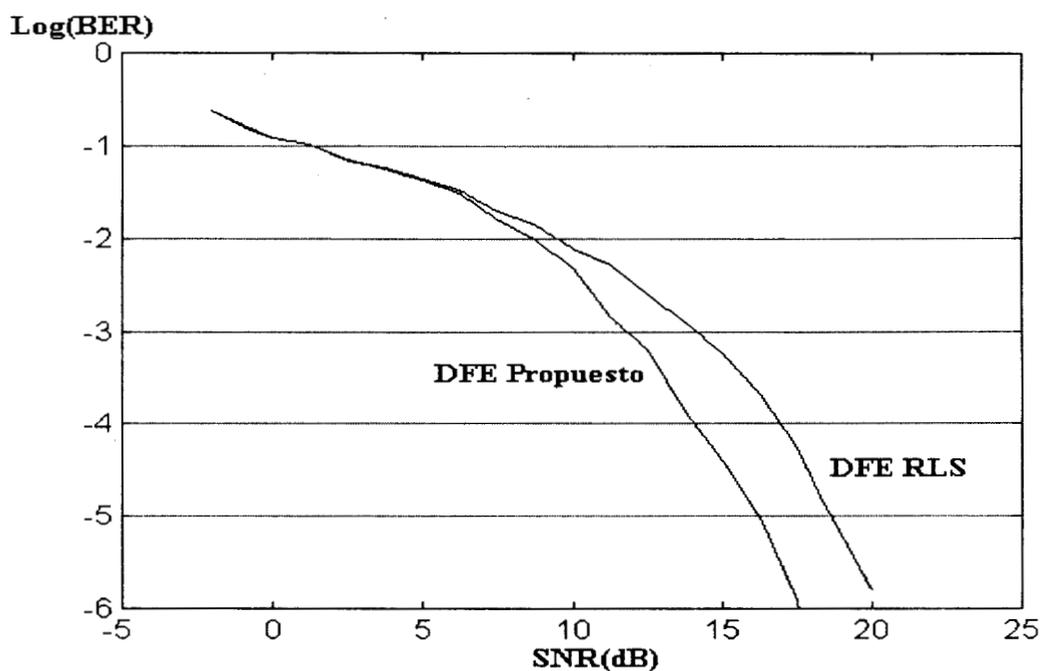
Tres casos diferentes fueron considerados para la evaluación del sistema, las cuales son: canales de comunicación estacionarios, canales de comunicación de lentamente variable con el tiempo, canales de comunicación con rápidas variaciones con el tiempo.

### **Caso 1: Canal de Comunicación Estacionario**

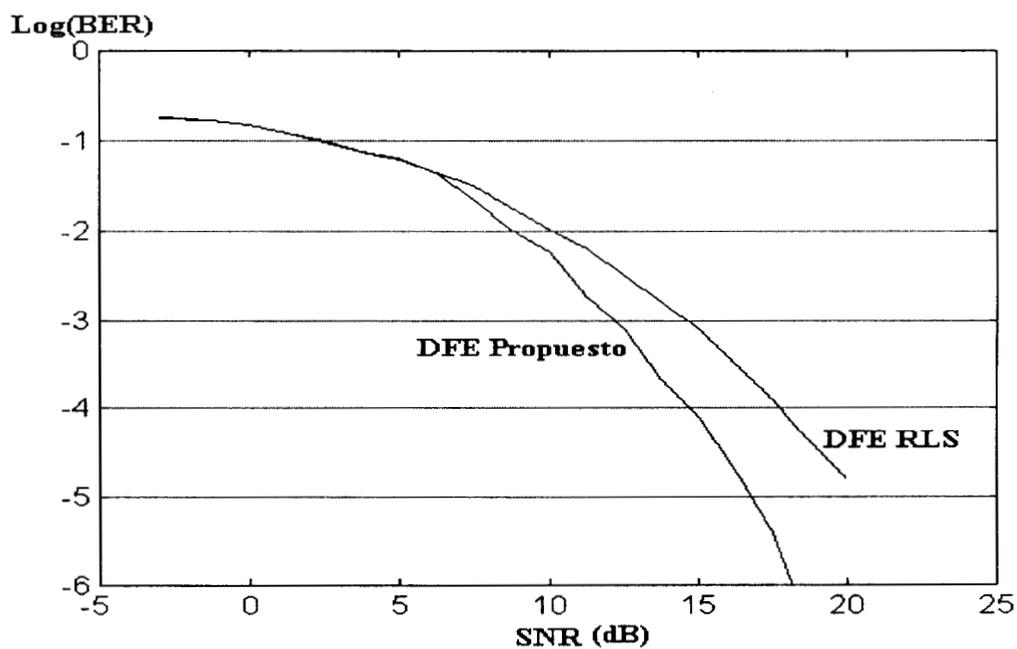
El funcionamiento de la estructura propuesta se evaluó usando tres diferentes canales estacionarios, cuyas densidades espectrales de potencia se muestran en las figuras 7.6 – 7.8 respectivamente. La respuesta a impulso del canales de comunicaciones mostrado en la figura 7.6 (canal 1) corresponde a un canal de alta calidad, típica de un canal telefónico, que es relativamente fácil de compensar por el DFE adaptable [7.2]. En contraste, los canales de comunicación 2 (figura 7.7) y 3 (la Figura 7.8) tienen profundos huecos espectrales que ocasionan serias distorsiones en la información. Las características espectrales de los canales mostrados en las figs. 7.7 y 7.8, a menudo encontradas en canales de comunicación móvil con multitraectorias, causan seria distorsión a las señales transmitidas [7.2]. La figura 7.9 muestra el número promedio de errores (BER) de la estructura propuesta y el DFE convencional que usan el algoritmo RLS [7.2] cuando ambos son requeridos igualar los tres canales de comunicación mencionados anteriormente. El factor de olvido [7.2] del algoritmo RLS es igual a 0.99.

### **Caso 2: Canal de Comunicación con Variaciones Lentas.**

Seguidamente evaluamos la estructura propuesta cuando es requerida a evaluar canales de dos canales de comunicación que varían lentamente con el tiempo. Inicialmente se empleamos el método de "Snapshot" [7.29], en el cual asumimos que el canal permanece constante durante un tiempo igual a la duración de 100 símbolos. Aquí la respuesta a impulso del canal de comunicación equivalente es dado por las ecuaciones (7.31)-(7.34), con  $m_1=120$  y  $m_2=200$  y  $k$  esta dada por la parte entera de  $u/100$ , donde  $u$  es el número del símbolo. El resultado de simulación se muestra en la figura 7.10, donde el factor de olvido usado para el algoritmo RLS es igual a 0.99. La figura 7.11 muestra el funcionamiento del DFE propuesto y el DFE convencional usando el algoritmo RLS cuando ambos son requeridos a igualar continuamente en un canal de comunicación el cual varía lentamente con el tiempo, donde el canal equivalente es dado por las ecuaciones (7.31)-(7.34), con  $m_1 = 1200$ ,  $m_2 = 2000$  y  $k$  es el número de símbolo. Un factor de olvido igual a 0.99 se usó en el algoritmo RLS. La figura 7.12 muestra el trazo de los coeficientes teóricos y estimados usando la estructura propuesta en un canal que varía lentamente con el tiempo, mientras que la Fig. 7.13 muestra el trazo de los coeficientes cuando se usa el algoritmo



**Figura 7.10.** BER obtenido cuando la estructura propuesta y la convencional usando el algoritmo RLS son requeridas a igualar un canal de comunicación que varía lentamente con el tiempo. La simulación se llevo a cabo usando el método conocido como snapshot.



**Figura 7.11.** BER obtenido cuando la estructura propuesta y la convencional usando el algoritmo RLS son requeridas a igualar un canal de comunicación que varía lenta pero continuamente con el tiempo.

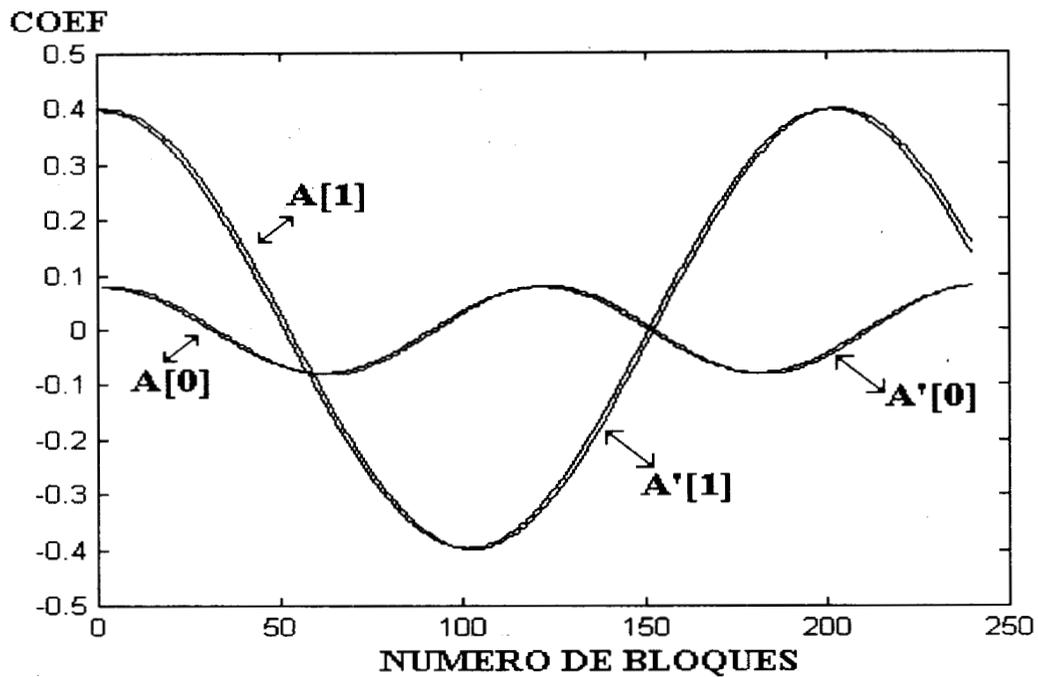


Figura 7.12. Trayectoria de los coeficientes teóricos  $A[i]$  y de los estimados  $A'[i]$  usando la estructura propuesta cuando esta es requerida a igualar un canal lentamente variable con el tiempo.

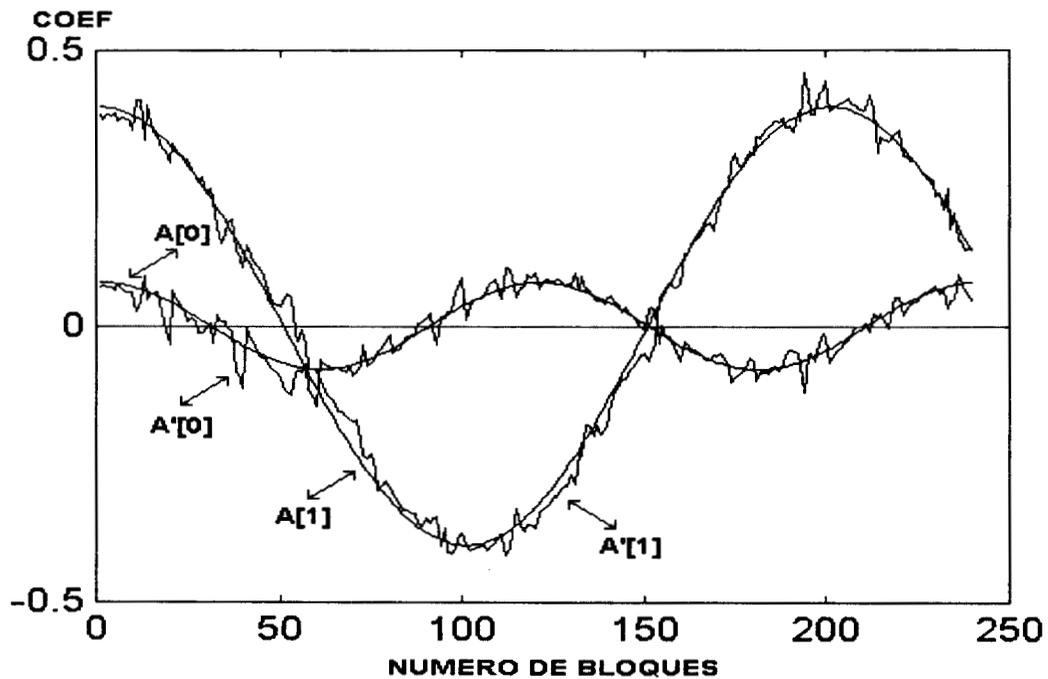


Figura 7.13. Trayectoria de los coeficientes teóricos  $A[i]$  y de los estimados  $A'[i]$  usando la estructura DFE convencional, con el algoritmo RLS, cuando esta es requerida a igualar un canal lentamente variable con el tiempo.

RLS convencional. Estas figuras muestran que la estructura propuesta se funciona mejor que el DFE convencional usando el algoritmo RLS cuando el canal de comunicación varía lentamente en el tiempo.

### **Caso 3: Canal de Comunicación con Variaciones Rápidas**

Seguidamente evaluamos la convergencia del DFE propuesto y lo comparamos con el funcionamiento del DFE convencional cuando se requiere que ambos iguallen canales de comunicación los cuales presentan variaciones rápidas con el tiempo, frecuentemente encontrados en sistemas móviles de comunicación. La respuesta de impulso de canal varía según las ecuaciones, (7.31)-(7.34) en el rango de muestra. Se asume que  $\rho = 3$  dB el cual puede considerarse como una situación estándar en la Ciudad de México. El muestreo de la señal de entrada del DFE convencional se asume igual a la velocidad de símbolo, mientras en el DFE el propuesto, tomando en la cuenta que se trata de un sistema de tiempo continuo, la velocidad de muestreo se asumió igual a 10 veces la velocidad del símbolo. En ambos casos el canal de comunicación varía continuamente con el tiempo. La figura 7.14 muestra el promedio bits erróneos (BER) del DFE propuesto junto con el BER proporcionado por el DFE convencional adaptado usando el algoritmo RLS. El factor de olvido del algoritmo RLS es igual a 0.9, el cual es el mejor valor según las simulaciones mostrada en la figura 7.15. Las figuras 7.16 y 7.17 que muestran el trazo de coeficientes teóricos y estimados obtenidos usando los algoritmos propuesto y convencional operando en un ambiente que varía rápidamente con el tiempo. Las figuras 7.16 y 7.17 muestran que la estructura propuesta presenta un funcionamiento considerablemente mejor que el de la estructura DFE convencional de tiempo discreto con adaptación del algoritmo RLS.

## **7.5. Conclusiones**

En este capítulo se propuso un igualador de tiempo continuo de decisión retroalimentada basado en una red de Hopfield modificada de tiempo continuo, para igualación de canales de comunicación móvil variables con el tiempo, en el cual la señal de salida del DFE se obtiene en tiempo continuo, y una red neuronal Hopfield modificada de tiempo continuo se usa para actualizar el vector de coeficientes de la estructura del DFE. Así la salida del DFE y la actualización del vector de los coeficientes puede obtenerse en menos tiempo y con menor consumo de potencia que el requerido por sus contrapartes digitales. El funcionamiento de la estructura propuesta fue evaluado y comparado con el funcionamiento del DFE convencional usando el algoritmo RLS para la adaptación del vector de coeficientes. Para realizar la evaluación del sistema se asumen tres casos diferentes: canales de comunicación invariantes con el tiempo (estacionarios), canales de comunicación con variaciones lentas con el tiempo y canales de comunicación con variaciones rápidas con el tiempo. En todos los casos se asumió un esquema de modulación 4-QAM. Estos resultados muestran que ambos, el DFE propuesto y convencional, se comportan de una manera similar cuando ambos son requeridos a igualar canales de comunicación estacionarios. Sin embargo el DFE propuesto presenta un mejor funcionamiento que el convencional cuando ambos son requeridos a igualar canales de

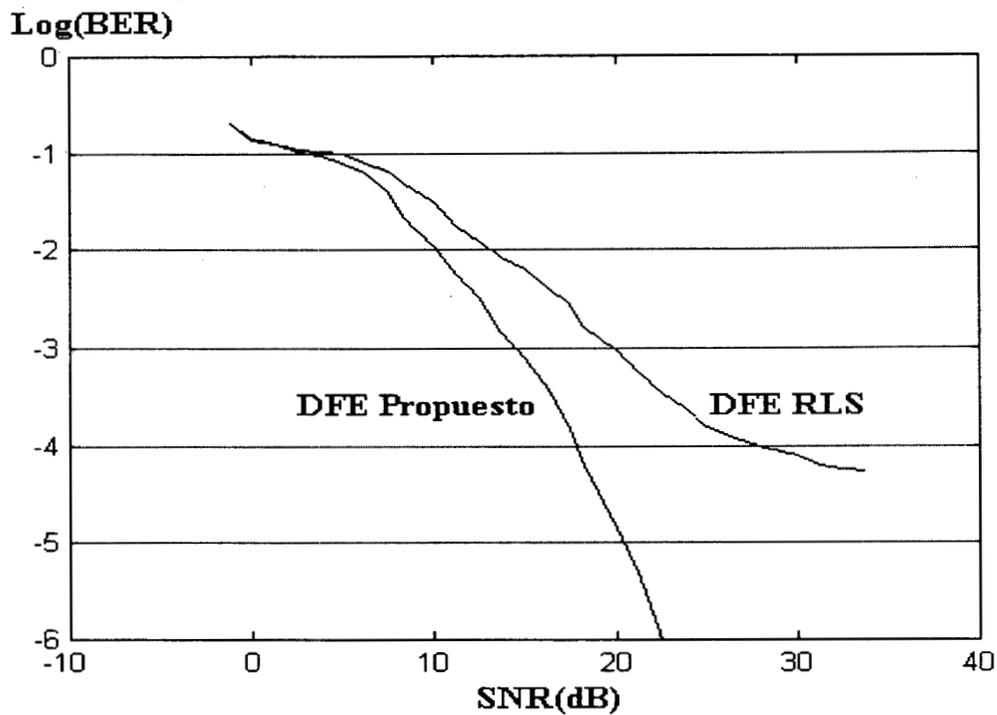


Figura 7.14. BER obtenido cuando la estructura propuesta y la convencional usando el algoritmo RLS son requeridas a igualar un canal de comunicación que varía rápida y continuamente con el tiempo.

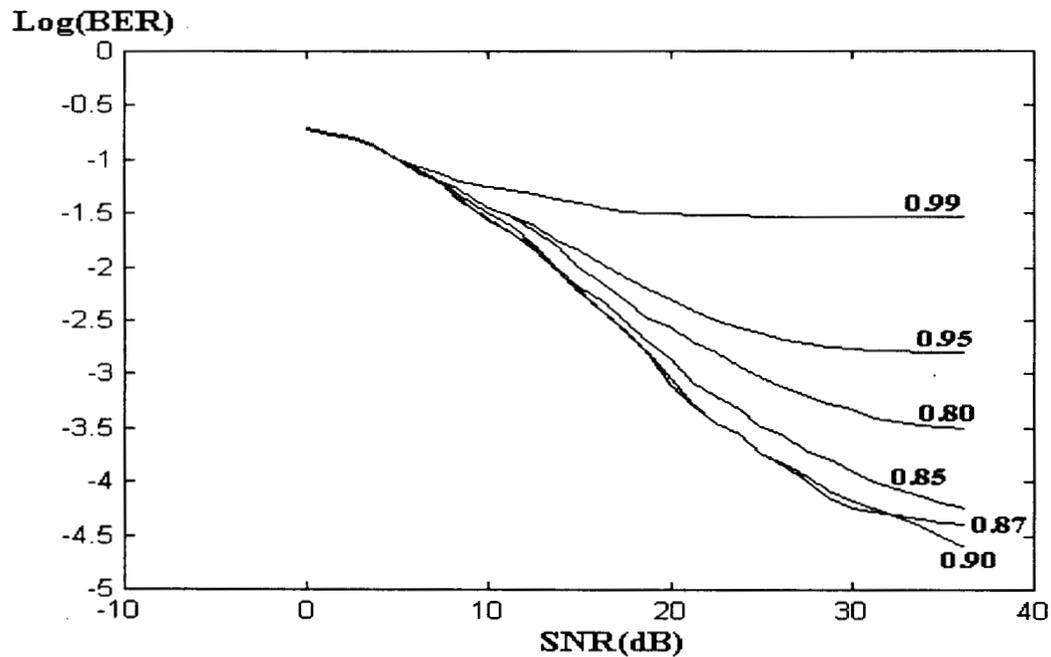


Figura 7.15. BER obtenido cuando la estructura convencional usando el algoritmo RLS con diferentes factores de olvido, es requerida a igualar un canal de comunicación que varía rápida y continuamente con el tiempo.

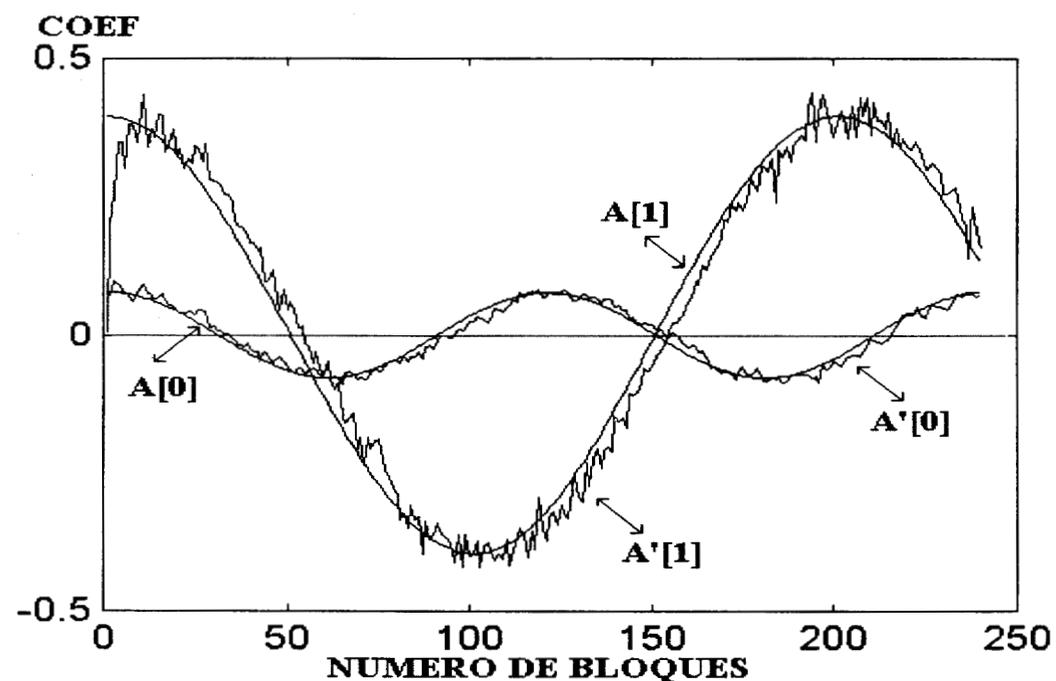


Figura 7.16. Trayectoria de los coeficientes teóricos  $A[i]$  y de los estimados  $A'[i]$  usando la estructura propuesta cuando esta es requerida a igualar un canal lentamente variable con el tiempo.

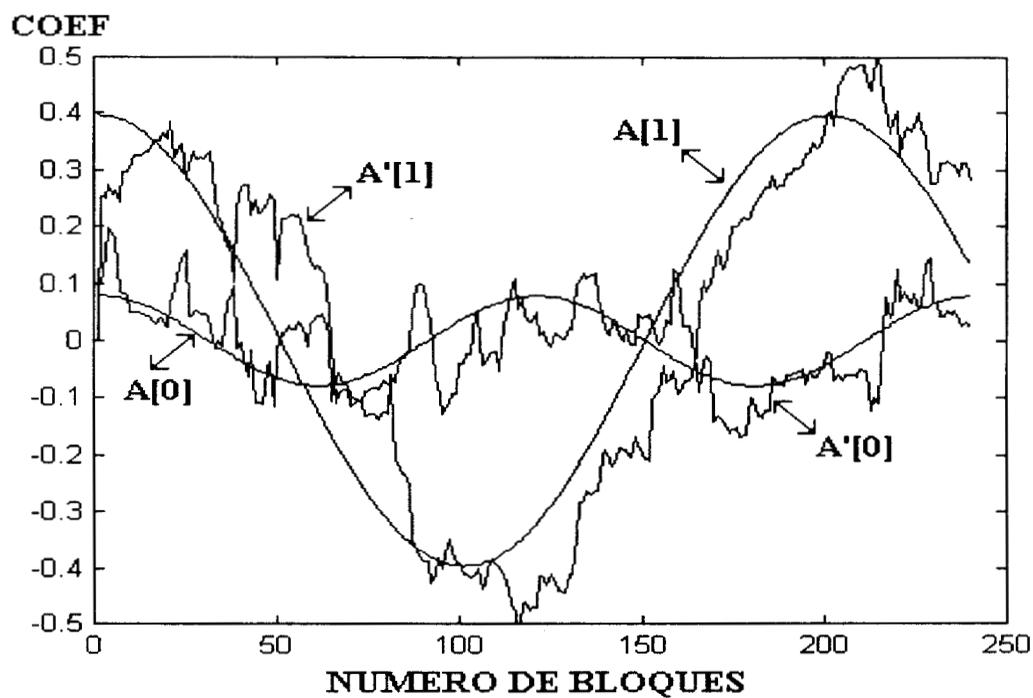


Figura 7.17. Trayectoria de los coeficientes teóricos  $A[i]$  y de los estimados  $A'[i]$  usando la estructura DFE convencional, con el algoritmo RLS, cuando esta es requerida a igualar un canal lentamente variable con el tiempo.

comunicación variantes en el tiempo. Esto es porque el esquema propuesto esta operando en modo de tiempo continuo y entonces se actualiza más frecuentemente. Así tiene potencialmente velocidades de convergencia mucho más rápidas y entonces una mayor habilidad para igualar canales que varían rápidamente con el tiempo. Además presenta también un consumo de potencia potencialmente menor y menor tamaño.

## Conclusiones

Los filtros adaptables han tenido un gran impacto en la solución de una amplia variedad de problemas prácticos en diversos campos de las ciencias e ingeniería debido, principalmente, a sus características que les permiten adaptarse a diversas situaciones las cuales no pueden determinarse de antemano. Debido a lo anterior los sistemas adaptables han sido objeto de intensa investigación durante las dos últimas décadas, habiendo sido propuestos durante este tiempo un número importante de estructuras y algoritmos de aprendizaje. Sin embargo, pese a esta intensa investigación y al número importante de propuestas de estructuras y algoritmos, aún permanecen por resolver gran cantidad de problemas en el campo de los filtros adaptables.

Para resolver algunos de los problemas que permanecen aun sin solución en el campo de los filtros adaptables, cuya solución es de gran importancia para el avance del conocimiento en este campo, se ha realizado una amplia investigación tendiente a la búsqueda de soluciones a los problemas antes mencionados, recopilándose los resultados mas trascendentales en la presente tesis doctoral. Así durante la realización de la presente investigación se propuso un algoritmo, basado en la búsqueda de gradiente con factor de convergencia variable, con el fin de mejorar las características de convergencia de este tipo de algoritmos, mayor velocidad de convergencia y menor error residual del sistema, manteniendo al mismo tiempo una complejidad computacional razonablemente baja, proporcional al orden del filtro, típicamente  $2N+k$  donde  $N$  es el orden del filtro y  $k$  una constante. Sin embargo, aunque este algoritmo reduce la distorsión debida a fluctuaciones en la potencia del ruido aditivo, tiene una complejidad computacional que podría ser alta para aplicaciones en las cuales el filtro adaptable requiriera varios cientos de coeficientes. Con este fin se desarrollo un filtro en el dominio de la frecuencia el cual simultáneamente reduce el retardo de bloque introducido por este tipo de sistemas e incrementa considerablemente su velocidad. Se propuso además una estructura para filtrado adaptable que emplea un algoritmo de adaptación basado en redes neuronales, el cual puede implementar tanto en forma analógica como en forma digital. También se propusieron dos estructuras de redes neuronales para reconocimiento de caracteres manuscritos. Todas las propuestas realizadas se han evaluado extensivamente, probando su eficiencia, por medio de simulación computacional. Los resultados obtenidos se han mostrado en los diversos capítulos que componen esta tesis doctoral.

## 8.1 Introducción

Los sistemas adaptables han despertado interés entre la comunidad científica debido sus posibles aplicaciones para resolver gran cantidad de problemas prácticos en el campo de la ciencias e ingeniería. Durante las últimas dos décadas se han propuestos un número importante de estructuras y algoritmos de adaptación para la solución de diversos problemas, sin embargo existen un número importante de problemas aún sin resolver. A los problemas aun sin solución, se ha aumentado la necesidad o requerimiento de un mejor funcionamiento de los sistemas adaptables, lo que ha llegado a ser posible debido al avance de la tecnología en varios campos tales como computación, comunicaciones, control, etc.

La presente investigación doctoral se ha realizado para proponer soluciones a tres de los problemas mas importantes que aun permanecen sin resolver en le campo de los sistemas adaptables como son: a) Mejorar el funcionamiento de los sistemas adaptables, basados en la búsqueda del gradiente, cuando operan en situaciones en la cuales existen fluctuaciones en la potencia del ruido aditivo, sin reducir en forma significativa la velocidad de convergencia ni incrementar su complejidad computacional. b) Reducir el retardo de bloque e incrementar la velocidad de convergencia de los algoritmos de bloque operando en el dominio de la frecuencia. c) Reducir problemas de estabilidad numérica en algoritmos del tipo RLS, posibilitando además su realización analógica. Durante realización de esta investigación, se han propuesto tres algoritmos de adaptación para filtros adaptables los cuales han publicados en revistas internacionales con arbitraje estricto.

En este capitulo se presentan las conclusiones de cada sistema propuesto durante la investigación doctoral realizada. Algunas de las conclusiones presentadas en este capitulo ya se han mencionado anteriormente en la última sección de cada unos de los capítulos 4, 5, 6 y 7. Sin embargo se considera conveniente retomar algunas de ellas con el fin de presentar una conclusión integral del presente trabajo. Junto con las conclusiones de cada sistema propuesto, se mencionará además la posible línea de investigación que podría derivarse de este en futuro cercano.

## 8.2 Conclusiones y Línea de Investigación Futura

El capítulo 4 propuso un algoritmo adaptable LMS normalizado de paso variable, para aplicaciones en las cuales se presentan fluctuaciones en la potencia del ruido aditivo, o la potencia del ruido aditivo es grande, tal como sucede en ocasiones tanto en cancelación de ruido como de eco. El algoritmo propuesto se basa en el hecho de que para evitar distorsión durante intervalos de tiempo en los cuales la potencia del ruido aditivo se incrementa, el factor de convergencia debe disminuir en la misma proporción en la que el ruido aditivo aumenta. El algoritmo propuesto se ha evaluado y comparado con otros algoritmos propuestos hasta la fecha, tales como el NLMS, el VSLMS, el NACFLMS y el VECLMS y el CCLMS, usando simulación computacional. Los resultados obtenidos muestran que el algoritmo propuesto reduce de manera significativa, la distorsión que se presenta en los algoritmos convencionales cuando la potencia del ruido aditivo se incrementa, sin reducir la capacidad del algoritmo para seguir variaciones en las características de la señales de referencia. Este hecho indica que el algoritmo propuesto

presenta mejores características que otros algoritmos propuestos hasta la fecha. Este algoritmo tiene una complejidad proporcional al orden del filtro la cual podría ser alta en aplicaciones que requieren filtros de alto orden. Sin embargo, para este tipo de aplicaciones, en el capítulo 5 se presenta un algoritmo adaptable con una menor complejidad computacional operando en bloque, con una razonablemente alta velocidad de convergencia.

El capítulo 5 presentó un algoritmo para procesamiento adaptable en bloque en el dominio de la frecuencia, el cual reduce el retardo de bloque introducido por éste tipo de sistemas. Para obtener un menor retardo de bloque la salida del filtro fue calculada como la suma de  $M$  convoluciones parciales. Cada una de esas convoluciones fue calculada usando el método conocido como "Overlap-Save" con 50% de traslape para reducir la complejidad computacional del algoritmo. Un factor de convergencia cercanamente óptimo para el algoritmo basado en la búsqueda del gradiente es estimado usando el método de los periodogramas modificados promediados con 50% de traslape. Este hecho permite estimar el factor de convergencia usando los mismos datos obtenidos de las transformaciones efectuadas para el cálculo de la salida del sistema.

El algoritmo propuesto se ha evaluado usando simulación computacional y comparado con los algoritmos convencionales, tales como FLMS, MDF, FBAF y JALG. Los resultados obtenidos muestran que el algoritmo propuesto proporciona mucha mayor velocidad de convergencia que el algoritmo FLMS, con un retardo de bloque mucho menor, aunque su complejidad computacional es tanto mayor, especialmente cuando de algoritmo propuesto opera con bloque relativamente cortos. El algoritmo propuesto también proporciona una velocidad de convergencia mayor que los algoritmos MDF y FBAF con el mismo retardo de bloque y similar complejidad computacional. Este también presenta una velocidad de convergencia mayor que el algoritmo JALG cuando  $M > 1$ , debido a que el vector de coeficientes es adaptado mas frecuentemente. Sin embargo en esta situación la complejidad del algoritmo propuesto es un tanto mayor. Resultados obtenidos muestran también que la velocidad de convergencia del algoritmo propuesto se incrementa cuando el retardo de bloque disminuye, debido a que en esta situación el vector de coeficientes es adaptado mas frecuentemente. Sin embargo esta reducción en el retardo de bloque e incremento en la velocidad de convergencia resulta en una mayor complejidad computacional. Así hay un compromiso entre la velocidad de convergencia, el retardo de bloque y la complejidad computacional de algoritmo propuesto.

Finalmente, en el capítulo 6 se propuso una estructura para filtrado adaptable la cual es realizable tanto en forma analógica como en forma digital. La adaptación de la estructura propuesta se lleva a cabo en tiempo continuo usando una red neuronal de Hopfield modificada. El sistema se evaluó asumiendo una realización analógica y se compara con el funcionamiento obtenido por una estructura transversal adaptada usando un algoritmo RLS convencional. El sistema propuesto se puede decir que es una realización del algoritmo tipo RLS, sin embargo este no presenta problemas de estabilidad, ya que no requiere ninguna inversión de matriz explícita para realizar el algoritmo RLS.

Para evaluación del sistema propuesto, este operó como identificador de sistemas y como predictor lineal. Los resultados obtenidos muestran que el sistema propuesto, operando

como identificador de sistemas en condiciones estacionarias, presenta una velocidad de convergencia inicial similar a la proporcionada por el algoritmo RLS convencional. Aunque el sistema propuesto presenta un menor desajuste que el presentado por el algoritmo RLS convencional debido al sobre muestreo introducido en la realización analógica. Además, cuando el sistema propuesto opera en condiciones en las cuales las señales de entrada y referencia son no estacionarias, como se encuentra frecuentemente en los sistemas de comunicación móviles, éste presenta características de convergencia superiores a proporcionadas por el algoritmo RLS convencionales, especialmente cuando las estadísticas de las señales involucradas varían rápidamente con el tiempo.

Las simulaciones computacionales realizadas con el sistema propuesto operando como predictor lineal muestran un funcionamiento muy similar al del algoritmo RLS convencional cuando ambos operan bajo las mismas condiciones. Los resultados indican que la estructura propuesta presenta un funcionamiento altamente satisfactorio cuando opera en ambientes tanto estacionarios como no estacionarios, y debido a su realización analógica, parece ser una buena alternativa a los filtros adaptables digitales, en aplicaciones en las cuales las señales de entrada y referencia tienen componentes de alta frecuencia. El sistema propuesto se puede realizar en forma analógica con menor consumo de potencia, y con un tamaño menor que su contra parte digital.

La capítulo 7 propuso un igualador de canales rápidamente variantes con el tiempo, tal como los que se encuentran en la telefonía móvil. El igualador propuesto tiene estructura de decisión retroalimentada (DFE), y la modificación del vector de los coeficientes de ésta se realiza usando la red neuronal de Hopfield modificada. El método utilizado para la adaptación del vector de coeficientes del filtro es el mismo que el propuesto en el capítulo 6. Así la salida de DFE y la actualización del vector de los coeficientes puede obtenerse en menos tiempo y con menor consumo de potencia que el requerido por sus contra partes digitales. El funcionamiento de la estructura propuesta fue evaluado y comparado con el funcionamiento del DFE convencional usando el algoritmo RLS para la adaptación del vector de coeficientes. Para realizar la evaluación del sistema se asumen tres casos diferentes: canales de comunicación invariantes con el tiempo (estacionarios), canales de comunicación con variaciones lentas con el tiempo y canales de comunicación con variaciones rápidas con el tiempo. En todos casos se asumió un esquema de modulación 4-QAM. Estos resultados muestran que ambos, el DFE propuesto y convencional, se comportan de una manera similar cuando ambos son requeridos a igualar canales de comunicación estacionarios. Sin embargo el DFE propuesto presenta un mejor funcionamiento que el convencional cuando ambos son requeridos a igualar canales de comunicación variantes en el tiempo, especialmente cuando el canal varía rápidamente con el tiempo. Esto es debido a que el esquema propuesto está operando en tiempo continuo y entonces se actualiza el vector de coeficientes más frecuentemente. Los resultados obtenidos indican que el igualador propuesto es una buena alternativa para el igualador digital cuando es requerido a igualar canales que varían con el tiempo.

Como posible tema de investigación que se podría realizar en un futuro cercano, se puede considerar una extensión del trabajo realizado en el capítulo 6 y el capítulo 7. La estructura propuesta en el capítulo 6, cuando el número de orden  $N$  de filtro es muy grande ( $N$  es mayor que 100), necesita un número de conexiones de orden  $N^2$ . Este número grande de

conexiones dificulta su realización tanto digital como analógica. Para disminuir número de conexiones, se puede reemplazar una red neuronal de Hopfield de orden  $N$  a una estructura de varias redes neuronales celulares (CNN) de orden  $M$ , donde  $M$  puede ser 3 o 5. La red neuronal celular se ha investigado ampliamente y ha sido realizada tanto en forma analógica como digital. Esta se ha usado para la detección de borde, eliminación de ruido, etc. en el campo del procesamiento de imágenes. Sin embargo no existe una manera eficiente para obtener los pesos de las conexiones para aplicaciones generales.

Otra manera de disminuir número de conexiones de la red neuronal de Hopfield podría ser usando una estructura de filtrado con una forma de realización en paralelo, en la cual cada filtro tiene orden  $N/K$ , donde  $N$  es orden de filtro transversal y  $K$  es número de filtros en paralelo. En este caso número de conexión reduce desde  $N^2$  a  $N^2/K^2$ . Estas estructuras en paralelo ya están siendo desarrolladas, habiéndose obtenido hasta la fecha algunos resultados preliminares promisorios, tanto con estructuras de respuesta al impulso finita como infinita

En general, los sistemas adaptables no lineales, tal como los filtros con respuesta a impulso infinita, presentan problemas de convergencia a mínimos locales, durante el proceso de adaptación. Con el fin de obtener la convergencia al mínimo global evitando arribar a cualquiera de los mínimos locales, se han propuesto algunos métodos tal como el método de annealing simulado. Recientemente han propuesto también algoritmos genéticos (GA) para algunos problemas prácticas. Estos algoritmos tienen varias operaciones genéticas, tales como mutación adaptable, crossover, etc., y opera para la búsqueda del mínimo global en un sistema adaptable no lineal. Los algoritmos genéticos (GA) podrían ser un tema de investigación futura e interesante ya que se podría combinar con las investigaciones realizada, para el desarrollo de algoritmos de adaptación para estructuras de filtros adaptable de respuesta a impulso infinita.

Otro tema de investigación futura podría ser el desarrollo de algoritmos de adaptación con factores de convergencia variables con el tiempo, como los propuestos en el capítulo 4, que no involucren operaciones lógicas, de manera que puedan ser realizados fácilmente ya sea en forma digital como en forma analógica.

Finalmente, el desarrollo de sistemas en el dominio de la frecuencia es otro tema abierto, en el cual la reducción del retardo de bloque e incremento en la velocidad de convergencia siguen siendo temas fundamentales.

# CAPITULO 9

## BIBLIOGRAFÍA

Este capítulo presenta la bibliografía mas relevante usada durante la presente investigación junto con las publicaciones generadas tanto en revistas como en memorias de congreso. Las referencias se agruparon por capítulos con el fin de facilitar la lectura de la tesis.

### 9.1 Referencias Capitulo 1

- [1.1] C. Cowan y P. Grant "Adaptive Filters," Prentice Hall, Englewood Cliffs NJ 1985.
- [1.2] L. Lung y T. Soderstrom "Theory and Practice of Recursive Identification," MIT Press, Cambridge Ma, 1985.
- [1.3] B. Widrow y E. Wallach, "On Statistical Efficiency of The LMS Algorithm with Not Stationary Input Signals," IEEE Trans. on Information Theory, vol. IT-30, pag. 211-221, marzo de 1984.
- [1.4] J. Nagumo y A. Noda, "A Learning Method for System Identification", IEEE Trans on Automatic Control, vol. AC-12, pag. 282-287, junio de 1967.
- [1.5] S. Narajan, A Peterson y J. Narashimha, "Transform Domain LMS Algorithm," IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-26, pag. 304-314, agosto de 1978.
- [1.6] E. Ferrara, "Fast Implementation of LMS Adaptive Filter," IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-28, pag. 474-475, agosto de 1980.
- [1.7] J. Chao, H. Pérez Meana y S. Tsujii, "A Fast Adaptive Algorithm Using Eigenvalue Reciprocals as Stepsizes," IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-38, pag. 1343-1352, agosto de 1990.
- [1.8] J. Chao, H. Pérez Meana y S. Tsujii, "A Jumping Algorithm for Adaptive Filtering," Trans. of The IEICE, vol. 73A, pag. 1196-1206, julio de 1991.
- [1.9] G. Panda, B. Mulgrew, C. Cowan y P. Grant, "Orthogonalizing Efficient Block Adaptive Filter," IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-34, pag. 1573-1581, diciembre de 1986.
- [1.10] M.R. Asharif, F. Amano, S. Unagami y K. Murano, Acoustic "Echo Canceler Based on Frequency Bin Adaptive Filter," GLOBECOM, pag. 1940-1943, 1987.
- [1.11] A. Gilliore, "Experiments with Subband Acoustic Echo Cancelers," Proc. of The ICASSP. Pag. 1141-1144, 1987.
- [1.12] A Gilliore y M. Venerli, "Adaptive Filtering in Subbands," Proc. of The ICASSP, pag. 1572-1575, 1988.
- [1.13] W. Kellermann, "Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes," Proc. of The ICASSP, pag. 2570-2573, 1988.
- [1.14] B. Hatty, "Recursive Least Square Algorithm Using Multirate Systems for Cancellation of Acoustical Echoes," Proc. of The ICASSP, pag. 1145-1148, 1990.

- [1.15] H. Pérez Meana y F. Amano, "A New Subband Echo Canceler Structure," Trans of The IEICE, vol. E73, pag. 1625-1631, octubre de 1990.
- [1.16] F. Amano y H. Pérez Meana, "A New Subband Echo Canceler Structure," Proc. of The ICASSP, pag. 3585-3588, mayo de 1991.
- [1.17] H. Pérez Meana y F. Amano, "Acoustic Echo Cancellation Using Multirate Techniques," Trans. of The IEICE, vol. E-74, pag. 3559-3568, noviembre de 1991.
- [1.18] F. Amano, H. Pérez Meana, A De Luca y G. Duchén, "A Multirate System for Acoustic Echo Cancellation," IEEE Trans. on Communications, vol. C-43, pag. 2172-2176, julio de 1995.
- [1.19] G.D. Long, D. Schwed y D. Falconer, "Study of a Pole-Zero Adaptive Echo Canceler," IEEE Trans on Circuit and Systems, vol. CAS-34, pag. 765-769, julio de 1987.
- [1.20] H. Fan y K. Jenkins, "An Investigation of an IIR Echo Canceler," IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-36, pag. 1819-1833, diciembre de 1988.
- [1.21] J. Shynk, "Adaptive IIR Filtering," IEEE Signal Processing Magazine, pag. 4-21, abril de 1989.
- [1.22] H. Pérez Meana y S. Tsujii, "A New Orthogonalized IIR Adaptive Filter." IEE Electronics Letters, vol. 24, pag. 1566-1567, diciembre de 1988.
- [1.23] H. Pérez Meana y S. Tsujii, "A Fast Parallel Form IIR Adaptive Filter Algorithm." IEEE Trans. on Signal Processing, vol. SP-39, pag. 2118-2122, octubre de 1991.
- [1.24] H. Pérez-Meana y S. Tsujii, "IIR Adaptive Filtering via Discrete Legendre Functions," IEE Electronics Letters, vol. 24, pag. 450-451, marzo de 1988.
- [1.25] H. Pérez Meana y S. Tsujii, "A System Identification Algorithm Using Orthogonal Functions," IEEE Trans. on Signal Processing, vol. SP-39, pag. 753-755, marzo de 1991.
- [1.26] CW Gritton y A.W. Lin "Echo Cancellation Algorithms," IEEE ASSP Magazine, pag. 3037, abril de 1984.
- [1.27] H. Pérez Meana y M. Nakano, "Cancelación de Eco en Sistemas de Telecomunicación." Mundo Electrónico, vol. 207, pag. 143-150, junio de 1990.
- [1.28] K. Murano y F. Amano, "Echo Cancelling Algorithms," Enciclopedia de Telecomunicaciones, vol. 6, Marcel Decker Inc., pag. 383-409, 1993
- [1.29] N. Furuya, Y. Fukushi, Y. Ito, J. Tanaka y T. Araseki, "High Performance Custom VLSI Echo Canceler," Proc. of The ICC, pag. 1498-1592, 1985.
- [1.30] S. Minami y T. Kawasaki, "A Double Talk Detection Method," Proc. of The ICASSP, pag. 1492-1497, 1985.
- [1.31] J. Makhoul, "A Class of All Zero Lattice Digital Filter: Properties and Applications," IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-26, pag. 306-314, agosto de 1978.
- [1.32] V. Somayazulu, S. Mitra y J. Shynk, "Adaptive Line Enhancement Using Multirate Techniques," Proc. of The ICASSP, pag. 928-931, 1989.

## 9.2 Referencias Capitulo 2

- [2.1] C. W. Gritton y A.W. Lin "Echo Cancellation Algorithms", IEEE ASSP Magazine, pag. 30-37, abril de 1984.
- [2.2] B. Widrow y E. Wallach, "On Statistical Efficiency of The LMS Algorithm with Not Stationary Input Signals", IEEE Trans. on Information Theory, vol. IT-30, pag. 211-221, marzo de 1984.
- [2.3] J. Nagumo y A. Noda, "A Learning Method for System Identification", IEEE Trans on Automatic Control, vol. AC-12, pag. 282-287, junio de 1967.
- [2.4] C. Cowan y P. Grant, "Adaptive Filters", Prentice Hall, Englewood Cliffs NJ 1985.
- [2.5] L. Lung y T. Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, Cambridge Ma, 1985.
- [2.6] J. Chao, H. Pérez Meana y S. Tsujii. "A Fast Adaptive Algorithm Using Eigenvalue Reciprocals as Stepsizes", IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-38, pag. 1343-1352, agosto de 1990.
- [2.7] J. Chao, H. Pérez Meana y S. Tsujii, "A Jumping Algorithm for Adaptive Filtering". Trans. of The IEICE, vol. 73A, pag. 1196-1206, julio de 1991.
- [2.8] J. Makhoul, "A Class of All Zero Lattice Digital Filter: Properties and Applications", IE EE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-26, pag. 306-314, agosto de 1978.
- [2.9] S. Narajan, A Peterson y J. Narashimha, "Transform Domain LMS Algorithm", IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-31, pag. 609-615, junio de 1983.
- [2.10] E. Ferrara, "Fast Implementation of LMS Adaptive Filter", IEEE Trans. on Acoustic, Speech and Signal Processing, vol ASSP-28, pag. 474-475, agosto de 1980.
- [2.11] M.R. Asharif, F. Amano, S. Unagami y K. Murano, "Acoustic Echo Canceler Based on Frequency Bin Adaptive Filter ", GLOBECOM, pag. 1940-1943, 1987.
- [2.12] G. Panda, B. Mulgrew, C. Cowan y P. Grant., "Orthogonalizing Efficient Block Adaptive Filter", IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-34, pag. 1573-1581, diciembre de 1986.
- [2.13] A. Gilliore, "Experiments with Subband Acoustic Echo Canceler".. Proc. of The ICASSP. Pag. 1141-1144, 1987.
- [2.14] A Gilliore y M. Venerli,, "Adaptive Filtering in Subbands ", Proc. of The ICASSP, pag. 1572-1575, 1988.
- [2.15] W. Kellermann, "Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes", Proc. of The ICASSP, pag. 2570-2573, 1988.
- [2.16] B. Hatty, "Recursive Least Square Algorithm Using Multirate Systems for Cancellation of Acoustical Echoes", Proc. of The ICASSP, pag. 1145-1148, 1990.
- [2.17] V. Somayazulu, S. Mitra y J. Shynk, "Adaptive Line Enhancement Using Multirate Techniques", Proc. of The ICASSP, pag. 928-931, 1989.
- [2.18] H. Pérez Meana y F. Amano, "A New Subband Echo Canceler Structure", Trans of The IEICE, vol. E73, pag. 1625-1631, octubre de 1990.
- [2.19] H. Pérez Meana y F. Amano, "Acoustic Echo Cancellation Using Multirate Techniques," Trans. of The IEICE, vol. E-74, pag. 3559-3568, noviembre de 1991.

- [2.20] G.D. Long, D. Schwed y D. Falconer, "Study of a Pole-Zero Adaptive Echo Canceler", IEEE Trans on Circuit and Systems, vol. CAS-34, pag. 765-769, julio de 1987.
- [2.21] H. Fan y K. Jenkins, "An Investigation of an IIR Echo Canceler", IEEE Trans. on Acoustic, Speech and Signal Processing", vol. ASSP-36, pag. 1819-1833, diciembre de 1988.
- [2.22] J. Shynk, "Adaptive IIR Filtering", IEEE Signal Processing Magazine, pag. 4-21, abril de 1989.
- [2.23] H. Pérez Meana y S. Tsujii, "A New Orthogonalized IIR Adaptive Filter", IEE Electronics Letters, vol. 24, pag. 1566-1567, diciembre de 1988.
- [2.24] H. Pérez Meana y S. Tsujii, "A Fast Parallel Form IIR Adaptive Filter Algorithm". IEEE Trans. on Signal Processing, vol. SP-39, pag. 2118-2122, octubre de 1991.
- [2.25] H. Pérez Meana y S. Tsujii, "IIR Adaptive Filtering via Discrete Legendre Functions", IEE Electronics Letters, vol. 24, pag. 450-451, marzo de 1988.
- [2.26] H. Pérez Meana y S. Tsujii. "A System Identification Algorithm Using Orthogonal Functions, "IEEE Trans. on Signal Processing, vol. SP-39, pag. 753-755, marzo de 1991.
- [2.27] H. Pérez Meana, M. Nakano, J. C. Sánchez, L. Niño de Rivera y G. Duchén, "Echo Cancellation in Audio Terminals," Memoria Técnica, MEXICON 94, pag. 159-164. 1994.
- [2.28] H. Pérez Meana y M. Nakano, "Cancelación de Eco en Sistemas de Telecomunicación," Mundo Electrónico, vol. 207, pag. 143-150, junio de 1990.
- [2.29] K. Murano y F. Amano, "Echo Cancelling Algorithms," Enciclopedia de Telecomunicaciones, vol. 6, Marcel Decker Inc., pag. 383-409, 1993
- [2.30] N. Furuya, Y. Fukushi, Y. Ito, J. Tanaka y T. Araseki, "High Performance Custom VLSI Echo Canceler," Proc. of The ICC, pag. 1498-1592, 1985.
- [2.31] S. Minami y T. Kawasaki, " A Double Talk Detection Method." Proc. of The ICASSP, pag. 1492-1497, 1985.
- [2.32] M. Nakano, H. Perez, "A Time Varying Step Size Normalizaed LMS Algorithm for Adaptive Echo Canceler Structure", IEICE Tans. on Fundamentals of Electronics Computer Sciences, Vol. E-78-A, pag. 254-258, Febrero de 1995.
- [2.33] F. Casco, H. Perez, M. Nakano, M. Lopez, " A Variable Step Size (VSS-CC) NLMS Algorithm", IEICE Trans Fundamentals of Electronics Computer Sciences, Vol. E78-A, pag. 1004-1009. Agosto de 1995.
- [2.34] H. Pérez, M. Nakano, J. Ramírez, A. De Luca y J. C. Sánchez, "A Parallel Form Decision Feedback Equalizer," Proc. of The 1993 Midwest Conference on Circuit and Systems.
- [2.35] B. Widrow, M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", Proceedings of the IEEE, vol. 78, No. 9, pag. 1415-1442, Septiembre de 1990
- [2.36] R. P. Lippmann, "A Introduction to Computing with Neural Nets", IEEE ASSP , vol. 4, no. 2, pag. 4-22, 1987.
- [2.37] D. P. Hush y B. G. Horne, "Progress in Supervised Neural Networks", IEEE Signal Processing Magazine, pag. 8-39, enero de 1993.
- [2.38] R. H. Nielsen, "Neurocomputing", Addison-Wesley, Reading, USA, 1991.

- [2.39] J. Hertz, A. Krogh, R. G. Palmer, "Introduction to the Theory of Neural Computation", Addison-Wesley, Redwood City, CA, 1991.
- [2.40] J. A. Freeman, D. M. Skapura, "Redes neuronales --- algoritmo, aplicación y técnica de programación.", Addison-Wesley Iberoamericana, 1991, Wilmington, USA.
- [2.41] F. F. Soulie, E. Viennet, B. Lamy, "Multi-Modular Neural Network Architectures: Applications in Optical Character and Human Face Recognition", International Journal of Pattern Recognition and Artificial Intelligence, vol. 7, no. 4, pag. 721-755, 1993.
- [2.42] S. S. Young, P. D. Scott, et al, "Object Recognition Using Multilayer Neural Network", IEEE Trans. On Image Processing, vol. 6, no. 3, pag. 357-371, marzo de 1997.
- [2.43] J. D. Jackel, H. P. Graf, et. al, "An Application of Neural Net Chips: Handwritten Digit Recognition", Proc. Of the IEEE international Conference on Neural Networks, pag. II107-II115, 1988.
- [2.44] Y. LeCun, et al., "Handwritten Digit Recognition: Application of Neural Network Chips and Automatic Learning", IEEE Communications Magazine, pag. 41-46, noviembre de 1989
- [2.45] K. Fukushima y N. Wake, "Handwritten Alphanumeric Character Recognition by the Neocognitron", IEEE Trans. On Neural Networks, vol. 2, no. 3. pag. 355-365, mayo de 1991.
- [2.46] T. Kohonen, Self-organization and associative Memory , Springer-Verlag, Berlin. 1984.
- [2.47] S. Young, "A Review of Large-vocabulary Continuous-Speech Recognition", IEEE Signal Processing Magazine, Septiembre 1996.
- [2.48] N. Morgan, H. A. Bourlard, "Neural Networks for Statistical Recognition of Continuous Speech", Proceeding of the IEEE vol. 83, no.5, pag. 742-771, mayo de 1995.
- [2.49] R. Bajaj, S. Chaudhury, "Signature verification using multiple Neural Classifier", Pattern Recognition, vol. 30, no.1, pag.1-7, 1997.
- [2.50] K. Huang, H. Yan, "Off-Line Signature verification based on Geometric feature extraction and Neural Network Classification", Pattern Recognition, vol. 30, no. 1, pag. 9-15, 1997.
- [2.51] M. Nakano, H. Perez, J. Sanchez y L. Niño, "Multilayer Hopfield and Hamming Neural Network with Non Binary Input Patterns", Proc. of International Symposium on Artificial Neural Networks, ISANN 94, pag. 121-127, Diciembre de 1994.
- [2.52] J. J. Hopfield , D. W. Tank, "Neural Computation of Decisions on Optimization Problems", Biological Cybernetics, vol. 52, pag.141-152, 1985.
- [2.53] W. Jefferey, R. Rosner, "Optimization Algorithms: Simulated Annealing and Neural Network Processing", The astrophysical Journal, 310, pag. 473-481, 1986.
- [2.54] Y. T Zhou, R. Chellappa, A. Vaid, B. K. Jenkins, "Image Restoration Using a Neural Network", IEEE Trans. Acoustic Speech, Signal Processing, vol. 36, no. 7, pag. 1141-1151, Julio 1988.
- [2.55] K. Faez y M. Kamel, "Image Reconstruction From Contour Data Using a Back-Propagation Neural Network", Proc of ICASSP 94, pag. V297-V300.

- [2.56] F. U. Dowlah, A. J. DeGroot, S. R. Parker y V. R. Vermuri, "Back-propagation neural networks: systolic implementation for Seismic Signal Filtering", *Neural Networks*, vol. 1, no. 3, pag. 138-153, Julio de 1989.
- [2.57] M. Nakano, H. Perez, "Adaptive Filtering and Prediction Based on Hopfield Neural Networks", *Proc. of ICNN 97*, pag. II680-II684, junio de 1997.
- [2.58] E. Rich, "Artificial Intelligence", McGraw-Hill, Auckland, 1983.
- [2.59] A. Waibel, et.al, "Phoneme Recognition Using Time Delay Neural Networks", *IEEE Trans. On Acoustics Speech, and Signal Processing*, vol. 37, no. 3, marzo de 1989.
- [2.60] J. Schurmann, *Pattern Classification*, Wiley-Interscience, New York, 1996.
- [2.61] D. H. Hubel, T. N. Wiesel, "Brain Mechanisms of Vision", *Neuro-Vision Systems – principales and applications*. Pag. 163-176, IEEE Press, New York, 1993.
- [2.62] M. A. Weiss, *Data Structures and Algorithm Analysis*, The Benjamin/Cummings Publishing Company, inc. Redwo

### 9.3 Referencias Capitulo 3

- [3.1] S. Haykin, "Adaptive Filter Theory", Prentice Hall, Englewood Cliff NJ, 1991
- [3.2] J. Proakis, M. Rader, F. Ling, Ch. Nikias, "Advanced Signal processing, McMillian Publishing Co., Singapore
- [3.3] B. Widrow, and S. Stern, "Adaptive Signal Processing", Prentice Hall, Inc. Englewood Cliff, NJ. 1985
- [3.4] H. Perez, M. Nakano, "Características de convergencia de los algoritmos RLS y LMS con Señales Estacionarias y no Estacionarias", *Memoria técnica de Electro'97*, pag. 1-4, 1997.
- [3.5] R. Hecht-Nielsen, "Neurocomputing", Addison-Wesley Publishing Co., Reading, 1989.
- [3.6] J. A. Freeman, D. M. Skapura, "Redes Neuronales – algoritmos, aplicaciones y técnicas de programación", Addison-Wesley Iberoamericana, Wilmington, 1991
- [3.7] J. Nagumo y A. Noda, "A Learning Method for System Identification", *IEEE Trans. on Automatic Control*, vol. AC-12, no. 3, pag. 282-287, Junio de 1967.
- [3.8] J. B. Foley y F. M. Boland, "Comparison Between Steepest Descent and LMS Algorithms in Adaptive Filters", *Proc. of IEE part F*, vol. 134, no. 3, pag. 283-289, Junio de 1987.

### 9.4 Referencias Capitulo 4

- [4.1] H. Perez and F. Amano, "A New Subband Echo Canceller Structure, *Trans. the IEICE*, vol. 73E, pp.1625-1631, Oct. 1990.
- [4.2] J. Chao, H. Perez and S. Tsujii, "A Jumping Algorithm for Adaptive Filtering," *Trans. of The IEICE*, vol. J73-A, pp. 1196-1209, July 1990.
- [4.3] G. Panda, B. Mulgrew, C. Cowan and P. Grant, "Orthogonalizing Efficient Block Adaptive Filter," *IEEE Trans. on ASSP*, vol. ASSP-34, pp.1573-1581, Dec. 1986.

- [4.4] H. Perez and F. Amano, "A New Subband Echo Canceler Structure," *Trans. of The IEICE*, vol. E-73, pp. 1625-1631, Oct. 1990.
- [4.5] A. Gilliore and M. Venerli, "Adaptive Filtering in Subbands," *Proceedings of The IEEE ICASSP*, pp.1572-1575, April 1988.
- [4.6] C. Cowan and P. Grant, "Adaptive Filters," Prentice Hall, Englewood Cliff N.J., 1985.
- [4.7] J. Nagumo and A. Noda, "A Learning Method for System Identification," *IEEE Trans. on Automatic Control*, vol.12, pp. 282-287, June 1967.
- [4.8] H. Itakura and Y. Nishikawa, "On Some Characteristics of an Echo Canceler Using a Learning Identification Algorithm, " *Trans. of the IECE* vol. 60J, pp.1015-1022, Nov. 1977.
- [4.9] S. Minami and T. Kawasaki, "Double-Talk Detection Method," *Proc. of The ICASSP'85*, pp.1492-1497, 1985.
- [4.10] R.H. Kwong and E.W. Johnston, "A Variable Step Size LMS Algorithm:" *IEEE Trans. on Signal Processing*, vol. 40, pp.1633-1642, July 1992.
- [4.11] A. Sugiyama, "An Interference Robust Stochastic Gradient Algorithm," *IEEE Proc. of ICASSP*, vol. 3, pp. 539-543, 1993.
- [4.12] I. Nakanishi y Y. Furui, "A New Adaptive Convergence Factor with Constant Damping Parameter," *IEEE Trans. on Fundamentals*, vol. E78-A, No.6, pag. 649-655, junio de 1995.
- [4.13] T. Aboulnasr y K. Mayas, "A Robust Variable Step Size LMS-Type Algorithm: Analysis and Simulations," *IEEE Trans. on Signal Processing*, vol. 45, No. 3, pag. 631-639, marzo de 1997.
- [4.14] F. Casco Sánchez, "Algoritmos LMS de Paso Variable" *Disertación Doctoral. Universidad Autónoma Metropolitana, Septiembre de 1997.*

## 9.5 Referencias Capitulo 5

- [5.1] K. Murano and F. Amano, "Echo Cancelling Algorithms," *Encyclopedia of Telecommunications*, vol. 6, Marcel Dekker Inc., pp. 383-409, 1993.
- [5.2] C. Cowan y P. Grant, *Adaptive Filters*, Prentice Hall, Englewood Cliff, NJ, 1985.
- [5.3] C. W. Gritton y A.W. Lin "Echo Cancellation Algorithms", *IEEE ASSP Magazine*, pag. 30-37, abril de 1984.
- [5.4] B. Widrow y E. Wallach, "On Statistical Efficiency of The LMS Algorithm with Not Stationary Input Signals", *IEEE Trans. on Information Theory*, vol. IT-30, pag. 211-221, marzo de 1984.
- [5.5] J. Nagumo y A. Noda, "A Learning Method for System Identification", *IEEE Trans on Automatic Control*, vol. AC-12, pag. 282-287, junio de 1967.
- [5.6] L. Lung y T. Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, Cambridge Ma, 1985.
- [5.7] G.D. Long, D. Schwed y D. Falconer, "Study of a Pole-Zero Adaptive Echo Canceler", *IEEE Trans on Circuit and Systems*, vol. CAS-34, pag. 765-769, julio de 1987.
- [5.8] H. Fan y K. Jenkins, "An Investigation of an IIR Echo Canceler", *IEEE Trans. on Acoustic, Speech and Signal Processing*, vol. ASSP-36, pag. 1819-1833, diciembre de 1988.

- [5.9] J. Shynk, "Adaptive IIR Filtering", IEEE Signal Processing Magazine, pag. 4-21, abril de 1989.
- [5.10] H. Pérez Meana and S. Tsujii, "A System Identification Using Orthogonal Functions," IEEE Transactions on Signal Processing, vol. 39, No. 3, pp.752-7555, March 1991.
- [5.11] H. Pérez Meana and S. Tsujii, "A Fast Parallel Form IIR Adaptive Filter Algorithm," IEEE Transactions on Signal Processing, vol. 39, No. 9, pp. 2118-2122, Sept., 1991.
- [5.12] H. Pérez Meana y S. Tsujii, "A New Orthogonalized IIR Adaptive Filter", IEE Electronics Letters, vol. 24, pag. 1566-1567, diciembre de 1988.
- [5.13] H. Pérez Meana y S. Tsujii, "IIR Adaptive Filtering via Discrete Legendre Functions" . IEE Electronics Letters, vol. 24, pag. 450-451, marzo de 1988.
- [5.14] H. Pérez Meana, M. Nakano, J. C. Sánchez, L. Niño de Rivera y G. Duchén. "Echo Cancellation in Audio Terminals," Memoria Técnica, MEXICON 94, pag. 159-164, 1994.
- [5.15] H. Pérez Meana y M. Nakano, "Cancelación de Eco en Sistemas de Telecomunicación," Mundo Electrónico, vol. 207, pag. 143-150, junio de 1990.
- [5.16] S. Tsujii, "Handbook of Digital Signal Processing" , The IEICE Press, 1992.
- [5.17] E.R. Ferrara, "A Fast Implementation of LMS Adaptive Filters", IEEE Trans. on ASSP, Vol. ASSP-28, pp. 474-475, August 1990.
- [5.18] G. Clark, S. Mitra and S. Parker, "Block Implementation of Adaptive Digital Filters," IEEE Trans. on Circuit and Systems, vol. CAS-28, pp. 548-592, 1981.
- [5.19] F. Amano, "Frequency Domain Echo Cancellation and Its Applications to Teleconferencing Systems," Ph. D dissertation, Tokyo Institute of Technology, July 1993 (in Japanese).
- [5.20] R. W. Chang, "A New Equalizer Structure for Fast Start-up Digital Communications," The Bell Syst. Tech Journal, vol. 30, No. 6, , pag. 1969-2014, julio de 1971.
- [5.21] J. Makhoul, "A Class of All Zero Lattice Digital Filter: Properties and Applications", IE EE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-26, pag. 306-314, agosto de 1978.
- [5.22] S. Narajan, A Peterson y J. Narashimha, "Transform Domain LMS Algorithm", IEEE Trans. on Acoustic Speech and Signal Processing, vol. ASSP-31, pag. 609-615, junio de 1983.
- [5.23] G. Panda, B. Mulgrew, C. Cowan y P. Grant,, "Orthogonalizing Efficient Block Adaptive Filter", IEEE Trans. on Acoustic, Speech and Signal Processing, vol. ASSP-34, pag. 1573-1581, diciembre de 1986.
- [5.24] J. Chao, H. Pérez Meana and S. Tsujii, " A Jumping Algorithm for Adaptive Signal Processing" Trans. of The IEICE. Vol. J73-A, pp.. 1196-1206, July de 1990.
- [5.25] J. Chao, H. Pérez Meana y S. Tsujii, "A Fast Adaptive Filter Algorithm Using Eigenvalue Reciprocals as Step Size", IEEE Trans. on Signal Processing, Vol. SP-38, pp. 1343-1352, October 1990.
- [5.26] M.R. Asharif and F. Amano, "Frequency Bin Adaptive Filtering (FBAF) Algorithm and Its Applications to Acoustic Echo Cancelling," Trans. of The IEICE. vol. E74, No.8, pp. 2276-2283, August 1991.

- [5.27] M. R. Asharif and F. Amano, "Acoustic Echo Cancellation Based on Frequency Bin Adaptive Filtering (FBAF)," Proceedings and The GLOBCOM'87, pp.1940-1944, November 1987.
- [5.28] M.R. Asharif, F. Amano, S. Unagami and K. Murano, "A New Structure of Echo Canceller Based on Frequency Bin Adaptive Filtering (FBAF), Proceedings of The IEICE Digital Signal Processing Symposium pp. 165-169, November 1986.
- [5.29] J. Shynk, "Frequency Domain and Multirate Adaptive Filtering," IEEE Signal Processing Magazine, vol. 9, No. 1, pp.14-37, January 1992.
- [5.30] J. S. Soo y K. K. Pang, "Multidelay Block Frequency Domain Adaptive Filter," IEEE Trans. on Acoustic, Speech and Signal Processing," vol. 38, No. 2, pag 333-376, Feb. 1995.
- [5.31] H. Pérez Meana and F. Amano, "Acoustic Echo Cancellation Using Multirate Techniques," IEICE Transactions, vol. E-74, No. 11, pp. 3559-3568, 1991.
- [5.32] H. Pérez Meana and F. Amano, " A New Subband Echo Canceler Structure", Trans of the IEICE, Vol. E73, pp. 1625-1631, Octubre de 1990.
- [5.33] M. Honig and D. Messerschmitt, Adaptive Filters: Structures, Algorithms and Applications, Kluwer Academic Publishers, Boston Mass. USA, 1984.
- [5.34] J.B. Evans, "A New Variable Step Size Method Suitable for VLSI Implementation," Proc. of The International Conference on Acoustic, Speech and Signal Processing, pp. 2105-2108, 1991.
- [5.35] M. Nakano, H. Perez, "A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structure", IEICE Tans. on Fundamentals of Electronics Computer Sciences, Vol. E-78-A, pag. 254-258, Febrero de 1995.
- [5.36] F. Casco, H. Perez, M. Nakano, M. Lopez, " A Variable Step Size (VSS-CC) NLMS Algorithm", IEICE Trans Fundamentals of Electronics Computer Sciences, Vol. E78-A, pag. 1004-1009. Agosto de 1995.
- [5.37] J.B. Foley and F.M. Boland, "Comparison Between Stepest Descent and LMS Algorithms in Adaptive Filters," Proc. of The IEE part F, vol. 134, No. 3, pp. 283-289, june 1987.
- [5.38] J.S. Lim and A.V.Oppenheim, Advanced Topics in Signal Processing, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.
- [5.39] J. G. Proakis, M. Rader, F. Ling y Ch. Nikkias, Advanced Signal Processing, MacMillan Publishing Company, Singapore 1992.

## 9.6 Referencias Capitulo 6

- [6.1] S. Haykin, "Adaptive Filter Theory," Prentice Hall, Englewood Cliffs NJ, 1991.
- [6.2] D. Messerschmitt, "Echo Cancellation in Speech and Data Transmission," IEEE J. Of Selected Areas in Communications, Vol. SAC-2, No. 2, pag. 283-297, marzo de 1992.
- [6.3] M. Nakano-Miyatake, H. Perez-Meana, L. Niño-de-Rivera F. Casco-Sanchez y J. Sanchez-Garcia, "A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structures," IEICE Trans. on Fundamentals, vol. E-78, No. 2, pag. 254-258, Feb. 1995.

- [6.4] H. Raymond, E. Kwong y W. Johnston, "A Variable Step Size LMS Algorithm," IEEE Trans. On Signal Processing, Vol. 40, No. 8, pag. 304-314, agosto de 1992.
- [6.5] S. Kinjo y H. Ochi, "A New Robust Block Adaptive Filter for Colored Signal Input," IEICE Trans. on Fundamentals, Vol. E78, No.3, pag. 437-439, marzo de 1995.
- [6.6] H. Perez-Meana, M. Nakano-Miyatake, L.Ortiz-Balbuena, A. Martinez-Gonzalez y J. Sanchez-Garcia, "A Fast Block Adaptive Filter Algorithm with Short Processing Delay," IEICE Transactions on Fundamentals, Vol.E79, No. 5, pag. 721-726. Mayo de 1996.
- [6.7] G. Clark, S. Mitra and S. Parker, "Block Implementation of Adaptive Filters," IEEE Trans. On CAS, Vol. CAS-28, No. 3, pag.. 584-592, marzo de 1981.
- [6.8] J. Shink, "Frequency Domain and Multirate Adaptive Filtering," IEEE Signal Processing Magazine, Vol. 9. No. 1, pag. 14-37, enero de 1992.
- [6.9] H. Pérez-Meana y F. Amano, "Acoustic Echo Cancellation Using Multirate Techniques," IEICE Trans., vol. E74, No. 11, pag. 3559-3568, noviembre de 1991.
- [6.10] F. Amano, H. Perez-Meana, A. De Luca y G. Duchen, "A Multirate Acoustic Echo Canceler Structure," IEEE Trans. on Communications, vol. 43, No. 7, pag. 2172-2176, julio de 1995.
- [6.11] D. Falconer y L. Ljung, "Application of Fast Kalman to Adaptive Equalization," IEEE Trans. On Communications, vol. COM-26, No.10, pag. 1439-1446, octubre de. 1978.
- [6.12] J. Shink, "IIR Adaptive Filtering," IEEE Signal Processing Magazine. Vol. 5, No. 1, abril 1988.
- [6.13] H. Perez-Meana y S. Tsujii, "A Fast Parallel Form IIR Adaptive Filter Algorithm." IEEE Transactions on Signal Processing, Vol. 39, No. 9, pag. 2118-2122, Septiembre de 1991.
- [6.14] H. Perez-Meana y S. Tsujii, "A System Identification Using Orthogonal Functions," IEEE Trans. on Signal Processing, vol. 39, No. 3, marzo de 1991.
- [6.15] A. Martinez-Gonzalez. L. Ortiz-Balbuena, H. Perez-Meana, L. Niño-de-Rivera y J. Ramirez-Angulo, "Analog Propose of Adaptive Filter Using All Pass Functions and LMS Approach," Proc. of ISITA'94, pag. 1351-1355, noviembre de 1994.
- [6.16] S. Karni y G. Zeng, "The Analysis of the Continuous-Time LMS Algorithm," IEEE Trans on ASSP, vol. ASSP-37, No 4, pag 595-597, abril de 1989.
- [6.17] W. Wu, R. Chen y S. Chang, "An Analog Architecture for Estimation of ARMA Models," IEEE Trans. on Signal Processing, vol. 41, pag. 2946-2953, septiembre de. 1993.
- [6.18] L. Ortiz-Balbuena, A. Martinez-Gonzalez, H. Perez-Meana, L. Niño-de-Rivera y J. Ramirez-Angulo, "A Continuous Time Adaptive Filter Structure," Proc of ICASSP'95 vol. II, pag..1061-1064,1995.
- [6.19] M. Nakano-Miyatake, H. Perez-Meana, L. Ortiz-Balbuena, L Niño-de-Rivera, y J. Sanchez, "A continuous Time RLS adaptive Filter Structure Using Hopfield Neural Networks," Proc. of ISITA'96, vol. II, pag. 614-617, septiembre de. 1996.
- [6.20] M. Nakano-Miyatake, H. Perez-Meana, J. Sanchez, L. Niño-de-Rivera y L. Ortiz, "A Decision Feedback Equalizer structure Using Hopfield Neural Networks," Proc. of ICSPAT'96, pag. 555-559, octubre de 1996.

- [6.21] B. Kosko, "Neural Networks for Signal Processing," Prentice Hall, Englewood Cliff, NJ, 1992.
- [6.22] B. Lee, W. Yang y B. Sheu, "Design of a Neural -based A/D Converter Using Modified Hopfield Network," IEEE J. of Solid State Circuit, vol. SC. No. 4, pag. 1129-1135, agosto de 1989.
- [6.23] R. Lippman, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pag. 4-22, abril de 1987.
- [6.24] J. Hopfield y D. Tank, "Neural Computation of Decisions in Optimization Problems," Biological Cybernetics, vol. 52, pag. 141-152, 1985.
- [6.25] D. Tank y J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, y a Linear Programming Circuit," IEEE Trans. On Circuit and Systems, vol. CAS-33,, No. 5, pag. 533-541, mayo de 1986.
- [6.26] G. Wilson y G. Pawley, "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank," Biological Cybernetics, vol. 58, pag. 63-70, 1988.
- [6.27] J. Hertz, A. Krogh, R. Palmer, "Introduction to The Theory of Neural Computation," Addison Wesley, Reading, Mass, 1991.
- [6.28] M. Nakano-Miyatake, H. Perez-Meana, J. Sanchez-Garcia, L. Niño-de-Rivera y L. Ortiz-Balbuena, "A Continuous Time Equalizer Structure Using Hopfield Neural Networks", Proc of IASTED International Conference on Signal and Image Processing, pag. 168-172, noviembre de 1996.
- [6.29] L. E. Lim y A. V. Oppenheim, "Advanced Topics in Signal Processing," Prentice Hall, Englewood Cliffs, NJ, 1988.
- [6.30] Hector Perez-Meana y Mariko Nakano-Miyatake, A Continuous Time Structure for Filtering and Prediction, Lecture Notes in Computer Science No. 1240, Biological and Artificial Computation: From Neuroscience to Technology, Springer Verlag, Berlin, pag.1241-1250, 1997.

## 9.7 Referencias Capitulo 7

- [7.1] S. Haykin. "Adaptive Filter Theory, " Prentice Hall, Englewood Cliffs Nj, 1991.
- [7.2] J.G. Proakis, "Digital Communications", McGraw-Hill, New York, NY, 1985.
- [7.3] D.Falconer and L. Ljung, "Application of Fast Kalman to adaptive Equalization", IEEE Trans. on Communications, vol. COM-26, no. 10, pag. 1439-1446, Octubre de 1978.
- [7.4] G. Ungerboeck, "Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data Transmission Systems, "IEEE Trans. on Communications, vol. COM-22, no. 5, pag. 624-636, mayo de 1974.
- [7.5] S. Sanpei, "Development of Japanese Adaptive Equalizing Technology toward High Bit Rate Data Transmission in Land Mobile Communications", IEICE Trans. vol. E-74, no. 6, pag.1512-1521.
- [7.6] B. Mulgrew, "Applying Radial Basis Functions," IEEE Signal Processing Magazine, pp. 50-65, marzo de 1996.

- [7.7] S. Chen, S. McLaughlin, B. Mulgrew and P. M. Grant, "Adaptive Bayesian Decision Feedback Equalizer for Dispersive Radio Channels", IEEE Trans. on Communications, vol. 43, no. 5, mayo de 1995.
- [7.8] G. Kechriotis, E. Zervas y E. S. Manolakos, "Using Recurrent Neural Networks for Adaptive Communication Channel Equalization", IEEE Trans. on Neural Networks, vol. 5, no. 2 marzo de 1994.
- [7.9] G. D'Aria, F. Muratore y V. Palestini, "Simulation and performance of the pan-European land mobile radio system", IEEE Trans. Veh. Technol. vol. 41, pag. 177-189, 1992.
- [7.10] P. Balaban y J. Salz, "Dual Diversity Combining and Equalization in Digital Cellular Mobile Radio", IEEE Trans. Veh. Technol., vol. 40, no.2, pag. 342-354, 1991.
- [7.11] R. A. Ziegler y J. M. Cioffi, "Estimation of Time-Varying Digital Radio Channels," IEEE Trans. Veh. Technol. , vol. 41, no.2 , pag. 134-151, mayo de 1992.
- [7.12] K. Pahlavan y J. W. Matthews, "Performance of adaptive Matched Filter Receivers Over Fading Multipath Channels", IEEE Trans. Comm. vol. 38, no. 12, pag. 2106-2113, Diciembre de 1990.
- [7.13] W. R. Graun y U. Dersch, "A Physical Mobile Channel Model", IEEE Trans. Veh. Technol. vol. 40, no. 2, pag. 472-482, mayo de 1991.
- [7.14] M. K. Tsatsanis y G. B. Giannakis, "Equalization of Rapidly Fading Channels: Self-Recovering Methods", IEEE Trans. on Comm. vol. 44, no. 5, mayo de 1996.
- [7.15] P. Fines y A. H. Aghvami, "A new Medium and High Bit Rate 16-Ary QAM Demodulator for Land Mobile Satellite Communications", IEICE Trans. vol. E-74, no. 8, Agosto de 1991.
- [7.16] K. A. Al-mashouq y I. S. Reed, "The Use of Neural Nets to Combine Equalization with Decoding for Severe Intersymbol Interference Channels," IEEE Trans. on Neural Networks, vol. 5, no. 6, pag. 982-988, Noviembre de 1994.
- [7.17] G. D. Fornry, JR., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference, " IEEE Trans. inform. Theory, vol. IT-18, no.3, mayo de 1972.
- [7.18] F. Ling y J. Proakis, "Adaptive Lattice Decision Feedback Equalizers – Their Performance and Applications to Time Variant Multipath Channels," IEEE Trans. on Comm. vol. 33, no. 4, pag. 348-356, abril de 1985.
- [7.19] M. Nakano, H. Perez, J. Sanchez, L. Niño y L. Ortiz, "A Decision Feedback Equalizer structure Using Hopfield Neural Networks", Proc. of ICSPAT'96, pag.
- [7.20] M. Nakano, H. Perez, J. Sanchez, L. Niño y L. Ortiz, "A Continuous Time Equalizer Structure Using Hopfield Neural Networks", Proc. of IASTED 96, pag.
- [7.21] A. Martinez, L. Ortiz, H. Pérez, L. Niño y J. Ramirez, "Analog Propose of Adaptive Filter Using All Pass Functions and LMS Approach," Proc. of ISITA' 94, pag. 1351-1355, Noviembre de 1994.
- [7.22] M. Nakano, H. Pérez, L. Ortiz, L. Niño y J. Sanchez, " A Continuous Time RLS Adaptive Filter Structure Using Hopfield Neural Networks", Proc. of the ISITA'96, Septiembre de 1996.
- [7.23] S. Karni y G. Zeng, "The Analysis of the Continuous-Time LMS Algorithm", IEEE Trans. on ASSP, vol. ASSP-37, no. 4, pag. 595-597, Abril 1989.

- [7.24] L. Artiz, A. Martinez, H. Perez, L. Niño and J. Ramirez, "A Continuous Time Adaptive Filter Structure", Proc. of ICASSP'95, vol. II, pag. 1061-1064, 1995.
- [7.25] W. Wu, R. Chen y S. Chang, "An Analog Architecture for Estimation of ARMA Models," IEEE Trans. Signal Processing, vol. 41, pag. 2946-2953, Septiembre de 1993.
- [7.26] B. Kosko, "Neural Networks for Signal Processing", Prentice Hall, Englewood Cliff, NJ, 1992.
- [7.27] J. Hertz, A. Krogh y R. G. Palmer, "Introduction to the Theory of Neural Computation", Addison-Wesley, Reading, 1994.
- [7.28] S. Abe, A. H. Gee, "Global Convergence of the Hopfield Neural Network with Nonzero Diagonal Elements, " IEEE Trans. CAS-II, vol. 42, no. 1, pag. 39-45, Enero de 1995.
- [7.29] A. A. Giordano y F. M. Hsu, "Least Square Estimation with Applications to Digital Signal Processing", Wiley-International, New York, 1985.

## **9.8 Publicaciones Realizadas.**

Esta sección presenta la lista de las publicaciones realizadas durante la realización de la presente investigación doctoral, las cuales se dividen en: 4 publicaciones en revistas internacionales con arbitraje estricto, 1 capítulo de libro publicada por una editorial de reconocido prestigio, 12 publicaciones in extenso en memorias de congresos internacionales y 11 publicaciones in extenso en memorias de congresos nacionales.

### **9.8.1. Revistas Internacionales con Arbitraje Estricto**

- [1] Mariko Nakano, Héctor Pérez, Luis Niño de Rivera, Fausto Casco and Juan Carlos Sánchez, "A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structure," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science, vol. E78, No. 2, pag. 254-258, Febrero de 1995.
- [2] Fausto Casco, Hector Perez, Mariko Nakano y Mauricio López, "A Variable Step Size NLMS Algorithm Based on Correlation Criterion," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science. Vol. E78-A, No. 8, Agosto de 1995
- [3] Héctor Pérez-Meana, Mariko Nakano, Laura Ortiz-Balbuena, Alejandro Martínez-González, and Juan Carlos Sánchez-García, "A Fast Block-Type Adaptive Filter Algorithm with Short Processing Delay", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science. Vol. E79-A, No. 5 Mayo de 1996

- [4] M. Nakano-Miyatake y H. Perez-Meana, "Analog Adaptive Filtering Based on a Modified Hopfield Network," A aparecer en IEICE Transactions on Fundamentals, Nov. de 1997.

### **9.8.2. Capítulos en Libro**

- [5] H. Perez-Meana y M. Nakano-Miyatake, "A Continuous Time Structure for Filtering and Prediction using Hopfield Neural Networks," Lectures in Computer Science, Springer Verlag, 1997.

### **9.8.3. Artículos in Extenso en Memorias de Conferencias Internacionales**

- [6] Mariko Nakano M., Héctor Pérez M., Luis Niño de Rivera and Juan Carlos Sánchez , "An Extended Hamming Neural Network Algorithm for Non Binary Pattern Recognition," a aparecer en Proc. of The Midwest Conference on Circuit and Systems, Agosto de 1994, pag. 607-609, publicado en 1995.
- [7] Mariko Nakano, Héctor Pérez, Laura Ortiz, Luis Niño-de-Rivera, y Juan Carlos Sánchez "A Continuous Time RLS Adaptive Filter Structure Using Hopfield Neural Networks", 1996 IEEE International Symposium on Information Theory and Its Applications, pag. 614-617, Sept. 1996
- [8] Mariko Nakano, Héctor Pérez, Juan Carlos Sanchez, Luis Niño de Rivera, y Laura Ortiz."A Decision Feedback Equalizer Structure Using Hopfield Neural Networks" ICSPAT 96, pag. 555-559, Oct. 1996
- [9] Laura Ortiz, Alejandro Martinez, Héctor Pérez, Mariko nakano, Juan Carlos Sánchez, "Analog Echo Cancellor for Long-distance Telephone System", ICSPAT 96, pag. 438-441, Oct. 1996.
- [10] Mariko Nakano, Hector Pérez, Juan Carlos Sánchez, Luis Niño de Rivera, y Laura Ortiz, " A Continuous Time Equalizer Structure Using Hopfield Neural Networks", IASTED, Signal and Image Processing 96, pag. 168-172, Nov. 1996.
- [11] M. Nakano Miyatake y H. Pérez Meana, "An Analog Decision Feedback Structure for Land Mobile Communications," Proc. of The 1st Analog VLSI Workshop, pag. 131-134, Columbus Ohio, USA, mayo de 1997.
- [12] M. Nakano Miyatake y Héctor Pérez Meana, "Adaptive Filtering and Prediction Based on Hopfield Neural Networks", Proc of The IEEE ICNN, pag. 680-684, Houston USA, junio de 1997.

#### 9.8.4. Artículos in Extenso en Memorias de Conferencias Nacionales.

- [13] Hector Perez, Mariko Nakano, Laura Ortiz, Alejandro Martinez, "Filtrado Adaptivo en el Dominio de Frecuencia con Corto Tiempo de Retardo", Conielectrocomp, 95pag. 116-119. Abril 1995
- [14] Mariko Nakano, Héctor Pérez, Irak Vicarte, Gonzalo Duchén, "La red neuronal Neocognitron: una revisión tutorial," Conielectrocomp'95, pag. 175-180, Abril 1995.
- [15] Mariko Nakano, Hector Pérez, Gonzalo Duchén and Alejandro Martínez, "A Hierarchical Neural Network for Hand-Written Character Recognition", Conielectrocomp, 96, pag. 302-305, Feb. 1996
- [16] Hector Perez, Mariko Nakano, Laura Ortiz, Alejandro Martinez, Luis Niño de Rivera, "A Short Delay Jumping Adaptive Filter Algorithm", Conielectrocomp, 96, pag. 217-222, Feb. 1996
- [17] Luis Niño de Rivera, Mariko Nakano, Juan Carlos Sánchez, Héctor Pérez, Edgar Sánchez, "A New Approach for Human Face Classification with Shifted Position Independence", Conielectrocomp'96, pag. 211-213. Feb. 1996.
- [18] Marco Martinez, Héctor Pérez, Mariko Nakano, "Evaluación Experimental de Cuatro "Versiones del Algoritmo LMS Operando Como Cancelador de Eco", CIIIIE'96, pag. 155-162, Sept. 1996.
- [19] Mariko Nakano, Héctor Perez, Luis Niño de Rivera, Juan Carlos Sánchez, y Yudiel Perez, "A Frequency Scrambler Structure for Secure Communication Systems", CIIIIE'96, pag. 163-169, Sept. 1996.
- [20] Luis Niño de Rivera, Laura Ortiz, Mariko Nakano, Héctor Pérez, Gonzalo Duchén, "Linea de Retardo Analogica Simple para Filtrado Adaptivo", Electro'96, pag. 111-115, Oct. 1996.
- [21] H Pérez-Meana, M. Nakano-Miyatake, L. Niño de Rivera y Oyarsabal, J. Sánchez García, "Modificaciones al Algoritmo LMS para Reducir su Complejidad o Mejorar sus Características, Memoria Técnica CONIELECTROCOMP'97, pag. 32-26, Puebla México, febrero de 1997.
- [22] M. Nakano Miyatake, H. Pérez Meana, L. Niño de Rivera y Oyarsabal, Gonzalo Duchén Sánchez, Yudiel Pérez Espejo, "Scrambler Structure Using Cosine Transform," Memoria Técnica CONIELECTROCOMP'97, pag. 209-212, Puebla México, febrero de 1997.
- [23] H. Pérez Meana y M. Nakano Miyatake, "Características de Convergencia de los Algoritmos RLS y LMS con Señales Estacionarias y No Estacionarias," Memorias del ELECTRO97, pag. 1-4, Chihuahua Chih octubre de 1997.

